

Article

# Low Complexity Adaptive Detection of Short CPM Bursts for Internet of Things in 6G

Zihao Pan , Heng Wang, Bangning Zhang and Daoxing Guo \*

College of Communication Engineering, Army Engineering University of PLA, Nanjing 210001, China

\* Correspondence: xyzgfg@sina.com

**Abstract:** With the standardization and commercialization of 5G, research on 6G technology has begun. In this paper, a new low-complexity soft-input–soft-output (SISO) adaptive detection algorithm for short CPM bursts is proposed for low-power, massive Internet of Things (IoT) connectivity in 6G. First, a time-invariant trellis is constructed on the basis of truncation in order to reduce the number of states. Then, adaptive channel estimators, recursive least squares (RLS), or least mean squares (LMS), are assigned to each hypothetical sequence by using the recursive structure of the trellis, and survivor processing (PSP) is used to improve the quality of channel estimation and reduce the number of searching paths. Then, the RLS adaptive symbol detector (RLS-ASD) and LMS adaptive symbol detector (LMS-ASD) could be acquired. Compared to using a least-squares estimator, the RLS-ASD avoids matrix inversion for the computation of branch metrics, while the LMS-ASD further reduces the steps in the RLS-ASD at the cost of performance. Lastly, a soft information iteration process is used to further improve performance via turbo equalization. Simulation results and analysis show that the RLS-ASD improves performance by about 1 dB compared to the state-of-the-art approach in time-variant environments while keeping a similar complexity. In addition, the LMS-ASD could further significantly reduce complexity with a power loss of approximately 1 dB. Thus, a flexible choice of detectors can achieve a trade-off of performance and complexity.

**Keywords:** continuous-phase modulation; short bursts; soft input–soft output; adaptive algorithms



**Citation:** Pan, Z.; Wang, H.; Zhang, B.; Guo, D. Low Complexity Adaptive Detection of Short CPM Bursts for Internet of Things in 6G. *Sensors* **2022**, *22*, 8316. <https://doi.org/10.3390/s22218316>

Academic Editor: Claudia Campolo

Received: 9 September 2022

Accepted: 26 October 2022

Published: 29 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

### 1.1. Background and Motivation

Globally, 5G networks have been deployed since 2020, and some key capacities are being standardized but do not meet the communication demands for 2030 and beyond. Researchers have begun to focus on 6G wireless communication networks to meet future requirements [1]. The IoT has recently drawn substantial interest due to its great potential for several applications in 6G wireless networks [2–4]. With the mass deployment of wireless sensors, it is difficult work to replace batteries or provide connections to the power grid due to random and complex environments. In these scenarios, the energy efficiency and cost of sensors are critical problems to be solved [5]. Future heterogeneous IoT integrated 6G networks would embrace active and passive devices in providing various services with diverse resources.

A backscatter communication technique [5–8] is proposed for passive devices that can harvest energy from the environment, providing the possibility to realize low power consumption and high-efficiency data transmission. In active devices with radio-frequency links, continuous phase modulation (CPM) is a good candidate for the communication of high power efficiency in uplinks [9–11]. Continuous phase modulation (CPM) provides a feasible transmission scheme for power- and bandwidth-limited wireless communication because of its high spectral efficiency and power efficiency [12]. Moreover, its constant complex envelope is robust to the nonlinearities caused by power amplifiers (PAs), which further improves power efficiency [13].

However, compared to linear modulations, CPM transmission over frequency-selective channels is a challenging task because of nonlinearity [11]. Poor physical-layer waveforms worsen the performance of the communication system, and even the superior power efficiency brought by PA can be undone because of the high complexity in the receiver [9]. Therefore, this paper focuses on the design of equalizers for IoT devices that employ low data rates and short CPM burst transmissions. In practical communications, the receiver does not have perfect channel state information, which must be estimated.

Optimal coherent detection can be approximately achieved by inserting a sufficient number of pilot symbols. In [14], a universal data-aided method was proposed by inserting periodically data-dependent symbols for CPM in flat Rayleigh fading channels. The channel response was generated by a mean squared error front-end filter and interpolation filter, which was input to a coherent CPM demodulator using the Viterbi algorithm. In fast-fading flat channels, the authors in [15] provided a data-aided channel estimation algorithm with local B-splines, and the results demonstrated an excellent trade-off between bit error rate performance and complexity. In [16], the authors derived a maximal posterior symbol detector for CPM over a double-selected channel via the PAM decomposition of the CPM. Data-aided initial channel estimation was achieved by using the pilot symbol, and the quality of the channel estimation was iteratively improved with turbo estimation. To further reduce complexity, a suboptimal equalizer based on a linear minimal mean-square error equalization was achieved at a cost of 2 dB. However, the data-aided method could drastically increase the overhead-to-payload ratio in short bursts because of additional training bits. Similarly, the addition of a cyclic prefix or unique words in low-complexity frequency-domain equalization (FDE) for CPM [17–21] could create the same problem. Thus, pilot- or FDE-based methods are not feasible schemes for short CPM bursts.

To solve this problem, blind channel equalization in the time domain is an effective alternative to avoid increasing the overhead-to-payload ratio. The authors in [22] used the Tong–Xu–Kailath algorithm to CPM and extracted the second-order statistics of the signal for channel estimation. The eigenvector method was used to identify the channel from a fourth-order cross-cumulant matrix under the GSM channel in [23]; to track the time-variant channel, the turbo-estimation approach was used to improve the quality of the channel estimates. However, the exact calculation of statistics requires a sufficient number of symbols to restrain the error due to noise, and the channel estimates obtained in short bursts are not reliable.

Other blind equalization can be achieved by the CPM–channel trellis. The joint trellis can be built from CPM and channel memory, which can be represented by a finite state machine (FSM). In [24–27], the FSM was described with a hidden Markov model (HMM). Thus, the channel response could be regarded as the unknown parameter in the HMM, which could be estimated with the expectation–maximization/Baum–Welch algorithm. However, HMM-based equalization techniques are suitable for time-variant channels. Thus, the idea that data-aided estimation could be embedded into the structure of the joint trellis [28,29]. A forward/backward adaptive soft-in soft-out (FAFB-SISO) algorithm [29] based on a full-state trellis was derived for linear modulation, which was optimal. Then, a forward adaptive SISO (FA-SISO) algorithm [30] that considered the channel correlation in only one direction was proposed for MSK; the author in [31] proposed various reduced-state A-SISO (RS-A-SISO) algorithms for reducing complexity.

Recently, a new noncoherent symbol-by-symbol detection algorithm of short quaternary CPM burst [9] was proposed as a better approach than the conventional one when short bursts are transmitted with only a few training bits. However, a major drawback of the detector is its high complexity because of matrix inversion for the calculation of branch metrics. In this paper, we propose a symbol-by-symbol detector with low complexity for short CPM bursts in IoT communication systems.

### 1.2. Contribution

- The symbol detector is proposed on the basis of the trellis. To reduce the number of states and tackle a memory that linearly grows with time, a simplified trellis was constructed on the basis of the truncation of the memory.
- On the basis of the simplified trellis, the channel response is acquired by embedding the adaptive channel estimator, RLS or LMS algorithm, in the calculation of the trellis branch metric. The channel estimation is updated iteratively by using the recursiveness of the trellis, which can avoid matrix inversion when the LS estimator is used to calculate metrics and reduce complexity.
- In processing the trellis search, the BCJR algorithm was applied with per-survivor processing (PSP) in the forward direction, which could iteratively ensure the accuracy of channel estimation for each state.
- Lastly, a link-level simulation platform of the communication system was established to evaluate the performance of different symbol detectors.

### 1.3. Paper Outline

The rest of this paper is organized as follows. Section 2 provides the representation of CPM signals transmitted over fading channels and a communication system. Next, in Section 3, the new low-complexity symbol detector and the turbo scheme are introduced and are described in Section 4. The performance of the proposed symbol detector was evaluated and analyzed on the basis of the link-level simulation platform in Section 4. Lastly, the paper is concluded in Section 5.

