一种多用户上行放大转发中继系统中快速收敛的信道估计方法

林和昀　袁超伟　杜建和

A fast algorithm with convergence for channel estimation in multi-user uplink amplify-and-forward relay system

Lin He-Yun　Yuan Chao-Wei　Du Jian-He

引用信息 Citation: Acta Physica Sinica, 65, 210201 (2016)　DOI: 10.7498/aps.65.210201
在线阅读 View online: http://dx.doi.org/10.7498/aps.65.210201
当期内容 View table of contents: http://wulixb.iphy.ac.cn/CN/Y2016/V65/I21

物理学报 Acta Phys. Sin. Vol. 65, No. 21 (2016) 210201

一种多用户上行放大转发中继系统中快速收敛的信道估计方法*

林和昀[1)] 袁超伟[1)†] 杜建和[2)]

1) (北京邮电大学信息与通信工程学院, 北京 100876)

2) (中国传媒大学信息工程学院, 北京 100024)

(2016 年 6 月 21 日收到; 2016 年 7 月 27 日收到修改稿)

针对传统交替最小二乘算法存在的收敛缓慢问题, 本文在多用户上行放大转发中继系统中基于 Levenberg Marquardt (LM) 算法, 提出了一种能够快速收敛的信道估计方法, 实现了用户 - 中继信道和中继 - 基站信道的独立估计. 在基站, 通过对中继多次放大转发的信号进行建模, 构造出具有平行因子结构的三维信号张量模型, 并采用 LM 算法对该模型进行拟合, 从而得到系统中两跳链路的信道状态信息. 理论分析与仿真结果表明, 与已有二线性交替最小二乘方法相比, 所提方法具有近乎相同的估计精度; 当中继放大因子矩阵为随机矩阵或者包含近似共线性相关列时, 所提方法具有更快的收敛速度.

关键词: 张量模型, 信道估计, Levenberg Marquardt 算法, 放大转发中继

PACS: 02.10.Xm, 84.40.Ua, 43.60.Gk **DOI:** 10.7498/aps.65.210201

1 引 言

在无需改变无线通信系统终端硬件设施的前提下, 中继节点能够消除弱覆盖区域, 扩展传输范围, 提升系统性能, 从而得到了人们的广泛关注 [1]. 在各种中继转发协议中, 放大转发 (amplify-and-forward, AF) 协议由于实现简单、无需涉及复杂的信号处理过程, 从而得到了大量的应用 [2−4]. 文献 [2—4] 分别从最优中继放大因子矩阵、信源-中继联合预编码矩阵以及信道容量等方面对 AF 中继系统进行了研究. 虽然文献 [2—4] 取得了较好的效果, 但是上述方法都是在假设信宿已知所有信道精确的信道状态信息 (channel state information, CSI) 情况下进行的研究. 然而, 在实际的无线通信系统中, 真实的信道 [5] 情况是未知的. 系统只有通过信道估计技术 [6] 获得精确的 CSI 后, 才能进一步进行优化方案的设计, 提升系统性能.

近几年来, 人们将张量 [7,8] 信号处理技术与多输入多输出 (multiple input multiple output, MIMO) [9] 中继技术相结合, 利用张量信号处理技术多个独立的维度资源来提升中继系统的性能, 并取得了一系列的研究成果 [10−13]. 文献 [10] 在上行多用户直接序列码分多址协作通信系统中提出了基于平行因子 (parallel factor, PARAFAC) 模型的接收机设计方案, 实现了信道和信号的联合估计. 然而, 该方案仅适用于分簇式中继场景, 要求不同簇之间的信道是正交的, 并不能直接适用于普通的 AF 中继场景中. 文献 [11] 将双 Khatri-Rao 空时码引入包含信源-信宿直接信道链路的协作通信系统中, 并设计基于四维互嵌式平行因子 (Nested PARAFAC) 模型的张量接收机, 实现了信道和信号的半盲检测. 虽然文献 [10, 11] 在半双工 AF 协作通信系统中取得了很好的效果, 但是所提方法只能对系统中信源-中继-信宿 (S-R-D) 的组合信道进行

* 国家高技术研究发展计划 (批准号: 2015AA01A705, 2014AA01A701) 和中国传媒大学理工科规划项目 (批准号: 3132016XNG1618) 资助的课题.

† 通信作者. E-mail: yuancw2000@bupt.edu.cn

 http://wulixb.iphy.ac.cn

估计, 而无法实现两跳信道的独立估计. 然而, 在实际的AF中继系统中, 为了优化整个系统, 需要获得信源-中继(S-R)和中继-信宿(R-D)两跳信道独立的CSI. 为此, 文献[12]提出了一种基于导频信号的信道估计方法, 该方法在信源通过传输K个相同的导频信号块, 在信宿构造出具有PARAFAC结构的接收信号, 并采用二线性交替最小二乘(bilinear alternating least-squares, BALS)的方法实现了两跳信道的独立估计. 为了避免使用导频信号, 文献[13]提出了一种基于多组Khatri-Rao空时码的半盲估计方案, 同时获得了两跳信道和信号的独立估计. 虽然文献[12, 13]都能够很好地获得AF中继系统中两跳信道的CSI, 但是所提方法都需要经过较长的训练时间才能构造出包含CSI的张量模型, 从而导致了较低的系统频谱效率. 此外, 当文献[12]中的放大因子矩阵是随机的或者矩阵的列之间近似共线性时, 或者导频信号块的个数小于中继的天线数目时, BALS算法的收敛速度将会变得缓慢. 因此, 针对AF中继系统, 如何在更短的训练时间内获得两跳信道的CSI, 有待于进一步研究.

