

A Novel Semi-blind Receiver Algorithm Based on PARATUCK2 Model in MIMO Relay Communication System

Wang Rui, Rui Guo-Sheng, Zhang Yang, and Xue Peng

Abstract—Most receiver algorithms based on PARATUCK2 model adopted traditional ALS methods in one way and two hops MIMO relay communication system now days. Its low convergence speed and high computational cost are main factors restricting its application under real-time conditions. For the reason, a novel blind receiver Algorithm based on PARATUCK2 model is proposed based on works of building one way and two hops MIMO relay communication system. Firstly, the algorithm adds received data a time dimensionality on the premise of no supervise sequence, then chooses truncated window reasonable, creates iteration relationship between auxiliary matrix and base matrix of right subspace, corrects the channel parameter adaptively, and make it closer to true parameters in least error square sense. Computational cost of the algorithm is reduced by avoiding Moore Penrose pseudo-inverse operation efficiently through tracking and iterating the singular of matrix. The algorithm also improves convergence speed by utilizing the estimated result of previous moment. Simulation result shows that the algorithm has lower computational cost, higher convergence speed than PT2-ALS, and the same estimation performance with ALS premised on that no supervised sequence added.

Index Terms—MIMO, relay system, semi-blind, PARATUCK2, ALS.

I. INTRODUCTION

As the popularization of wireless communication media and interactive application service, Requirement of high rate transmission and high reliability for mobile user is becoming more and more urgent. Therefore, MIMO relay communication system draws attention widely, which is absolutely a promising research field [1], [2].

Generally, CSI (Channel State Information) is of importance for transmitter optimization and symbol detection [2]. Traditional methods use supervised train sequence to obtain Channel matrix, in which the Channel estimation is supervised or no blind. On the one hand, the train sequence decreases the spectrum efficiency; on the other hand, it is difficult to obtain accurate CSI when channel coefficient varies fast. Blind receiver algorithm can accomplish channel estimation and symbol detection without any redundant information except the received data. The semi-blind receiver algorithm only needs the code matrix and relay gain matrix,

which are available in engineering application. The proposed novel receiver algorithm is based on the semi-blind condition.

Some researchers had found that tensor analysis is extraordinary efficient tool for channel estimation and symbol detection [3], [4]. The supervised algorithm was proposed to estimate one way AF (Amplify and Forward) channel coefficient in [5], [6]. The ALS (Alternating Least Squares) based synchronous estimation of one way partial channel was proposed in [7], which needs the strict identifiability conditions. L.R. Ximenes proposed a semi-blind receiver algorithm which is based on PARATUCK2 (PARAFAC TUCKer2) model. The algorithm used KRST (Khatri-Rao Space Time) coding to improve diversity and multiplex gain at source, and amplify and re-transmit the signal at relay, built received signal to be a PARATUCK2 tensor model at destination, and used ALS based algorithm to the channel estimation and symbol detection. In addition, the algorithm also used the source-destination channel to choose initial value to decrease compute complexity. However, the compute complexity is undesirable when network scale is large.

It can be concluded that the semi-blind receiver algorithms, which is based on PARAFAC extended model, mostly adopt ALS based receiver algorithm. Because of its high compute complexity and less convergence speed, it is unsuitable for real-time application. So the PT2-AT (ParaTuck2 Adaptive Tracking) is proposed based on the works in [8]. Firstly, the algorithm adds a received data time dimensionality on the premise of no supervise sequence, then chooses truncated window reasonable, creates iteration relationship between auxiliary matrix and base matrix of right subspace, corrects the channel parameter adaptively, and makes it closer to true parameters in least error square sense. Computational cost of the algorithm is reduced by avoiding Moore Penrose pseudo-inverse operation efficiently through tracking and iterating the singular of matrix. The algorithm also improves convergence speed by utilizing the estimated result of previous moment. Simulation result shows that the algorithm has lower computational cost, higher convergence speed, and the same estimation performance with ALS premised on that no supervised sequence added.

Section II introduces MIMO relay communication model. Section III describes the basic idea of the algorithm, and the research content. Section IV contains the choice of truncated window, and algorithm flow. Section V contains simulation of the algorithm, and analysis of the simulation.

Notation: A bold-face lower-case letter \mathbf{x} denotes a vector, and a bold-face capital letter \mathbf{X} a matrix. A high order tensor of size $I \times J \times K$ is denoted by a

Manuscript received September 26, 2016; revised December 24, 2016. This work was supported by the Natural Science Foundation of China under Grant No.41606117 and National High Technology Research and Development Program of China under grant No.2015AA7015087.

The authors are with Naval Aeronautical Engineering Institute, Yantai, 264001, China (e-mail:sinoeagle2008@gmail.com).

calligraphic $\mathcal{X} \in \mathbb{C}^{J \times K}$. The transpose, complex conjugate transpose and pseudo-inverse are denoted by \mathbf{X}^T , \mathbf{X}^H , \mathbf{X}^+ respectively. $\|\mathbf{X}\|$ is the Frobenius norm of \mathbf{X} , the Khatri-Rao product is denoted by \odot . $\text{vec}(\mathbf{X})$ is a vector built by stacking the columns of \mathbf{X} one above each other.

II. SYSTEM MODEL

The basic principle of one-way and two-hop MIMO relay communication is illustrated in Fig. 1 [9], where M_S , M_R , M_D denote the number of antennas at the source (S), relay (R) and destination (D) nodes, respectively.

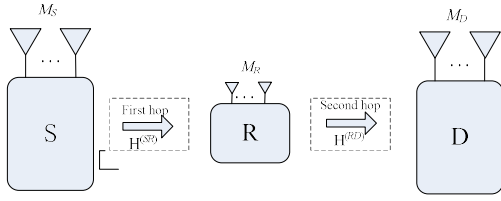


Fig. 1. One-way and two-hop MIMO relay communication system.

The source - relay channel and the relay- destination channel are assumed to be flat gauss channel, and constant during the transmission protocol. The transmission process is divided into two hops. During the first one, each source antenna transmits the signal after KRST coding; during the second one, the source antenna keep silent, and the relay will amplify the received signal to the destination.