## 2. System Model

### 2.1. CPM Representation

Before discussing the communication system model, we give the representation of a CPM signal. Let  $\mathbf{a}_n = [a_0, a_1, \dots, a_n]^T$  be the vector containing independent and identically distributed (iid) symbols, where elements  $a_i$  belong to the M-ary alphabet defined by  $a_i \in \{\pm 1, \pm 3, \dots, \pm(M-1)\}$ . The equivalent baseband representation  $x(t, \mathbf{a}_n)$  of the transmitted M-ary CPM signal is written as follows:

$$x(t, \mathbf{a}_n) = \sqrt{\frac{E_s}{T}} \exp(j\phi(t, \mathbf{a}_n)), 0 \leq t \leq nT. \quad (1)$$

where  $T$  represents the symbol duration, and  $E_s$  is the symbol energy. The information-bearing phase is defined as  $\phi(t, \mathbf{a}_n) = 2\pi h \sum_{i=0}^n a_i q(t - iT)$ , where  $h$  is the modulation index.  $q(t) = \int_0^t g(\tau) d\tau$  is the phase pulse, such that  $q(t) = 1/2$  for  $t \leq L_{cpm}T$ , where  $L_{cpm}$  is the memory length of the CPM. If  $L_{cpm} = 1$ , we speak of a full-response CPM, and if  $L_{cpm} \geq 1$ , a partial-response CPM is obtained.

The CPM signal is transmitted through a fading channel, and distorted by additive white Gaussian noise (AWGN). In complex baseband representation, the received continuous-time signal is given by

$$y(t) = x(t, \mathbf{a}_n) * h(t) + v(t), 0 \leq t \leq nT. \quad (2)$$

where  $*$  is the convolutional operator. A discrete-time representation of the received signal is needed for digital signal processing. Then, choosing a sampling period  $T_s$ , such that  $T_s = T/\varepsilon$ , where  $\varepsilon$  is the number of samples per symbol. We use notation  $y_k = y(kT_s)$ ,  $x_k = x(kT_s)$ ,  $h_k = h(kT_s)$ ,  $v_k = v(kT_s)$ . Then, the  $k$ th sample of  $y(t)$  by fractionally spaced sampling every  $T_s$  in Equation (2) is

$$\begin{aligned} y_k = y(kT_s) &= \sum_{i=0}^{l-1} h_i(kT_s) x(kT_s - iT_s) + v(kT_s) \\ &= \sum_{i=0}^l h_{i,k} x_{k-i} + v_k. \end{aligned} \quad (3)$$

where  $T_s$  is the channel resolution, and  $l$  is the length of the channel in terms of  $T_s$ .

To obtain the matrix form for Equation (3), we give column vectors  $\mathbf{x}_n, \mathbf{y}_n, \mathbf{v}_n$  comprising the observables of  $x(t, \mathbf{a}_n), y(t), v(t)$  from symbol time 0 up to time  $n$ . Therefore, the discrete-time representation of the received signal, CPM, and noise can be written as  $\mathbf{y}_n = [y_0, y_1, y_2 \dots, y_{n\epsilon-1}]^T, \mathbf{x}_n = [x_0, x_1, x_2 \dots, x_{n\epsilon-1}]^T, \mathbf{v}_n = [v_0, v_1, v_2 \dots, v_{n\epsilon-1}]^T$ . Thus, the  $i$ th path of the CPM signal can be defined as

$$\mathbf{x}_{n-i} = [0_i, x_0, x_1, x_2 \dots, x_{n\epsilon-i-1}]^T \tag{4}$$

where  $0_i$  is a row vector containing  $i$  zeros elements. All paths of the CPM signals were merged together and could be rewritten as

$$\mathbf{X}(\mathbf{a}_n) = [\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-l+1}]^T \tag{5}$$

The channel response is  $\mathbf{h} = [h_0, h_1, \dots, h_{l-1}]^T$  and the discrete-time representation of the received signal in the form of matrix can be written as

$$\mathbf{y}_n = \mathbf{X}(\mathbf{a}_n)\mathbf{h} + \mathbf{v}_n \tag{6}$$

### 2.2. Serially Concatenated CPM System in Fading Channels

To derive the SISO symbol-by-symbol detector for CPM, we considered a serially concatenated CPM (SCCPM) setup for our system, as shown in Figure 1. When the CPM detector is SISO, CPM with channel coding could improve performance with a turbo equalization scheme in the form of SCCPM, where the decoder and detector exchange extrinsic information in terms of log-likelihood ratios (LLRs).

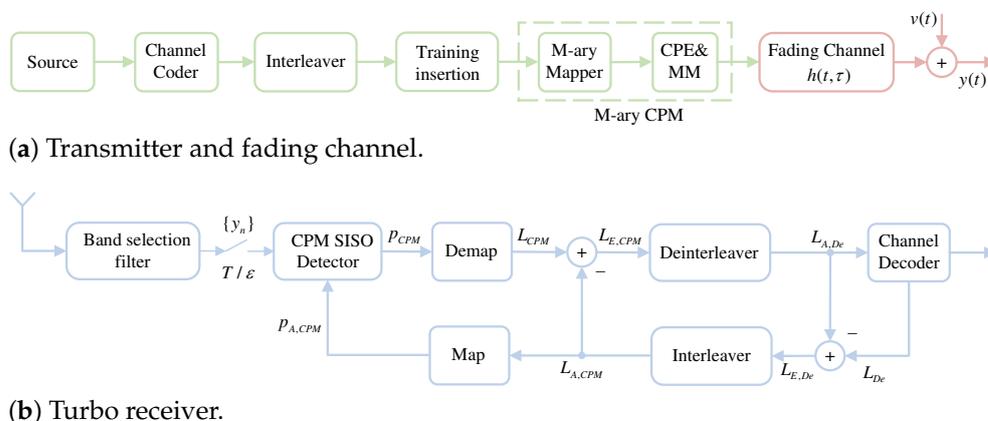


Figure 1. Communication system model of CPM, including transmitter, channel, and turbo receiver.

In the transmitter, a block of message bits  $m$  is encoded with a channel coder and permuted with an interleaver to randomize the order of the code bits. Subsequently, specific training bits are appended to the codeword in the training insertion block, helping the receiver in locking fast and reliably. An  $M$ -ary gray mapper was considered to obtain a symbol sequence by the obtained burst (in bits). Then, short CPM bursts are generated by a continuous phase encoder (CPE) and a memoryless modulator (MM) and transmitted over the channel. The channel was completely specified by linear distortions, represented by impulse response  $h(t, \tau)$  and noise variance  $\sigma_v^2$  that was assumed to be known at the detector.

On the receiver side, the discrete-time representation of the received signal is obtained by oversampling the output of the band selection filter. Then, given a priori LLRs  $L_{A,CPM}$  (set zero at first iteration) from the decoder to detector, the latter uses the received signal and a priori symbol probabilities  $P_{A,CPM}$  updated by  $L_{A,CPM}$  to calculate the posterior

probability  $P_{CPM}$  of transmitted symbol  $\mathbf{a}_n$ . Three steps are required from the  $P_{CPM}$  to the a priori LLRs  $L_{A,De}$ , which is fed to the decoder.

First, the symbol probability is converted into codeword probability by the demapper, and then converted into the form of LLRs  $L_{CPM}$ . Subsequently, extrinsic information  $L_{E,CPM}$  is computed by subtracting  $L_{A,CPM}$ , also called intrinsic information, from  $L_{CPM}$ . In turbo equalization, feeding back intrinsic information encounters problems of positive feedback, which leads to convergence to a suboptimal solution far from the globally optimal solution. Lastly, extrinsic information LLRs on the coded bits  $L_{E,CPM}$  are deinterleaved to obtain a priori LLRs for decoder  $L_{A,De}$ . Just like subtracting intrinsic information, the interleaver is included in the iteration process to further disperse the direct feedback effect. These LLRs in the decoder are mapped again and similarly fed back to the detector. The process is repeated for a given maximal number of iterations or until convergence. Next, the detailed processing of the CPM detector is given.

### 3. Derivation of the SISO Symbol-by-Symbol Detector

#### 3.1. Transition/Branch Metric

Bayes' theorem shows that the posterior probability density function (pdf) of transmitted symbol vector  $\mathbf{a}_n$  is given by:

$$f(\mathbf{a}_n, \mathbf{h} | \mathbf{y}_n) = \frac{f(\mathbf{y}_n | \mathbf{a}_n, \mathbf{h}) f(\mathbf{a}_n) f(\mathbf{h})}{f(\mathbf{y}_n)}. \quad (7)$$

The optimal estimated symbol sequence can be obtained by maximizing Equation (7). In the receiver, received signal  $\mathbf{y}_n$  is perfectly known and does not affect the maximization of Equation (7), which can be regarded to be a constant. Similarly, there is no prior information on the channel for the receiver, exerting no influence on the maximization problem. Therefore, the posterior pdf can be expressed as follows:

$$f(\mathbf{a}_n, \mathbf{h} | \mathbf{y}_n) \sim f(\mathbf{y}_n | \mathbf{a}_n, \mathbf{h}) f(\mathbf{a}_n), \quad (8)$$

where  $f(\mathbf{a}_n) = \prod_{i=0}^N a_i$  are the priori, which can be calculated with the extrinsic log-likelihood ratios (LLRs) from channel decoder. In standard equalization, i.e., with no iteration,  $f(\mathbf{a}_n)$  are discrete uniform priors.

The conditional pdf of  $\mathbf{y}_n$  conditioned on the  $\mathbf{a}_n, \mathbf{h}$  is written as

$$f(\mathbf{y}_n | \mathbf{a}_n, \mathbf{h}) = \frac{1}{(\pi\sigma_v^2)^{en}} \exp\left(-\frac{1}{\sigma_v^2} \|\mathbf{y}_n - \mathbf{X}(\mathbf{a}_n)\mathbf{h}\|^2\right). \quad (9)$$

Logarithm likelihood function Equation (12) can be simplified by taking the logarithm, which is in the following form:

$$\Lambda_n(\mathbf{a}_n, \mathbf{h}) = -\frac{1}{\sigma_v^2} \|\mathbf{y}_n - \mathbf{X}(\mathbf{a}_n)\mathbf{h}\|^2. \quad (10)$$

When the knowledge of the channel state information (CSI) is perfect, the most likely symbol sequence  $\hat{\mathbf{a}}_{ML}$  can be found with the maximization of Equation (10).

$$\hat{\mathbf{a}}_{ML} = \operatorname{argmax}_{\mathbf{a} \in \mathcal{A}^{1 \times n}} \Lambda_n(\mathbf{a}_n, \mathbf{h}). \quad (11)$$

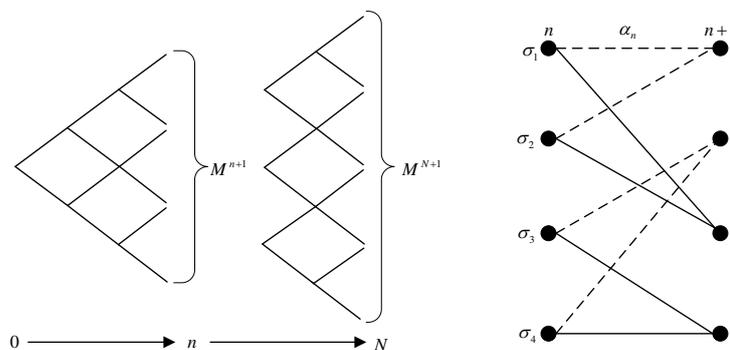
However, the channel response is unknown and can be estimated with the known transmission sequence denoted as  $\mathbf{h}_{est} = f[\mathbf{y}_n, \mathbf{a}_n]$ . Then, the branch metric of input symbol  $a_n$  at the  $n$ th time slot is written as follows:

$$\gamma_n(a_n) = F[a_n, \mathbf{y}_n, \mathbf{h}_{est}]. \quad (12)$$

where  $h_{est}$  is the channel response estimated by the hypothetical transmit sequence and the received sequence before time  $n$ . Using the data-aided idea, suboptimal sequential maximal likelihood estimation can be achieved through a Viterbi-like algorithm [28]. However, in this paper, we focus on the derivation of the symbol-by-symbol detector rather than the sequence detector.

### 3.2. Derivation of SISO Symbol Detection

Given the branch metric, before deriving the symbol detector, an appropriate trellis needs to be constructed. On the one hand, when the metric depends on all the previous input symbols before the  $n$ th symbol, the memory grows linearly with sequence length [29], and a tree is built as shown in Figure 2, each node of which represents a sequence path. To construct a time-invariant trellis, the truncation of the tree memory can be used to force a fixed number of states in the trellis. On the other hand, when memory length  $l$  is constant, the complexity of the trellis search is proportional to number of branches  $M^l$ , which becomes intractable for large memory  $l$  and/or constellation size  $M$ . To reduce complexity, a simplified trellis with a reduced number of states can be built by truncating the memory. Therefore, the following derivation of the SISO symbol detector was implemented on the basis of the truncation of the trellis.



**Figure 2.** (left) Forward tree structure; (right) trellis diagram of the binary scheme with truncated length  $L = 2$  at time  $n$ .

First, the symbol sequence can be divided into input symbols, trellis states at time  $n$ , and symbols obtained with PSP, which is defined as  $a_n = [a_n, \sigma_n, P(\sigma_n)]$ , where

$$\sigma_n = [a_{n-1}, a_{n-2}, \dots, a_{n-L}], \tag{13}$$

is the state of the trellis after truncation,  $L$  is the truncation length, and  $P(\sigma_n)$  is the survivor sequence of arrival state  $\sigma_n$  written as

$$P(\sigma_n) = [a_{n-L-1}, a_{n-L-2}, \dots, a_1, a_0]. \tag{14}$$

Taking binary CPM with truncation length  $L = 2$  as an example, the truncated trellis is shown in Figure 2, where the state of trellis is  $\sigma_n = [a_{n-1}, a_{n-2}]$  and the branch metric defined by Equation (12). We used the BCJR algorithm to achieve symbol-by-symbol CPM detection. There were  $M$  branches from  $n$ th time slot to each state at  $(n + 1)$ th time slot, and each branch was bundled with a channel response depending on all the symbols up to the  $n$ th symbol. Then, the appropriate channel response could be selected for the next state  $\sigma_{n+1}$  by applying PSP in the forward recursion of BCJR, PSP-BCJR. The description of the PSP-BCJR algorithm for the symbol detection of CPM is given below.

In the forward recursion, channel estimation at states  $\sigma_n$  can be defined as follows:

$$h(\sigma_n) = g(\mathbf{r}_n, \sigma_n, P(\sigma_n)). \quad (15)$$

The channel response is calculated from the symbol sequence arriving at state  $\sigma_n$ , and the received sequence and  $g(\cdot)$  imply the dependence of the channel response on the received and transmitted sequences, representing the estimation method. The state is transferred from  $\sigma_n$  to  $\sigma_{n+1}$  after entering symbol  $a_n$ . The computation of the branch metrics is determined with input symbol  $a_n$ , state  $\sigma_n$  and its survivor sequences  $P(\sigma_n)$ , and received signal  $\mathbf{y}_n$ :

$$\gamma(\sigma_n, \sigma_{n+1}) = F\{\sigma_n \rightarrow \sigma_{n+1}, g[\mathbf{y}_n, a_n, \sigma_n, P(\sigma_n)]\}. \quad (16)$$