Assume that the source antenna and relay antenna are independent. The nodes can be regarded as a single terminal with multiple antennas, or multiple terminals with single antenna. The system of Fig. 1 can be suited to many applications. For instance, the scene is corresponding to cellular mobile communication, which is high centralized relay network. Large number of mobile users share several relay to transmit the signal to base station. The scene of is corresponding to Ad hoc network, in which point to point communication needs many relay nodes. The source model and the relay channel model are described as follows.

A. The Source Model

Symbol matrix $\mathbf{S} \in \mathbb{C}^{N \times M_S}$ is coded by the KRST coding matrix $\mathbf{C} \in \mathbb{C}^{P \times M_S}$, in which N is the number of bit stream of each symbol. Time redundancy is generated through KRST coding, in which P is the reduplicate number of each bit. So each symbol period is extended to P times symbol period. The coded signal is given by

$$\tilde{\mathbf{S}}_{M_S \times PN} = (\mathbf{S} \odot \mathbf{C})^T \quad (1)$$

In which \odot is Khatri-Rao operator. $\tilde{\mathbf{S}}_{M_S \times PN}$ can be regarded as the mode-1 expand of virtual tensor $\tilde{\mathcal{S}} \in \mathbb{C}^{M_S \times P \times N}$. The columns of $\tilde{\mathbf{S}}_{M_S \times PN}$ is the signal transmitted during first hop, which is transmitted through radio frequency circuit,

such as DAC, RF modulation module. Then transmit it to the relay, and to the destination through the relay lastly. The transmission scheme of the source is illustrated in Fig. 2.

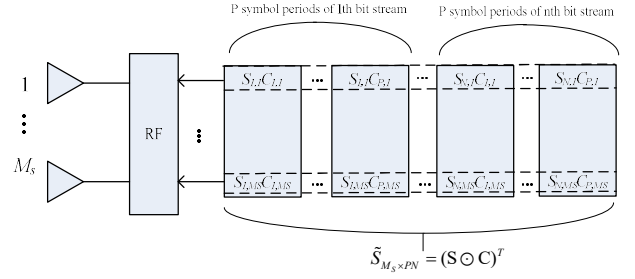


Fig. 2. Transmission scheme of the source.

B. The Relay Channel Model

The relay channel model is illustrated as follows.

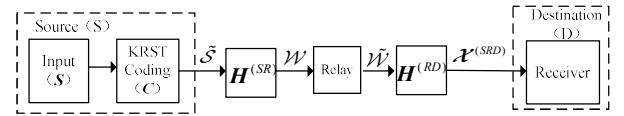


Fig. 3. The relay channel model.

The received signal by the relay through the channel $\mathbf{H}^{(SR)}$ transmission is given by

$$\mathbf{W}_{M_R \times PN} = \mathbf{H}^{(SR)} \tilde{\mathbf{S}}_{M_S \times PN} = \mathbf{H}^{(SR)} (\mathbf{S} \odot \mathbf{C})^T \quad (2)$$

Matrix $\mathbf{W}_{M_R \times PN}$ denotes mode-1 expansion of tensor $\mathcal{W} \in \mathbb{C}^{M_R \times P \times N}$, and its factor matrix are $(\mathbf{H}^{(SR)}, \mathbf{C}, \mathbf{S})$, which satisfy the condition of PARAFAC model. The received signal $\mathbf{W}_{M_R \times PN}$ through the channel $\mathbf{H}^{(SR)}$ is amplified by the gain matrix $\mathbf{G} \in \mathbb{C}^{N \times M_R}$ at the relay. The front slice of amplified tensor $\tilde{\mathcal{W}}$ is illustrated in Fig. 4.

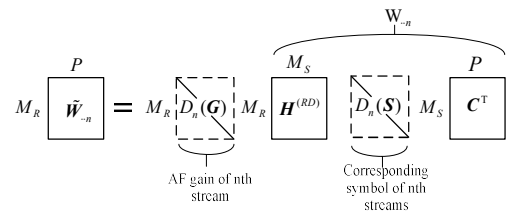


Fig. 4. The process of the relay.

In Fig. 4, $D_n(\mathbf{G})$ is diagonal matrix of the column of gain matrix \mathbf{G} , which denotes the amplify gain of the n th bit stream. The row number is at the left of frame, and the column number is at the right of frame.

The received signal at destination is a 3-order tensor $\mathcal{X}^{(SRD)} \in \mathbb{C}^{M_D \times P \times N}$, whose n th slice is given by

$$\begin{aligned} \mathbf{X}_{-n}^{(SRD)} &= \mathbf{H}^{(RD)} \tilde{\mathbf{W}}_{-n} \in \mathbb{C}^{M_D \times P} \\ &= \mathbf{H}^{(RD)} \mathbf{H}^{(SR)} D_n(\mathbf{S}) D_n(\mathbf{G}) \mathbf{C}^T \end{aligned} \quad (3)$$

From equation above, we can obtain the factor of $\mathcal{X}^{(SRD)}$

PARATUCK2 decomposition. $\mathbf{H}^{(RD)}$ and \mathbf{C} are factor matrix, $\mathbf{H}^{(SR)}$ is core matrix, \mathbf{G} and \mathbf{S} are interactive matrix. Correspondingly, the 3-mode expansion of $\mathcal{X}^{(SRD)}$ is given by

$$\mathbf{X}^{(3)} = \mathbf{X}_{M_D P \times N}^{SRD} = (\mathbf{I}_P \otimes \mathbf{H}^{(RD)}) \odot \begin{bmatrix} (\mathbf{C}_1 \otimes \mathbf{I}_{M_R}) \odot (\text{vec}(\mathbf{H}^{(SR)})) \\ \vdots \\ (\mathbf{C}_P \otimes \mathbf{I}_{M_R}) \odot (\text{vec}(\mathbf{H}^{(SR)})) \end{bmatrix} (\mathbf{S}^T \odot \mathbf{G}^T) \quad (4)$$

In which \otimes is Kronecker product, \mathbf{I}_P , \mathbf{I}_{M_R} is P order and MR order unit matrix. A novel receiver algorithm will be proposed based formulas 4 and ALS based algorithm, which combines PARATUCK2 model with SVD tracking algorithm.