According to the branch metric, the forward recursion (initially zero) is written as follows:

$$\beta_{n+1}^f(\sigma_{n+1}) = \sum_{\sigma_n: \sigma_n \rightarrow \sigma_{n+1}} \beta_n^f(\sigma_n) \exp(\gamma(\sigma_n, \sigma_{n+1})) P_n(\sigma_{n+1} | \sigma_n). \quad (17)$$

where  $P_n(\sigma_{n+1} | \sigma_n)$  is the probability that the state of termination  $\sigma_{n+1}$  is transferred from  $\sigma_n$ , which is the prior probability of input symbols  $a_n$ . The prior probability was assumed to be uniform discrete values that could be updated by the extrinsic probabilities fed back from the decoder after every turbo iteration. As shown in Figure 3, PSP is used in forward recursion to simplify searching, and survivor sequences  $P(\sigma_{n+1})$  and survivor channel estimation  $h(\sigma_{n+1})$  of  $\sigma_{n+1}$  can be selected according to

$$\hat{\sigma}_n = \arg \max_{\sigma_n: \sigma_n \rightarrow \sigma_{n+1}} \beta_n^f(\sigma_n) \exp(\gamma(\sigma_n, \sigma_{n+1})) P_n(\sigma_{n+1} | \sigma_n). \quad (18)$$

After completing the forward recursion, backward recursion is similarly calculated:

$$\beta_n^b(\sigma_n) = \sum_{\sigma_{n+1}: \sigma_n \rightarrow \sigma_{n+1}} \beta_{n+1}^b(\sigma_{n+1}) \exp(\gamma(\sigma_n, \sigma_{n+1})) P_n(\sigma_{n+1} | \sigma_n). \quad (19)$$

Lastly, a posteriori probabilities for symbol  $a_n = u$  are the summation over all recursions with the same input  $u$ .

$$P(a_n = u) = \sum_{u: \sigma_n \rightarrow \sigma_{n+1}} \beta_n^f(\sigma_n) \exp(\gamma(\sigma_n, \sigma_{n+1})) \beta_{n+1}^b(\sigma_{n+1}) P_n(\sigma_{n+1} | \sigma_n). \quad (20)$$

where  $u: \sigma_n \rightarrow \sigma_{n+1}$  represent all transitions induced by input  $u$ . In practice, for feasibility and numerical stability, BCJR can be realized in the log domain called Log-MAP. Therefore, in subsequent simulations, the CPM detector was implemented in the log domain. More information and details can be found in [32]. Summarizing Equations (15)–(20), each branch is assigned a channel estimator at time  $n$ , which can track the variations of the channel while estimating the symbol by alternating between symbol and channel estimation. In [9], the unknown channel in Equation (13) is replaced with maximal likelihood (ML) estimation, and a suboptimal sequence metric is derived as follows.

$$\begin{aligned} \Gamma_{ML}(\mathbf{a}_n) &= \frac{1}{\sigma_w^2} \mathbf{y}_n^H X(\mathbf{a}_n) h_{ML}(\mathbf{a}_n) \\ &= \frac{1}{\sigma_v^2} \mathbf{r}_n^H X(\mathbf{a}_n) \left( X^H(\mathbf{a}_n) X(\mathbf{a}_n) \right)^{-1} X^H(\mathbf{a}_n) \mathbf{y}_n \\ &= \frac{1}{\sigma_v^2} \mathbf{y}_n^H X(\mathbf{a}_n) X^+(\mathbf{a}_n) \mathbf{y}_n. \end{aligned} \quad (21)$$

In order to derive the symbol detector, an incremental metric is used as the branch metric:

$$\Delta_n(\mathbf{a}_n) = \Gamma_n(\mathbf{a}_n) - \Gamma_{n-1}(\mathbf{a}_{n-1}) = \frac{1}{\sigma^2} \mathbf{y}_n^H X(\mathbf{a}_n) h_{ML}(\mathbf{a}_n) - \frac{1}{\sigma^2} \mathbf{y}_{n-1}^H X(\mathbf{a}_n) h_{ML}(\mathbf{a}_{n-1}) \quad (22)$$

which is the implementation of  $F(\cdot)$  in Equation (12). In fact, the authors in [9] assigned an ML estimator to each state that could be regarded as an instance of the symbol–symbol detector derived in this paper. Since the noise was white and Gaussian, the ML estimator was equivalent to the least-squares (LS) estimator, and the metric in [9] is noncoherent. Therefore, the detector is denoted as the least-squares noncoherent symbol detector (LS-NSD). However, a major drawback of the LS-NSD is its high complexity in the computation of branch metrics caused by matrix inversion in LS estimation. To simplify the calculation of the branch metric, some ingenious channel estimation methods can be considered to take place of the LS estimator.

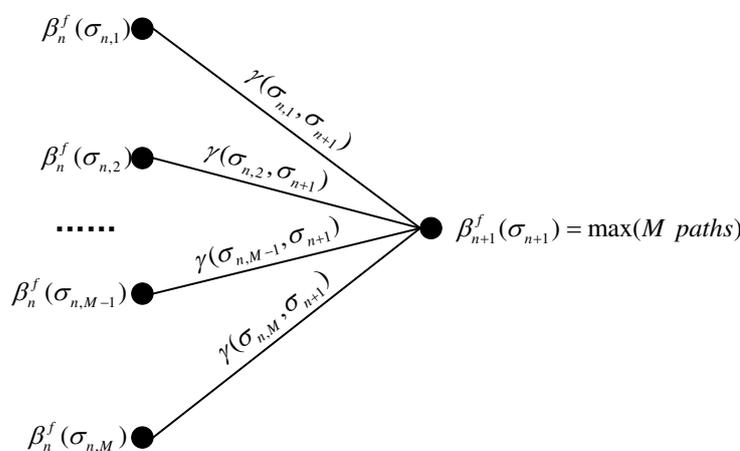


Figure 3. PSP-based forward recursion in trellis.

In the process of trellis search, the update of channel estimation has recursiveness, and the forward recursion can be seen as a constant input of new data. It is easy to associate RLS and LMS [33] as a fast algorithm for LS estimation that can be used to process real-time data, which is highly convertible with the forward recursion in trellis search. To simplify the calculation of the branch metric, RLS or LMS can be applied in the channel estimation of each branch to avoid matrix inversion. In addition, both algorithms can identify the characteristics of the dynamic system in real time, allowing for the more accurate tracking of channel variations in a dynamic channel environment.

RLS requires the initial values of the channel response and the  $P$ , which can be initialized by the training sequence. However, under the constraint of blind equalization, it is assumed that the LS-NSD is still used to calculate the channel response for the first  $t$  time slots, and the RLS estimator is used after the  $(t + 1)$ th time slots. The  $P$  is initialised by  $(X(\mathbf{a}_t)^H X(\mathbf{a}_t))^{-1}$  at the time  $t$ .

The gain vector at the  $k$ th moment (in terms of sample interval):

$$\mathbf{K}_{k+1} = \frac{\frac{1}{\lambda} \mathbf{P}_k \mathbf{x}(t_k)}{1 + \frac{1}{\lambda} \mathbf{x}(t_k)^T \mathbf{P}_k \mathbf{x}(t_k)} \quad (23)$$

Updating the inverse of the correlation matrix  $\mathbf{P}$  at the  $(k + 1)$ th moment:

$$\mathbf{P}_{k+1} = \frac{1}{\lambda} \mathbf{P}_k - \frac{1}{\lambda} \mathbf{K}_{k+1} \mathbf{x}(t_k)^T \mathbf{P}_k \quad (24)$$

Then, calculating the channel response:

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k - \mathbf{K}_{k+1}(\mathbf{x}(t_k)^T \hat{\mathbf{h}}_k - y_k). \quad (25)$$

where  $\mathbf{x}(t_k) = \{x_{k-i}\}_{i=0}^{l-1}$  are fractionally spaced signal samples during channel length  $l$ . To further reduce complexity, the channel estimator for the branch metric can use the LMS algorithm:

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k - \beta(\mathbf{x}(t_k)^T \hat{\mathbf{h}}_k - y_k)\mathbf{x}^*(t_k). \quad (26)$$

where  $\beta$  is the step size chosen to be a suitable constant in LMS. To ensure the convergence of the algorithm, the range of the step satisfies

$$0 < \beta < \frac{\kappa}{\text{tr}[\mathbf{R}]}, 0 < \kappa < 2, \quad (27)$$

where  $\text{tr}[\mathbf{R}]$  is the trace of the autocorrelation matrix of  $\mathbf{x}(t_k)$ . In the case of a time-variant channel, normalized least mean squares (NLMS) can be used to improve tracking channel variations, and the step size is written as follows:

$$\beta(k) = \frac{a}{\mathbf{x}(t_k)^T \mathbf{x}(t_k) + \gamma}, \quad (28)$$

where  $\gamma$  is a constant selected to prevent underflow.

Since the channel estimation methods are adaptive, the two symbol detectors proposed in this paper are denoted as the RLS adaptive symbol detector (RLS-ASD) and the LMS/NLMS adaptive symbol detector (LMS/NLMS-ASD).

### 3.3. Computational Complexity Analysis

Table 1 shows a detailed comparison of the complexity of RLS-ASD, LMS-ASD, and NLMS-ASD, and detection in [9] via floating-point operations (FLOP). The analysis of the computational effort of the different detectors is shown in Appendix A.

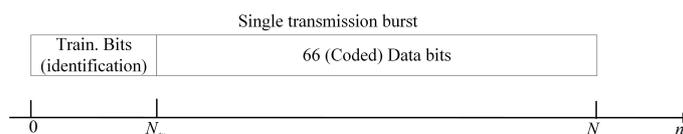
**Table 1.** Complexity comparison among detectors.

Detector Type	FLOP at $n$ th Transition Interval
LS-NSD	$l^3 + 2l^2\varepsilon + 2l\varepsilon + 2\varepsilon nl + l + 1$
RLS-ASD	$6l^2\varepsilon + 2l\varepsilon + 2\varepsilon nl - \varepsilon$
LMS-ASD	$2nl\varepsilon + 3\varepsilon l$
NLMS-ASD	$2nl\varepsilon + 5\varepsilon l - \varepsilon$

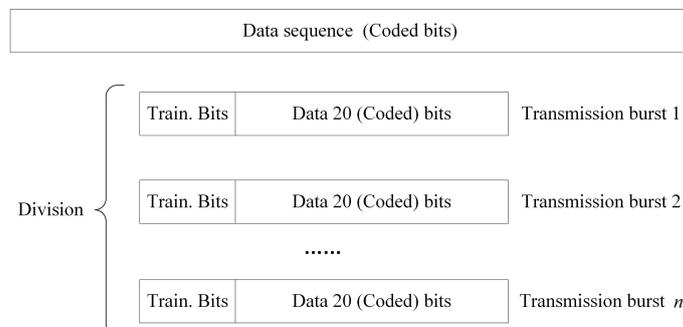
## 4. Simulation Results

### 4.1. Simulation Setup and Parameters

For the evaluation of the proposed blind turbo equalization, the parameters are summarized in Table 2, and the burst structure is shown in Figure 4. CPM with  $M = 4$ ,  $L = 2 - RC$ ,  $h = 1/2$  was taken as an example for the experiment and simulation. CPM signals of each burst were sent over 4-tap equal power Rayleigh fading channels. In the first case, the single burst structure is shown in Figure 4a, with the training sequence using only a small number of bits to lock quickly and reliably for the receiver rather than for channel estimation in the conventional one. A time-invariant channel is used throughout a burst when the relative velocity of the transmitter and receiver is not very high, and the channel in any two different bursts is independent. In the second case, we considered a time-variant channel. The data bits were coded and divided into multiple subsequences as shown in Figure 4b, where each block of 20 code bits with a training sequence was transmitted as a short burst on a different channel.



(a) Single short burst.



(b) Data sequence divided into multiple short bursts.

**Figure 4.** Short burst structure. (a) Single transmission burst at the output of the training insertion (TI) block with 66 payload coded bits. (b) Data sequence (in bits) is divided into  $n$  sequences of 20 bits, where each sequence was transmitted as a burst on a different channel.

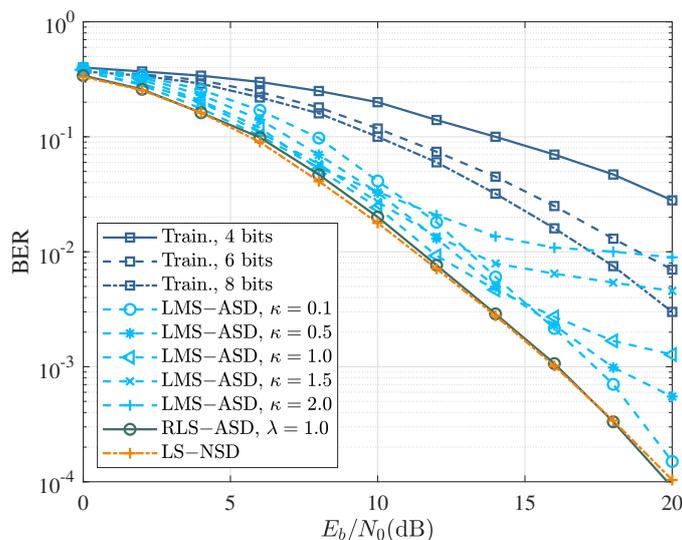
**Table 2.** Basic simulation parameters.

Parameters	Value	Remarks
Frequency pulse	Raised-cosine pulse	$L = 2$
Modulation order (M)	4	-
Modulation index (h)	1/2	$h = p/q$
Mapping	Gray	-
Training bits	4,6,8	ML estimation
Coderate	1/2	Convolutional code
Samples/Symbol	2	$\epsilon$
Multipath channel	Four-tap Rayleigh fading channels	Equal power

#### 4.2. Uncoded CPM: Single Burst and Time-Invariant Channel

In a single short burst, as shown in Figure 4a, the channel is assumed to be constant during a single burst and independent of each other. The uncoded performance of the different detectors in a time-invariant fading channel is shown in Figure 5. ‘Tran.’ curves represent the performance of the training-based method as the benchmark to beat. In a short burst, it is difficult for a few training data to provide an accurate estimation, resulting in poor performance. The LS-NSD curve is the performance of the symbol detector proposed as the baseline in [9], which served as the baseline of performance. In RLS-ASD, because of the time-invariant channel, RLS with forgetting factor  $\lambda = 1$  was applied to channel estimation in the branch metric. The performance of the RLS-ASD with 1 was roughly similar to that of LS-NSD, but with lower complexity.

Compared to LS-NSD and RLS-ASD, the LMS-based ASD had much lower complexity and practical feasibility, but its performance was strongly influenced by the step size. The LMS-ASD with fixed step sizes of  $\kappa = 0.1, 0.5, 1.0, 1.5, 2.0$  was selected according to Equation (30). BER curves show that the LMS-ASD with a greater  $\kappa$  achieved better BER performance than that of the smaller one in low  $E_b/N_0$ , but reached an error floor at high  $E_b/N_0$ . The results show that no error floor was observed for the LMS-ASD with  $\kappa = 0.1$ , while there is a significant loss of performance at low  $E_b/N_0$ .



**Figure 5.** BER as a function of  $E_b/N_0$  for different detectors. Curves Tran. and LS-NSD refer to the performance of training-based channel estimation and the symbol detection proposed in [9], respectively, representing the benchmark to beat.