III. BASIC IDEA

Under the condition described above, the semi-blind algorithm based ALS was proposed by article [8]. The algorithm updates a matrix at each step in LMS standard, and updates other matrix according the precious estimated result. Owing to the exhaustive method of ALS, and re-compute the factor matrix during each iteration, the algorithm has high compute complexity. We try to use the basic idea of adaptive idea to realize the tracking of factor matrix and fast convergence.

Formula (4) can be rewritten by

$$\mathbf{X}_{M_D P \times N}^{SRD} = (\mathbf{A} \odot \mathbf{F}) \mathbf{B}^T \quad (5)$$

In which $\mathbf{A} = (\mathbf{I}_P \otimes \mathbf{H}^{(RD)})$, $\mathbf{B}^T = (\mathbf{S}^T \odot \mathbf{G}^T)$,

$$\mathbf{F} = \begin{bmatrix} (\mathbf{C}_1 \otimes \mathbf{I}_{M_R}) \odot (\text{vec}(\mathbf{H}^{(SR)})) \\ \vdots \\ (\mathbf{C}_P \otimes \mathbf{I}_{M_R}) \odot (\text{vec}(\mathbf{H}^{(SR)})) \end{bmatrix}.$$

A time dimension is added to tensor $\mathcal{X}^{(SRD)} \in \mathbb{C}^{M_D \times P \times N}$, at the moment t, we have

$$\mathbf{X}^{(3)}(t) = \mathbf{H}(t) \mathbf{B}^T(t) \quad (6)$$

In which $\mathbf{H}(t) = \mathbf{A}(t) \odot \mathbf{F}(t)$, its dimension is $M_D P \times M_R M_S$. The dimension of $\mathbf{B}^T(t)$ is increasing with time. $\mathbf{X}^{(3)}(t)$ is the 3-mode expansion of $\mathcal{X}^{(SRD)}$.

The new observation tensor at the moment t+1 is $\mathcal{X}^{(SRD)}(t+1) \in \mathbb{C}^{M_D \times P \times N(t+1)}$, in which $N(t+1) = N(t) + 1$.

The 3-mode expansion of

$\mathcal{X}^{(SRD)}(t+1)$ is $\mathbf{X}^{(3)}(t+1) = \mathbf{H}(t+1) \mathbf{B}^T(t+1)$, in which $\mathbf{H}(t+1) = \mathbf{A}(t+1) \odot \mathbf{F}(t+1)$. In order to estimate the new $\mathbf{A}(t+1)$, $\mathbf{F}(t+1)$, $\mathbf{B}(t+1)$, ALS based algorithm can be used to make the estimation optimization. However, the algorithm needs Moore Penrose pseudo-inverse operation three times. Even that the initial value is selected, it also needs many iterations to converge.

$\mathbf{x}(t+1)$ is a new slice vector, whose dimension is $M_D P \times 1$, we have

$$\mathbf{X}^{(3)}(t+1) = [\mathbf{X}^{(3)}(t), \mathbf{x}(t+1)] \quad (7)$$

Considering channel varies slowly, i.e. $\mathbf{H}(t) \cong \mathbf{H}(t+1)$, we have

$$\mathbf{B}^T(t+1) \cong [\mathbf{B}^T(t), \mathbf{b}^T(t+1)] \quad (8)$$

Meaning that B has time-shift structure, the initial value of $\mathbf{b}(t+1)$ is given by

$$\mathbf{b}^T(t+1) = \mathbf{H}^+(t) \mathbf{x}(t+1) \quad (9)$$

According formula 8, $\mathbf{B}^T(t+1)$ can be constructed, and the LMS update of $\mathbf{H}(t+1)$ is given by

$$\mathbf{H}(t+1) = \mathbf{X}^{(3)}(t+1) (\mathbf{B}^T(t+1))^+ \quad (10)$$

Given $\mathbf{H}(t+1)$, $\mathbf{H}(t)$ can be substituted to update $\mathbf{b}^T(t+1)$. $\mathbf{H}(t+1)$ is the estimate of $\mathbf{A}(t+1) \odot \mathbf{F}(t+1)$, so the rth column of $\mathbf{A}(t+1) \odot \mathbf{F}(t+1)$ is given by

$$\mathbf{a}_r(t+1) \otimes \mathbf{f}_r(t+1) = \text{vec}(\mathbf{f}_r(t+1) \mathbf{a}_r^T(t+1)) \quad (11)$$

In which \mathbf{f}_r is the rth column vector of F, \mathbf{a}_r is the rth column vector of A. So $\mathbf{f}_r(t+1)$ can be estimated by product of the singular value of $\text{unvec}(\mathbf{f}_r(t+1) \mathbf{a}_r^T(t+1))$ and its left singular vector, and $\mathbf{a}_r(t+1)$ can be estimated by the conjugate of right singular vector until the R iterations. Lastly, $\mathbf{H}^{(SR)}(t+1)$ and $\mathbf{H}^{(RD)}(t+1)$ can be obtained by formula 5.

It should be mentioned that the basic idea is under the assumptions:

- 1) The system is constant or time varies slowly;
- 2) PARATUCK decomposition is unique at each moment;
- 3) R is known or estimated, and don't changes with time.

Though the basic idea of the algorithm is given above, several problems also exist:

- 1) Problem 1: compute complexity of the algorithm increases with time dimension increases;
- 2) Problem 2: pseudo-inverse operation results in compute

- complexity increasing during the iteration;
- 3) Problem 3: Traditional Svd Algorithm Results in Compute Complexity Increasing.
- The proposed algorithm will solve the problems above.

TABLE I: BASIC IDEA OF PARATUCK2 ADAPTING SEMI-BLIND ALGORITHM

Step 1
Assume $\mathbf{H}(t) \cong \mathbf{H}(t+1)$, obtain first estimation of $\mathbf{b}(t+1)$:
$\mathbf{b}^T(t+1) = \mathbf{H}^+(t)\mathbf{x}(t+1)$
Step 2
Obtain the first estimation of $\mathbf{D}_n(\mathbf{S}(t+1))$:
$\mathbf{B}^T(t+1) \cong [\mathbf{B}^T(t), \mathbf{b}^T(t+1)]$
Step 3
estimate $\mathbf{H}(t+1)$: $\mathbf{H}(t+1) = \mathbf{X}^{(3)}(t+1)(\mathbf{B}^T(t+1))^+$
Step 4
estimate $\mathbf{A}(t+1)$ and $\mathbf{F}(t+1)$ from $\mathbf{H}(t+1)$:
For $r=1$ to R , Do
$\mathbf{H}_r(t+1) = \text{unvec}([\mathbf{H}(t+1)]_{:,r})$
$[\mathbf{f}_r, \sigma_r, \mathbf{a}_r] = \text{SVD}(\mathbf{H}_r(t+1))$
$[\mathbf{F}(t+1)]_{:,r} = \sigma_r \mathbf{f}_r$ and $[\mathbf{A}(t+1)]_{:,r} = (\mathbf{a}_r)^*$
End
compute $\mathbf{H}^{(SR)}(t+1)$ 及 $\mathbf{H}^{(RD)}(t+1)$ by formula 5
Step 5
Re-estimate $\mathbf{B}(t+1)$:
$\mathbf{b}^T(t+1) = \mathbf{H}^+(t+1)\mathbf{x}(t+1)$
$\mathbf{B}^T(t+1) = [\mathbf{B}^T(t), \mathbf{b}^T(t+1)]$

IV. PT2-AT ALGORITHM

A. Pre-processing

In order to avoid the compute complexity increases with time dimension increases (problem 1), the algorithm add a sliding window to the observed tensor to estimate using limit observed data [10]. We choose the Truncated Window of length L ($L \geq R$). The last L columns of $\mathbf{X}^{(3)}(t+1)$ can be given by

$$\mathbf{X}_{TW}(t+1) = [\mathbf{x}(t+2-L), \dots, \mathbf{x}(t), \mathbf{x}(t+1)] \quad (12)$$

Define $\mathcal{X}_{TW}(t+1) \in \mathbb{C}^{M_D \times P \times L}$ tensor with new slice, $\mathbf{X}_{TW}(t+1)$ is the 3-mode expansion of the tensor. The PARATUCK2 decomposition of $\mathcal{X}_{TW}(t+1)$ can be estimated by the TWLS, i.e.

$$\min_{\mathbf{H}(t+1), \mathbf{B}_{TW}(t+1)} (\phi^{TW}(t+1)) \quad (13)$$

In which

$$\phi^{TW}(t+1) = \sum_{\tau=1}^L \lambda^{L-\tau} \|\mathbf{x}(u+\tau) - \mathbf{H}(t+1)\mathbf{b}^T(u+\tau)\|^2 \quad (14)$$

λ is a forgetting factor, $u = t+1-L$, $\mathbf{B}_{TW}(t+1)$ is

compose of the last L column of $\mathbf{B}(t+1)$. The window has different weight on the L slices at any moment. It is a rectangle sliding window when $\lambda = 1$, and exponential attenuation window when $\lambda < 1$.

Define the weighted matrix $\mathbf{X}_{TW}(t+1)$, which can be obtained from $\mathbf{X}_{TW}^{(3)}(t+1)$ i.e.

$$\mathbf{X}_{TW}(t+1) \stackrel{\text{def}}{=} \mathbf{X}_{TW}^{(3)}(t+1)\mathbf{A} \quad (15)$$

$\mathbf{A} = \text{diag}([\lambda^{(L-1)/2}, \lambda^{(L-2)/2}, \dots, \lambda^{1/2}, 1])$, \mathbf{X}_{TW} the update rule of is

$$[\lambda^{L/2}\mathbf{x}(t+1-L), \mathbf{X}_{TW}(t+1)] = [\lambda^{1/2}\mathbf{X}_{TW}(t), \mathbf{x}(t+1)] \quad (16)$$

Given the pre-processing observed data, we can construct PT2-AT (ParaTuck2 Adaptive Tracking) algorithm. For a truncated data $\mathbf{X}_{TW}(t)$, we can obtain from formula 5 and formula 15.

$$\mathbf{X}_{TW}(t) = (\mathbf{A}(t) \odot \mathbf{F}(t))\mathbf{B}_{TW}^T(t)\mathbf{A} \quad (17)$$

In addition, make SVD of $\mathbf{X}_{TW}(t)$, we have

$$\mathbf{X}_{TW}(t) = \mathbf{U}_{TW} \sum_{TW} \mathbf{V}_{TW}^H(t) \quad (18)$$

$\mathbf{U}_{TW}(t) \in \mathbb{C}^{M_D \times P \times R}$, $\sum_{TW}(t) \in \mathbb{C}^{R \times R}$, $\mathbf{V}_{TW}(t) \in \mathbb{C}^{L \times R}$. Under the assumptions, the rank of $\mathbf{X}_{TW}(t)$ is R . according formula 17 and 18, there exists a no singular matrix $\mathbf{W}_{TW}(t) \in \mathbb{C}^{R \times R}$. We can obtain

$$\begin{cases} \mathbf{A}(t) \odot \mathbf{F}(t) = \mathbf{E}_{TW}(t)\mathbf{W}_{TW}(t) \\ \mathbf{B}_{TW}^T(t)\mathbf{A} = \mathbf{W}_{TW}^{-1}(t)\mathbf{V}_{TW}^H(t) \end{cases} \quad (19)$$

In which $\mathbf{E}_{TW}(t) \stackrel{\text{def}}{=} \mathbf{U}_{TW} \sum_{TW}$. Auxiliary matrix $\mathbf{E}_{TW}(t) \stackrel{\text{def}}{=} \mathbf{U}_{TW} \sum_{TW}$ connects the two equations in formula 19. The purpose of PT2-AT is to update $\mathbf{W}_{TW}(t)$ and its inverse in recursion.