#### 4.3. Coded CPM: Multiple Burst and Time-Variant Channel

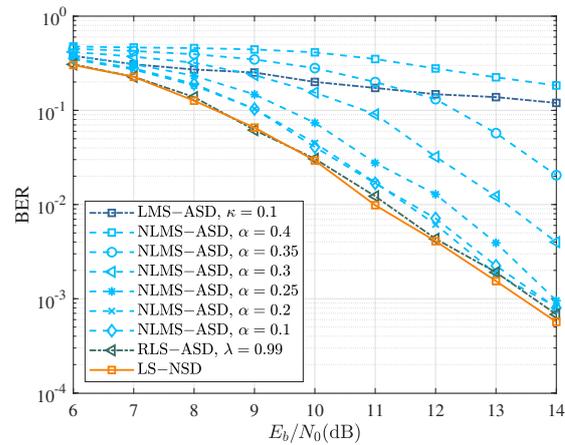
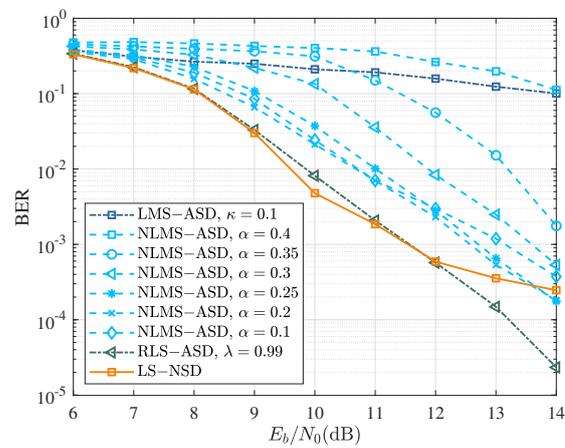
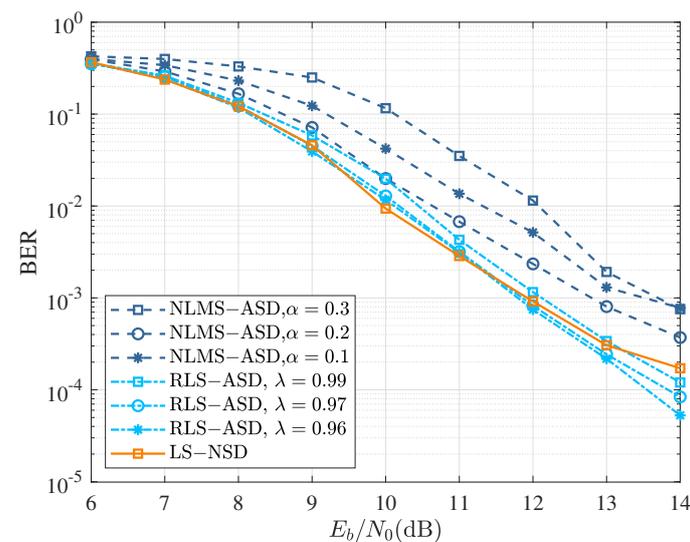
At a low Doppler spread with  $f_d T = 0.002$ , the BER performance of various symbol detectors for coded CPM is demonstrated in Figure 6. The performance of the four detectors without iteration is given in Figure 6a. The performance of the RLS-ASD with  $\lambda = 0.99$  was roughly similar to that of LS-NSD, but with lower complexity. BER curves show that the LMS-ASD achieved poorer BER performance than that of the NLMS-ASD since it utilized a fixed step size to achieve faster convergence, but ignored the steady-state misadjustment error. Therefore, NLMS-ASD had both faster convergence speed and a smaller misadjustment error. As the parameter  $a$  in Equation (31) was reduced, the performance of the NLS-ASD gradually improved, coming closer to that of RLS-ASD and LS-NSD.

After the third iteration, the comparison of the BER curves using various symbol detectors is shown in Figure 6. The curve of LS-ASD reached an error floor, while the BER of RLS-ASD had an obvious downward trend along with  $E_b/N_0$ . The performance of NLMS-ASD did not always increase with the decrease in parameter alpha, and the best performance was achieved with  $a = 0.2$ .

The comparisons of LS-NSD, NLMS-ASD, and RLS-ASD are shown for the third iteration in Figure 7 for  $f_d T = 0.01$ . At a high Doppler spread, LS-NSD could achieve slightly better performance than that of the one at a low Doppler spread since it obtained more gain from the diversity of the different channels with fast time variation. At the same time, as the Doppler spread increased, a smaller forgetting factor in RLS-ASD could achieve faster convergence. Then, NLMS-ASD with  $a = 0.2$  could achieve similar performance to that of LS-NSD. However, its complexity was lower than that of the LS-ASD or RLS-ASD. Thus, the NLMS-ASD with appropriate parameters could achieve an excellent trade-off between complexity and performance.

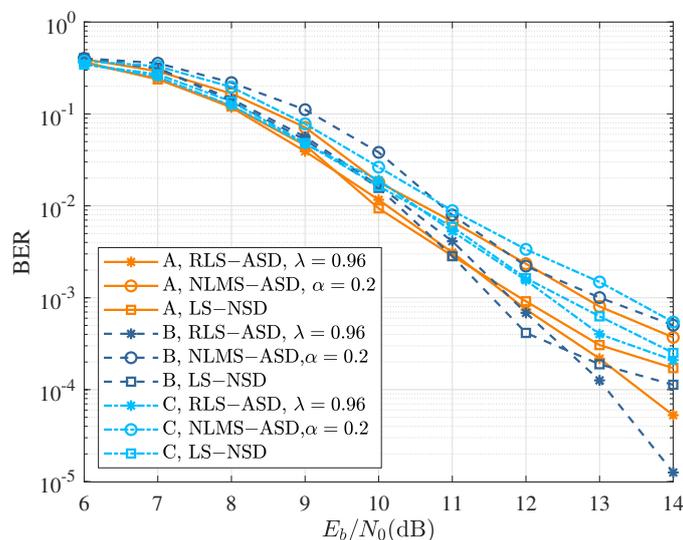
**Table 3.** Parameter of the coded and interleaver schemes; see also Table 2.

Scheme	Outer Code	Interleaver Length	Iterations
A	$R_c = 1/2, [5, 7]_8$	4831	3
B	$R_c = 1/2, [17, 15]_8$	4831	3
C	$R_c = 1/2, [5, 7]_8$	27,221	3

(a) BER as a function of  $E_b/N_0$  without iteration.(b) BER as a function of  $E_b/N_0$  after three iterations.**Figure 6.** BER performance of various turbo symbol detectors at  $f_d T = 0.002$  for Scheme A in Table 3.**Figure 7.** BER performance of various turbo symbol detectors after three iterations at  $f_d T = 0.01$  for Scheme A in Table 3.

Lastly, we consider the performance of various symbol detectors when different channel codes and interleavers are employed. The BER after three iterations at  $f_d T = 0.01$  for the

coded modulation schemes given in Tables 2 and 3 is shown in Figure 8. Compared with the benchmark of Scheme A, the performance could be improved with the convolutional codes with a larger number of states in Scheme B, especially for RLS-ASD. However, the longer interleaving length in Scheme C did not significantly improve performance, which may have been due to the smaller frame length of the signal.



**Figure 8.** BER performance of various turbo symbol detectors after three iterations at  $f_d T = 0.01$  for the coded modulation schemes given in Tables 2 and 3.

## 5. Conclusions

The paper proposed a class of symbol-by-symbol detectors based on the BCJR algorithm. The channel estimator was embedded in the forward recursion of the BCJR algorithm, and we illustrated the LS-NSD in [9] as an instance of the proposed detector. First, the performance of RLS-ASD and LMS-ASD with different step sizes was compared for a single CPM burst without coding and a time-invariant channel, using LS-NSD as a benchmark. RLS-ASD with  $\lambda = 1$  achieved similar performance to that of LS-NSD because RLS is a fast implementation of LS. LMS-ASD approached the benchmark as the step size decreased while reducing the FLOP number by about half.

In the second case, we considered multiple bursts with time-variant channels in the form of SCCPM. LS-NSD showed an error floor with different degrees at different maximal normalized Doppler spreads, while RLS-ASD achieved a performance improvement of 1 dB, which showed no or a light error floor. NLMS-ASD with appropriate parameters reduced the required number of FLOP by 45% at the expense of 1 dB power. Lastly, the performance of different code and interleave schemes are observed. Convolutional codes with a larger number of states could achieve more performance gains, most notably with RLS-ASD, while a larger interleaving depth could reduce performance.

For the short-burst CPM, in the case of time-invariant channels, the choice of different detectors could achieve a flexible trade-off between complexity and performance when the LS-NSD was a benchmark. In the case of time-variant channels, the proposed detector was more robust to time-variant environments than LS-NSD was. Furthermore, the limited computing capacity in IoT nodes motivates exploring other methods for reduced-complexity detection, which further reduces the amount of calculation. In addition, research on the influence of synchronization on our proposed method is practically significant, since perfect synchronization was assumed in the front end of our article. The mentioned open points are left for future research on the CPM receiver.

Moreover, in the future, 6G wireless networks would not be limited to only terrestrial communication, but would also be supplemented by nonterrestrial communication, especially the application of UAVs. In UAV-aided IoT communication systems, UAVs can

provide more opportunities for line-of-sight paths and improved coverage range due to their flexible deployment and controlled mobility. In addition, a physical-layer innovation technique, intelligent reflective surfaces (IRS) [34–36], can reconfigure a wireless propagation environment by intelligently reflecting signals from the transmitter to the receiver, which can improve the performance of networks but remains unexplored. The design of the receiver with the assistance of UAVs and IRSs is a natural future research topic. The mentioned open points are left for future research on the CPM receiver.

As future work, there are three prospective directions: (1) searching for another method of reduced-complexity detection to further achieve a balance between complexity and performance; (2) studying the synchronization techniques of CPM, and analyzing the interplay between detection and synchronization; (3) exploring the design of CPM receivers with the assistance of UAVs and IRSs.

**Author Contributions:** Writing—original draft preparation, Z.P.; writing—review and editing, H.W., B.Z. and D.G.; visualization, Z.P.; methodology, D.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** the authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

SISO	Soft input–soft output
PSP	Per-survivor processing
CPM	Continuous-phase modulation
PA	Power amplifiers
HMM	Hidden Markov model
FAFB-SISO	Forward backward adaptive SISO
FA-SISO	Forward adaptive SISO
RS-A-SISO	Reduced state A-SISO
LLR	Log-likelihood ratio
CPE	Continuous phase encoder
MM	Memoryless modulator
AWGN	Additive white Gaussian noise
CSI	Channel state information
LS-NSD	Least-squares noncoherent symbol detector
RLS-ASD	Recursive least-squares adaptive symbol detector
LMS-ASD	Least-mean-squares adaptive symbol detector
IRS	Intelligent reflective surfaces

## Appendix A. Complexity

The complexity of different detectors is measured in terms of the number of floating-point operations (FLOP). The number of operations for multiplying an  $m - by - n$  matrix with an  $n - by - l$  is  $2mnl - ml$ , including the  $mnl$  multiplications and the  $m(n - 1)l$  additions. First, the number of operations required for RLS-ASD and LMS-ASD to calculate a single branch metric is analyzed as follow.

The state and input of RLS-ASD at time  $n$  is denoted as  $\sigma_n$  and  $a_n$ , respectively, which binds the  $\mathbf{h}(\sigma_n)$  and  $P(\sigma_n)$ . The  $\mathbf{X}(a_n)$  in Equation (8) can be represented as

$$\mathbf{X}(a_n) = \begin{bmatrix} \mathbf{X}(a_{n-1}) \\ \boldsymbol{\theta}_n \end{bmatrix}, \boldsymbol{\theta}_n \in \mathbb{C}^{\epsilon \times l} \quad (\text{A1})$$

and the received signal  $\mathbf{y}_n$  is represented as

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{y}_{n-1} \\ \boldsymbol{\tau}_n \end{bmatrix}, \boldsymbol{\tau}_n \in \mathbb{C}^{\epsilon \times 1} \tag{A2}$$

For ease of derivation, we denote  $\boldsymbol{\theta}_n$  as  $\boldsymbol{\theta}_n = (\mathbf{x}(t_{(n-1)\epsilon}), \dots, \mathbf{x}(t_{n\epsilon-1}))$  and denote  $\boldsymbol{\tau}_n$  as  $\boldsymbol{\tau}_n = (r_{(n-1)\epsilon}, \dots, r_{n\epsilon-1})$ , where  $\mathbf{x}(t_k)$  is defined as  $\{x_{k-i}\}_{i=0}^{l-1}$ .

The gain vector for the branch at time  $n$  is given by

$$\mathbf{K}_{(n-1)\epsilon+i} = \frac{\frac{1}{\lambda} \mathbf{P}_{(n-1)\epsilon+i-1} \mathbf{x}(t_{(n-1)\epsilon+i})}{1 + \frac{1}{\lambda} \mathbf{x}(t_{(n-1)\epsilon+i})^T \mathbf{P}_{(n-1)\epsilon+i-1} \mathbf{x}(t_{(n-1)\epsilon+i})}, \tag{A3}$$

and the number of operations in Equation (A3) is  $2l^2 + l - 1$ .

Then, we update P Matrix by Equation (27)

$$\mathbf{P}_{(n-1)\epsilon+i} = \frac{1}{\lambda} \mathbf{P}_{(n-1)\epsilon+i-1} - \frac{1}{\lambda} \mathbf{K}_{(n-1)\epsilon+i} \mathbf{x}(t_{(n-1)\epsilon+i})^T \mathbf{P}_{(n-1)\epsilon+i-1}. \tag{A4}$$

and the number of operations in Equation (A4) is  $4l^2 - 1$ .

Last, calculating the channel response by using Equations (A3) and (A4)

$$\mathbf{h}_{(n-1)\epsilon+i} = \mathbf{h}_{(n-1)\epsilon+i-1} - \mathbf{K}_{(n-1)\epsilon+i} (\mathbf{x}(t_{(n-1)\epsilon+i})^T \mathbf{h}_{(n-1)\epsilon+i-1} - y_{(n-1)\epsilon+i}) \tag{A5}$$

which has  $3l$  operations.

The total number of operations the channel estimation for RLS-ASD is  $6\epsilon l^2 + 2\epsilon l - \epsilon$ . Then, given the chance, the number of operations for branch metric in Equation (13) at time  $n$  is  $2n\epsilon l$ . Therefore, the number of operations needed for the calculation of a single branch metric at time  $n$

$$N_{FLOP}(\Lambda_{RLS-ASD}) = 6\epsilon l^2 + 2n\epsilon l + 2\epsilon l - \epsilon. \tag{A6}$$

Similarly, the calculation of a single branch metric for LMS-ASD and NLMS-ASD is

$$N_{FLOP}(\Lambda_{LMS-ASD}) = 2n\epsilon l + 3\epsilon l. \tag{A7}$$

$$N_{FLOP}(\Lambda_{NLMS-ASD}) = 2n\epsilon l + 5\epsilon l - \epsilon. \tag{A8}$$

For each detector there is a similar forward and backward recursions in Equations (20) and (22), which can be represented as

$$N_{FLOP}(\beta_{n+1}^f) = N_{FLOP}(\beta_n^b) = 3M - 1. \tag{A9}$$

In the stage of calculating a posteriori probabilities for symbol  $a_n = u$ , we obtain

$$N_{FLOP}(P(a_n = u)) = 4M^L - 1. \tag{A10}$$

Finally, the total number of operations needed by the different detector can be obtained

$$N_{FLOP}(RLS - ASD) = M^{L+1} \sum_{n=1}^N (6\epsilon l^2 + 2n\epsilon l + 2\epsilon l - \epsilon) + 2NM^L(3M - 1) + MN(4M^L - 1). \tag{A11}$$

$$N_{FLOP}(LMS - ASD) = M^{L+1} \sum_{n=1}^N (2n\epsilon l + 3\epsilon l) + 2NM^L(3M - 1) + MN(4M^L - 1). \tag{A12}$$

$$N_{FLOP}(NLMS - ASD) = M^{L+1} \sum_{n=1}^N (2n\epsilon l + 5\epsilon l - \epsilon) + 2NM^L(3M - 1) + MN(4M^L - 1). \tag{A13}$$