At the moment $t+1$, we have

$$\begin{cases} \mathbf{A}(t+1) \odot \mathbf{F}(t+1) = \mathbf{E}_{TW}(t+1)\mathbf{W}_{TW}(t+1) \\ \mathbf{B}_{TW}^T(t+1)\mathbf{A} = \mathbf{W}_{TW}^{-1}(t+1)\mathbf{V}_{TW}^H(t+1) \end{cases} \quad (20)$$

Owing to the time shift structure of $\mathbf{B}_{TW}^T(t+1)$ and $\mathbf{B}_{TW}^T(t)$, We can use the common data block to connect formula 19 and formula 20. Before give the update rule, $\mathbf{E}_{TW}(t+1)$ and $\mathbf{V}_{TW}(t+1)$ can be estimated from $\mathbf{E}_{TW}(t)$ and $\mathbf{V}_{TW}(t)$.

B. Algorithm Flow

Step 1: SVD tracking. It should be mentioned that both left and right subspace need tracking [11]. At the same time, the updating W needs the matrix V , so the right subspace vector bases need to be computed. The PT2-AT uses SWASVD algorithm to SVD tracking, which has low compute complexity and high reliability [12]. Compared with SWASVD, a forgetting factor λ is introduced in the truncated window processing. The detail about the tracking algorithm is not mentioned.

Step 2: update W and W^{-1} . The recursion of W and W^{-1} needs the right subspace base vector [13], which can be obtained from step 1, i.e.

$$\begin{cases} \mathbf{B}_{TW}^T(t)\mathbf{A}(t) = \mathbf{W}_{TW}^{-1}(t)\mathbf{V}_{TW}^H(t) \\ \mathbf{B}_{TW}^T(t+1)\mathbf{A}(t+1) = \mathbf{W}_{TW}^{-1}(t+1)\mathbf{V}_{TW}^H(t+1) \end{cases} \quad (21)$$

The dimension of $\mathbf{B}_{TW}(t)$ and $\mathbf{V}_{TW}(t)$ at each moment is $L \times R$, the common data block $\tilde{\mathbf{B}}(t)$ is given by

$$\tilde{\mathbf{B}}(t) = [\mathbf{B}_{TW}(t)]_{2:L,:} = [\mathbf{B}_{TW}(t+1)]_{1:L-1,:} \quad (22)$$

For convenience, we redefine:

$$\begin{cases} \tilde{\mathbf{V}}(t) = [\mathbf{V}_{TW}(t)]_{2:N,:} \\ \tilde{\mathbf{v}}(t) = [\mathbf{V}_{TW}(t)]_{1,:} \\ \tilde{\mathbf{V}}(t+1) = [\mathbf{V}_{TW}(t+1)]_{1:N-1,:} \\ \tilde{\mathbf{v}}(t+1) = [\mathbf{V}_{TW}(t+1)]_{N,:} \end{cases} \quad (23)$$

Substitute formula 22 and 23 into formula 21, we can obtain

$$\lambda^{1/2}\mathbf{W}^{-1}(t)\tilde{\mathbf{V}}^H(t) = \mathbf{W}^{-1}(t+1)\tilde{\mathbf{V}}^H(t+1) \quad (24)$$

Rewritten as

$$\mathbf{W}^{-1}(t+1) = \lambda^{1/2}\mathbf{W}^{-1}(t)\tilde{\mathbf{V}}^H(t)(\tilde{\mathbf{V}}(t+1))^+ \quad (25)$$

$$\mathbf{W}(t+1) = \lambda^{-1/2}\tilde{\mathbf{V}}^H(t+1)(\tilde{\mathbf{V}}^H(t))^+\mathbf{W}(t) \quad (26)$$

We can accomplish the updating of $\mathbf{W}(t+1)$ using the formula 25 and formula 26. It should be mentioned that there exists inverse operation in formula 26. For rank-one matrix updating, the matrix inverse theorem can be used, which is described in appendix. So the iteration of the inverse of V is given by

$$(\tilde{\mathbf{V}}^H(t+1))^+ = \tilde{\mathbf{V}}(t+1) \left(\mathbf{I}_R + \frac{\tilde{\mathbf{v}}^H(t+1)\tilde{\mathbf{v}}(t+1)}{1 - \|\tilde{\mathbf{v}}(t+1)\|} \right) \quad (27)$$

According the same theorem, we have

$$(\tilde{\mathbf{V}}^H(t))^+ = \tilde{\mathbf{V}}(t) \left(\mathbf{I}_R + \frac{\tilde{\mathbf{v}}^H(t)\tilde{\mathbf{v}}(t)}{1 - \|\tilde{\mathbf{v}}(t)\|} \right) \quad (28)$$

Substitute formula 27 and 28 into formula 25 and 26, we can obtain the update of W and W^{-1}

Step 3: update of A and F . \mathbf{E}_{TW} and \mathbf{W}_{TW} can be computed by step1 and step 2. $\mathbf{A}(t+1)$ and $\mathbf{F}(t+1)$ can be computed according step 4 in Table I. Then $\mathbf{H}^{(RD)}$ and $\mathbf{H}^{(SR)}$ can be obtained by C and G .

Step 4: update of B . The updating row $\mathbf{b}^H = \mathbf{W}^{-1}\tilde{\mathbf{v}}^H$. Combined with $\mathbf{B}(t)$, $\mathbf{B}(t+1)$ can be constructed, in which $\mathbf{W}^{-1}(t+1)$ can be computed by formula 25.

The proposed PT2-AT algorithm is summarized in Table II.

TABLE II: PT2-AT ALGORITHM SUMMARIZATION

input at the moment t :	
old estimated value :	
$A(t), B(t), F(t), W(t), E(t), V(t)$	
Observed data :	
$X^{(3)}(t), X^{(3)}(t+1) = [X^{(3)}(t), x(t+1)]$	
Truncated window : construct $X_{TW}(t+1)$	
Updating at the moment $t+1$	
Step 1 : SVD tracking of $X_{TW}(t+1)$	
Revised SWASVD algorithm by introducing a forgetting factor. Its multiply number is $(3M_D P + 3N)R^2 + 2M_D P R$	
Output :	
$U(t+1) \in C^{M_D P \times R}$, $\Sigma(t+1) \in C^{R \times R}$, $V(t+1) \in C^{N \times R}$	
Step 2 : update $W(t), W^{-1}(t)$	
Reconstruct matrix :	
$\tilde{V}(t) = [V_{TW}(t)]_{2:N,:}$, $\tilde{v}(t) = [V_{TW}(t)]_{1,:}$	
$\tilde{V}(t+1) = [V_{TW}(t+1)]_{1:N-1,:}$, $\tilde{v}(t+1) = [V_{TW}(t+1)]_{N,:}$	
update	Mul num
construct $Z = \lambda^{-1/2} \tilde{V}^H(t+1) \tilde{V}(t)$	NR^2
update W :	
$W(t+1) = Z \left(I_R + \frac{\tilde{v}^H(t) \tilde{v}(t)}{1 - \ \tilde{v}(t)\ } \right) W(t)$	$2R^3 + 2R^2 + R$
update W^{-1} :	
$W^{-1}(t+1) = W^{-1}(t) Z^H(t) \left(I_R + \frac{\tilde{v}^H(t+1) \tilde{v}(t+1)}{1 - \ \tilde{v}(t+1)\ } \right)$	$2R^3 + 2R^2 + R$
Step 3 : updating A and F	

C. Compute Complexity

Compute complexity depends upon iteration number and operator number in the iteration. For convenience, considering the multiply number in one iteration, the number of PT2-ALS [14] is given by

$$\begin{aligned} & 11R^3 + 8R^2(M_D N + PN + M_D P + 3) \\ & + 8R(3M_D PN + M_D N + PN + M_D P) \end{aligned} \quad (29)$$

The single step number of PT2-AT algorithm is listed in Table II. To sum up these numbers, the total number is given

by:

$$4R^3 + R^2(4M_D P + 4N + 5) + 2R(2M_D P + M_D + 1) \quad (30)$$

Apparently the operation of PT2-AT is lower than PT2-ALS. We will discuss the quantitative analysis under some conditions in Section IV. It should be mentioned that iteration number is affected by the initial value and channel condition. It is difficult to discuss it in quantitative analysis, which will discuss by simulation under certain condition in Section IV.

V. SIMULATION RESULT ANALYSIS

The section will evaluate the algorithm in two aspects of compute complexity and estimation performance. Its final aim is to evaluate compute and estimation performance of the symbol and channel.

Under conditions where CSI is known, symbol and channel matrix are estimated by the algorithm, and the estimation result is evaluated by SER (Symbol Error Rate) and channel NMSE (Normalized Mean Square Error). Channel NMSE is defined as

$$\text{NMSE} = \frac{1}{K} \left(\sum_{k=1}^K \frac{\|\mathbf{H}_k - \hat{\mathbf{H}}_k\|_F^2}{\|\mathbf{H}_k\|_F^2} \right) \quad (31)$$

In which K is the run number of Monte Carlo, $K = 10000$, $N=8$, $NKM_S = 80000M_S$. \mathbf{H}_k is virtual channel matrix during k th run. And $\hat{\mathbf{H}}_k$ is the estimated channel matrix after the algorithm processing. Here \mathbf{H} includes $\mathbf{H}^{(SR)}$ and $\mathbf{H}^{(RD)}$. Channel matrix $\mathbf{H}^{(SD)}$, $\mathbf{H}^{(RD)}$ and $\mathbf{H}^{(SR)}$ satisfy distribution of complex gauss channel, in which the mean is zero, and the variation is $1/M_R$, $1/M_S$, $10^{-\alpha/10}/M_S$ respectively, and α denotes the attention ratio between SD channel and SRD channel, which is a tune value. The noise satisfy the gauss distribution, whose mean is zero. It should be mentioned that the mean SNR of SD and SRD in direct proportion to the symbol energy.

Symbol matrix $\mathbf{S} = \sqrt{E_S} \mathbf{S}_0$ is generated from 8PSK alphabet table, in which \mathbf{S}_0 is symbol matrix with unit energy, E_S is mean energy of each symbol. First row of \mathbf{S} and $\mathbf{H}^{(RD)}$ is known, which eliminates the scale ambiguous. The coding matrix is truncated DFT matrix, which guarantees that matrix \mathbf{C} is full column rank. Gain matrix \mathbf{G} is a random generated vermode matrix, which avoids the permute ambiguous of $\mathbf{H}^{(RD)}$ and $\mathbf{H}^{(SR)}$.

The length of truncated window $L=8$, the forgetting factor $\lambda = 0.7$, and the dimension R is the same as the number of source antennas.

A. Compute Complexity

We use multiply operation number in the iteration and

iteration number to compare the compute complexity of PT2-ALS and PT2-AT, which reflect the compute quantity and convergence speed respectively.

Considering the analysis at the section IV, and setting the parameter in the Table III, the ratio of multiply operation number between PT2-ALS and PT2-AT is computed to evaluate compute quantity of the two algorithms. The result is illustrated in Fig. 5. The figure shows that multiply number of PT2-ALS is ten times more than PT2-AT. With R decreases, the advantage of low compute quantity of PT2-AT is more apparently. It should be mentioned that total operation number is in relation with N, M_D and P , not just only with R . however, considering the formula of section IV, the compute quantity of PT2-AT is lower than PT2-ALS apparently.

TABLE III: SIMULATION PARAMETER SETTING

$$M_R = M_D = 8, N=L=8, P=4; \\ \text{ALS convergence error: } 10^{-3}$$

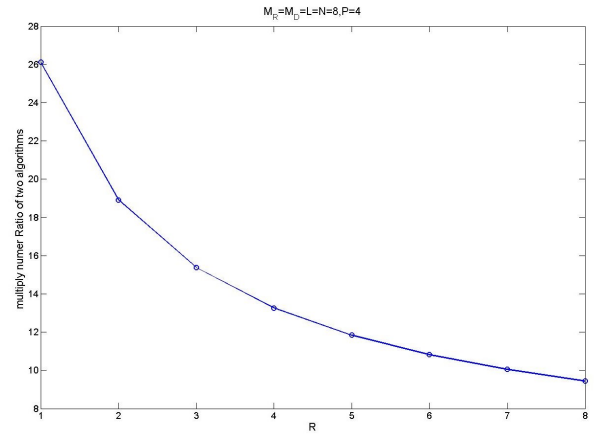


Fig. 5. Multiply operation number ratio of PT2-ALS and PT2-AT.