## References

1. You, X.; Wang, C.X.; Huang, J.; Gao, X.; Zhang, Z.; Wang, M.; Huang, Y.; Zhang, C.; Jiang, Y.; Wang, J.; et al. Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts. *Sci. China Inf. Sci.* **2021**, *64*, 64. [[CrossRef](#)]
2. Jiang, H.; Mukherjee, M.; Zhou, J.; Lloret, J. Channel modeling and characteristics analysis for 6G wireless communications. *IEEE Netw.* **2021**, *35*, 296–303. [[CrossRef](#)]
3. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Niyato, D.; Dobre, O.; Poor, H.V. 6G Internet of Things: A Comprehensive Survey. *IEEE Internet Things J.* **2022**, *9*, 359–383. [[CrossRef](#)]
4. Jiang, H.; Zhang, Z.; Xiong, B.; Dang, J.; Wu, L.; Zhou, J. A 3D Stochastic Channel Model for 6G Wireless Double-IRS Cooperatively Assisted MIMO Communications. In Proceedings of the 2021 13th International Conference on Wireless Communications and Signal Processing (WCSP), Changsha, China, 20–22 October 2021.
5. Ahmed, M.; Khan, W.U.; Ihsan, A.; Li, X.; Li, J.; Tsiftsis, A.T. Backscatter Sensors Communication for 6G Low-Powered NOMA-Enabled IoT Networks Under Imperfect SIC. *IEEE Syst. J.* **2022**. [[CrossRef](#)]
6. Jameel, F.; Khan, W.U.; Shah, S.T.; Ristaniemi, T. Towards Intelligent IoT Networks: Reinforcement Learning for Reliable Backscatter Communications. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019.
7. Khan, W.U.; Liu, J.; Jameel, F.; Khan, M.T.R.; Ahmed, S.H.; Jäntti, R. Secure Backscatter Communications in Multi-Cell NOMA Networks: Enabling Link Security for Massive IoT Networks. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020.
8. Khan, W.U.; Ihsan, A.; Nguyen, T.N.; Ali, Z.; Javed, M.A. NOMA-Enabled Backscatter Communications for Green Transportation in Automotive-Industry 5.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7862–7874. [[CrossRef](#)]
9. Chamaa, M.E.; Lankl, B. Noncoherent Symbol Detection of Short CPM Bursts in Frequency-Selective Fading Channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 771–782. [[CrossRef](#)]
10. Darsena, D.; Gelli, G.; Iudice, I.; Verde, F. Equalization Techniques of Control and Non-Payload Communication Links for Unmanned Aerial Vehicles. *IEEE Access.* **2018**, *6*, 4485–4496. [[CrossRef](#)]
11. Chayot, R.; Thomas, N.; Poulliat, C.; Boucheret, M.L.; Lesthient, G.; Van Wambeke, N. A New Exact Low-Complexity MMSE Equalizer for Continuous Phase Modulation. *IEEE Commun. Lett.* **2018**, *22*, 2218–2221. [[CrossRef](#)]
12. Aulin, T.; Sundberg, C. Continuous Phase Modulation—Part I: Full Response Signaling. *IEEE Trans. Commun.* **1981**, *29*, 196–209. [[CrossRef](#)]
13. Moqvist, P.; Aulin, T.M. Serially concatenated continuous phase modulation with iterative decoding. *IEEE Trans. Commun.* **2001**, *49*, 1901–1915. [[CrossRef](#)]
14. Ho, P.; Kim, J.H. Pilot symbol-assisted detection of CPM schemes operating in fast fading channels. *IEEE Trans. Commun.* **1996**, *44*, 337–347. [[CrossRef](#)]
15. Brown, C.; Vigneron, P.J. Signal recovery for CPM in frequency flat fast fading channels. In Proceedings of the 2012 IEEE Military Communications Conference, Orlando, FL, USA, 29 October–1 November 2012.
16. Chamaa, M.E.; Lankl, B. Turbo-Estimation for CPM Over Frequency-Selective Fast Fading Channel. *IEEE Trans. Commun.* **2020**, *68*, 2538–2550. [[CrossRef](#)]
17. Tan, J.; Stuber, G.L. Frequency-domain equalization for continuous phase modulation. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 2479–2490. [[CrossRef](#)]
18. Pancaldi, F.; Vitetta, G.M. Equalization algorithms in the frequency domain for continuous phase modulations. *IEEE Trans. Commun.* **2006**, *54*, 648–658. [[CrossRef](#)]
19. Ozgul, B.; Koca, M.; Delic, H. Double Turbo Equalization of Continuous Phase Modulation with Frequency Domain Processing. *IEEE Trans. Commun.* **2009**, *57*, 423–429. [[CrossRef](#)]
20. Thillo, W.V.; Horlin, F.; Ramon, V.; Bourdoux, A.; Lauwereins, R. Novel block constructions using an intrafix for CPM with frequency domain equalization. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 951–955. [[CrossRef](#)]
21. Han, R.; Jia, N.; Guo, Z.; Huang, J.; Xiao, D.; Guo, S. Prefiltered Single-Carrier Frequency-Domain Equalization for Binary CPM over Shallow Water Acoustic Channel. *Sensors* **2022**, *22*, 3821. [[CrossRef](#)] [[PubMed](#)]
22. Neugebauer, S.P.; Ding, Z. Blind SIMO channel estimation for CPM using the Laurent approximation. In Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (ISCAS), Vancouver, BC, Canada, 23–26 May 2004.
23. Kammeyer, K.D.; Kuhn, V.; Petermann, T. Blind and nonblind turbo estimation for fast fading GSM channels. *IEEE J. Sel. Areas Commun.* **2001**, *19*, 1718–1728. [[CrossRef](#)]
24. Anton-Haro, C.; Fonollosa, J.A.R.; Fonollosa, J.R. Blind Channel Estimation and Data Detection Using Hidden Markov Models. *IEEE Trans. Signal. Process.* **1997**, *45*, 241–247. [[CrossRef](#)]
25. Cirpan, A.H.; Tsatsanis, M.K. Blind Receivers for Nonlinearly Modulated Signals in Multipath. *IEEE Trans. Signal. Process.* **1999**, *47*, 583–586. [[CrossRef](#)]
26. Nguyen, H.; Levy, B.C. Blind and Semi-Blind Equalization of CPM Signals With the EMV Algorithm. *IEEE Trans. Signal. Process.* **2003**, *51*, 2650–2664. [[CrossRef](#)]
27. Pan, Z.; Xie, C.; Wang, H.; Wei, Y.; Guo, D. Blind Turbo Equalization of Short CPM Bursts for UAV-Aided Internet of Things. *Sensors* **2022**, *22*, 6508. [[CrossRef](#)] [[PubMed](#)]

28. Raheli, R.; Polydoros, A.; Tzou, C.-K. Per-Survivor Processing: A general approach to MLSE in uncertain environments. *IEEE Trans. Commun.* **1995**, *43*, 354–364. [[CrossRef](#)]
29. Anastasopoulos, A.; Mu, C.K. Adaptive soft-input soft-output algorithms for iterative detection with parametric uncertainty. *IEEE Trans. Commun.* **2000**, *48*, 1638–1649. [[CrossRef](#)]
30. Hansson, A.; Chugg, K.M.; Aulin, T. On forward-adaptive versus forward/backward-adaptive SISO algorithms for Rayleigh fading channels. *IEEE Commun. Lett.* **2001**, *5*, 477–479. [[CrossRef](#)]
31. Chung, K.; Chugg, K.M.; Heo, J. Reduced state adaptive SISO algorithms for serially concatenated CPM over frequency-selective fading channels. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), San Antonio, TX, USA, 25–29 November 2001.
32. Robertson, P.; Villebrun, E.; Hoeher, P. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In Proceedings of the Proceedings IEEE International Conference on Communications ICC '95, Seattle, WA, USA, 18–22 June 1995.
33. Sayed, A.H.; Kailath, T. A state-space approach to adaptive RLS filtering. *IEEE Signal. Proc. Mag.* **1994**, *11*, 18–60. [[CrossRef](#)]
34. Khan, W.U.; Lagunas, E.; Ali, Z.; Javed, M.A.; Ahmed, M.; Chatzinotas, S.; Ottersten, B.; Popovski, P. Opportunities for Physical Layer Security in UAV Communication Enhanced with Intelligent Reflective Surfaces. *arXiv* **2022**, arXiv:2203.16907.
35. Xiong, B.; Zhang, Z.; Jiang, H.; Zhang, J.; Wu, L.; Dang, J. A 3D non-stationary MIMO channel model for reconfigurable intelligent surface auxiliary UAV-to-ground mmWave communications. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 5658–5672. [[CrossRef](#)]
36. Wu, Q.; Guan, X.; Zhang, R. Intelligent Reflecting Surface-Aided Wireless Energy and Information Transmission: An Overview. *Proc. IEEE* **2022**, *110*, 150–170. [[CrossRef](#)]