B. Iteration Number

Define NRE (Normalized Reconstruction Error) as

$$\varepsilon_k = \left\| \mathbf{x}_{M_D P N}^{(SRD)} - (\mathbf{S}_k^T \otimes \mathbf{G}^T)^T \odot (\mathbf{C} \otimes \hat{\mathbf{H}}_k^{(RD)}) \mathbf{h}_k^{(SR)} \right\|_2^2 \quad (32)$$

The estimated result is substituted into formula 32 to connect ε_k with iteration number, which is shown in Fig. 6. It can be shown that the iteration number of PT2-ALS is about ten times than PT2-AT when $E_S = 25\text{dB}$ and $E_S = 25\text{dB}$. We can also see that the convergence speed is much fast for the high energy symbol.

In the aspect of multiply operation number, the algorithm avoids the inverse operation to decrease the compute quantity by SVD tracking. In the aspect of iteration number, the updating at the moment $t+1$ uses the prior knowledge of the moment t , in some sense, compared with PT2-ALS, the algorithm uses more information, hence improve the convergence speed.

C. Estimation Performance

In the aspect of estimation performance, compare traditional supervised BALS and LS-SVD algorithm. The basic idea of the two algorithms is similar. The orthogonal train sequence is transmitted to the relay, which is known at the destination. Before the re-transmission of the relay, the

time independent gain matrix is used. At destination, the received signal is processed using the train sequence, which satisfy PARAFAC model, and then use the ALS and LS to estimate the channel matrix. Zero force equalizer is used to estimate the symbol matrix for the two algorithms.

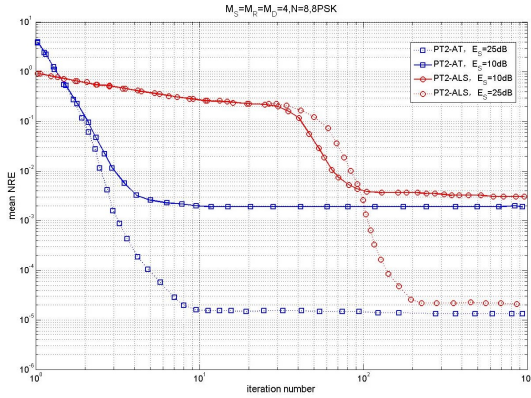


Fig. 6. The relation of iteration number and NRE.

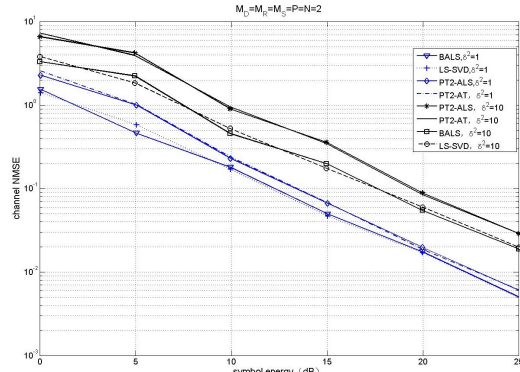


Fig. 7. Comparison of NMSE between PT2-AT and BALS, LS-SVD.

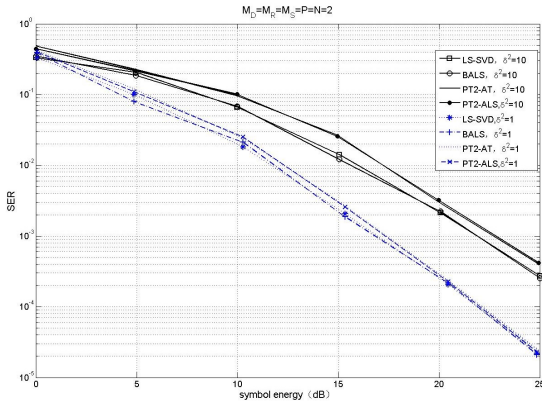


Fig. 8. comparison of SER between PT2-AT and BALS, LS-SVD.

We choose the two cases ($\sigma^2 = \sigma_{(R)}^2 = \sigma_{(D)}^2$) in which noise variance is 1 and 10 respectively. In order to have comparability, the three algorithms set the same transmission rate. The result of channel NMSE and SER are shown in Fig. 7 and Fig. 8. As can be seen from the Fig. 7 and Fig. 8 that: (1) In the aspect of channel NMSE, BALS and LS-SVD have the same performance with the same symbol energy, which results from the application of train sequence. NMSE performance of PT2-AT algorithm is worse than BALS and LS-SVD lightly when noise variance is 10; and is almost the same as the BALS and LS-SVD when noise variance is 1. (2) In the aspect of SER, BALS and LS-SVD has the similar performance with the same symbol energy. The SER

performance of PT2-AT is the same as BALS and LS-SVD with the different symbol energy.

Owing to assumption of varies slowly channel, the Khatri-Rao product structure of $\mathbf{H}(t+1)$ is neglected in the analysis process. So the model is different from real application when noise variance is very large, or the SNR is very low, which is the reason for the performance decreases with low SNR. However, the performance tends to the traditional channel estimation with SNR increases.

VI. CONCLUSION

A novel semi-blind receiver algorithm was proposed based on the model of one way and two hops MIMO relay system. Firstly, the algorithm adds a time dimensionality on the premise of no supervise sequence, then chooses trunk window reasonable, creates iteration relationship between auxiliary matrix and base matrix of right subspace, correct the channel parameter adaptively, and make it closer to true parameters in least error square sense. Computational cost of the algorithm is reduced by avoiding Moore Penrose pseudo-inverse operation efficiently through tracking and iterating the singular of matrix. The algorithm also improves convergence speed by utilizing the estimated result of previous moment. Simulation result shows that the algorithm has lower computational cost, higher convergence speed, and the same estimation performance with ALS premised on that no supervised sequence added. The algorithm is practical meaning for the engineering applications of semi-blind receiver based on the PARATUCK2 model.

APPENDIX

Pseudo-inversion lemma of Rank-one update:

Let $A \in \mathbb{C}^{M \times N}$ be a full column rank matrix. Considering the vectors $c \in \mathbb{C}^{M \times 1}$, $d \in \mathbb{C}^{N \times 1}$. Assumed that $A + cd^H$ is full column rank, then

$$(A + cd^H)^+ = A^+ + \frac{1}{\beta^*} A^+ h^H u^H - \frac{\beta^*}{\sigma} p q^H$$

$$\beta = 1 + d^H A^+ c, h = d^H A^+, u = (I_M - A A^+) c, k = A^+ c,$$

$$p = -(\|u\|^2 / \beta^* A^+ h^H + k), q^H = -(\|h\|^2 / \beta^* u^{H+} + k),$$

$$\sigma = \|h\|^2 \|u\|^2 + |\beta|^2.$$

Note that in the case $M = N$, $u=0$, $(A + cd^H)^+ = A^+ - 1 / \beta k h$.

REFERENCES

- [1] L. Cao, J. Zhang, and N. Kanno, "Multi-user cooperative communications with relay coding for uplink IMT-advanced 4G systems," in *Proc. in IEEE GLOBECOM*, Honolulu, HI, Dec. 2009, pp. 1-6.
- [2] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 4, pp. 570-578, July 1993.

- [3] M. Dohler and Y. Li, *Cooperative Communications: Hardware, Channel, and PHY*, John Wiley & Sons, 2010.
- [4] G. Favier, M. N. Costa, A. L. F. Almeida, and J. M. T. Romano, "Tensor space-time coding for MIMO wireless communication systems," *ELSEVIER Signal Processing*, no. 4, pp. 1079-1092, 2012.
- [5] N. D. Sidiropoulos and R. S. Budanmpati, "Khatri-Rao space time codes," *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2396-2407, Oct. 2002.
- [6] L. R. Ximenes, G. Favier, A. L. F. Almeida, and Y. C. B. Silva, "PARAFAC-PARATUCK semi-blind receivers for two-hop cooperative MIMO relay systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3604-3615, July 2014.
- [7] L. R. Ximenes, G. Favier, and A. L. F. Almeida, "Closed-form semi-blind receiver for two-hop MIMO relay systems using a double Khatri-Rao space-time coding," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 316-320, March 2016.
- [8] Y. Rong, M. Khandaker, and Y. Xiang, "Channel estimation of dual-hop MIMO relay system via parallel factor analysis," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2224-2233, Jun. 2012.
- [9] P. Lioliou, M. Viberg, and M. Coldrey, "Least-squares based channel estimation for MIMO relays," *International ITG Workshop on Smart Antennas*, pp. 90-95.
- [10] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," *IEEE transactions Signal Processing*, vol. 45, pp. 1222-1240, May 1997.
- [11] R. Badeau, G. Richard, and B. David, "Sliding window adaptive SVD algorithms," *IEEE transactions Signal Processing*, vol. 52, pp. 1-10, Jan. 2004.
- [12] R. Costantini, L. Sbaiz, and S. Susstrunk, "Higher order SVD analysis for dynamic texture synthesis," *IEEE transactions Signal Processing*, vol. 17, no. 1, pp. 42-52, Jan. 2008.
- [13] L. Lathauwer, "A Link between the canonical decomposition multi-linear algebra and simultaneous matrix diagonalization," *SIAM J. Matrix Annual applications*, vol. 28, no. 3, pp. 942-666, 2006.
- [14] M. Rajih, P. Comon, and R. A. Harshman, "Enhanced line search: A novel method to accelerate PARAFAC," *SIAM J. Matrix Annual applications, Tensor Decomposition Applications*, vol. 30, no. 3, pp. 1148-1171, Sep. 2008.

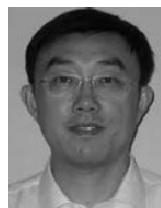


Wang Rui received the PhD degree in science and information engineering from the Naval Aeronautical Engineering Institute, China, in 2016.

He is currently an assistant professor with the Department of Electronic Information Engineering. From 2009-2010, he was a visiting researcher at the Institute of Acoustics of the Chinese Academy.

Dr. Wang Rui's research interests lie in the areas of

signal processing for communication and sensor array processing, and include blind methods for channel estimation, and tensor-based signal processing to communication and data analysis.

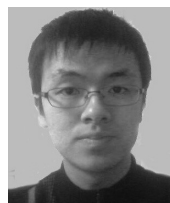


Rui Guo-Sheng received the PhD degree in Harbin Institute of Technology in 2000.

He is currently a professor with the Department of Electronic Information Engineering.

Professor RUI Guo-Sheng's research interests lie in the areas of signal processing for communication and sensor array processing, and include advanced technology of wireless communication and tensor

based signal processing.



Zhang Yang received the PhD degree in science and information engineering from the Naval Aeronautical Engineering Institute, China, in 2013.

He is currently an assistant professor with the Department of Electronic Information Engineering. From 2009-2010, he was a visiting researcher at the Institute of Electronics of the Chinese Academy.

Dr. Zhang Yang's research interests lie in the areas of signal processing for communication and sensor array processing, and include blind methods for channel estimation, and tensor-based signal processing to communication and data analysis.



Xue Peng received the PhD degree in science and information engineering from Tsinghua University, Beijing, China, in 2014.

He is currently an assistant professor with the Department of Electronic Information Engineering in the Naval Aeronautical Engineering Institute. From 2011-2012, he was a visiting researcher at the Institute of Electronics of the Chinese Academy.

Dr. Xue Peng's research interests lie in the areas of signal processing for communication and sensor array processing, and include blind methods for channel estimation, and tensor-based signal processing to communication and data analysis.