本文针对多用户上行AF中继系统, 提出了一种新的基于导频信号的信道估计方法. 该方法只需要在用户端发送一次训练信号, 通过中继节点的多次放大转发, 在基站构造出三维PARAFAC信号张量模型, 并采用Levenberg-Marquardt (LM)算法对该张量模型进行拟合, 实现了两跳信道的联合估计. 此外, 本文还对信道矩阵的可辨识性进行了讨论. 本文的主要创新点如下: 1)与文献[12]所提传输方法相比, 所提传输方法在用户端只需要发送一次训练信号, 具有更高的频谱效率; 2)与文献[12]提出的BALS拟合算法相比, 所提LM算法具有相似的估计精度, 并且在放大因子矩阵为随机矩阵或者矩阵的列之间近似共线性时, 所提LM算法具有更快的收敛速度; 3)所提算法具有稳健的估计性能, 在不同的场景下均具有较好的估计结果.

2 PARAFAC模型及其分解唯一性

考虑一个具有平行因子结构并且秩为R_1的三阶张量$\boldsymbol{\mathcal{W}} \in \mathbb{C}^{I_1\times I_2\times I_3}$, 它的加载矩阵分别是$\boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1\times R_1}$, $\boldsymbol{A}^{(2)} \in \mathbb{C}^{I_2\times R_1}$和$\boldsymbol{A}^{(3)} \in \mathbb{C}^{I_3\times R_1}$, 该矩阵的PARAFAC分解表达式为[7,8]

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{I}}_{R_1} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \boldsymbol{A}^{(3)}, \tag{1}$$

其中, $\boldsymbol{\mathcal{I}}_{R_1} \in \mathbb{C}^{R_1\times R_1\times R_1}$是一个三阶的单位矩阵. (1)式还可以写为如下的分解表达式:

$$\begin{aligned}&\boldsymbol{\mathcal{W}}(i_1,i_2,i_3)\\&= \sum_{r_1=1}^{R_1} \boldsymbol{A}^{(1)}(i_1,r_1)\boldsymbol{A}^{(2)}(i_2,r_1)\boldsymbol{A}^{(3)}(i_3,r_1),\end{aligned} \tag{2}$$

其中, $\boldsymbol{A}^{(1)}(i_1,r_1)$, $\boldsymbol{A}^{(2)}(i_2,r_1)$和$\boldsymbol{A}^{(3)}(i_3,r_1)$分别是$\boldsymbol{A}^{(1)}$, $\boldsymbol{A}^{(2)}$和$\boldsymbol{A}^{(3)}$中对应的元素. (2)式为一个三线性分解表达式, 表示$\boldsymbol{\mathcal{W}}$中的每个元素都可视为R_1个秩1的三重因子乘积之和. 沿着该模型的I_1, I_2和I_3三个维度进行展开, $\boldsymbol{\mathcal{W}}$可以表示为以下三种矩阵展开式:

$$\begin{aligned}\boldsymbol{W}^{(1)} &= (\boldsymbol{A}^{(2)} \odot \boldsymbol{A}^{(3)})\boldsymbol{A}^{(1)\mathrm{T}} \in \mathbb{C}^{I_3I_2\times I_1},\\ \boldsymbol{W}^{(2)} &= (\boldsymbol{A}^{(3)} \odot \boldsymbol{A}^{(1)})\boldsymbol{A}^{(2)\mathrm{T}} \in \mathbb{C}^{I_1I_3\times I_2},\\ \boldsymbol{W}^{(3)} &= (\boldsymbol{A}^{(1)} \odot \boldsymbol{A}^{(2)})\boldsymbol{A}^{(3)\mathrm{T}} \in \mathbb{C}^{I_2I_1\times I_3},\end{aligned} \tag{3}$$

其中$\odot$表示矩阵的Khatri-Rao乘积. 此时, 矩阵$\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$和$\boldsymbol{W}^{(3)}$均包含张量$\boldsymbol{\mathcal{W}}$中所有的元素, 区别在于各矩阵中元素的排列顺序不同. 根据PARAFAC模型的分解唯一性可知[7], 当满足不等式

$$k_{\boldsymbol{A}^{(1)}} + k_{\boldsymbol{A}^{(2)}} + k_{\boldsymbol{A}^{(3)}} - 2 \geqslant 2R_1 \tag{4}$$

时, 该张量模型的三个加载矩阵$\boldsymbol{A}^{(1)}$, $\boldsymbol{A}^{(2)}$和$\boldsymbol{A}^{(3)}$本质唯一(即只存在尺度模糊和列模糊). 其中, $k_{\boldsymbol{A}^{(1)}}$, $k_{\boldsymbol{A}^{(2)}}$和$k_{\boldsymbol{A}^{(3)}}$分别是三个加载矩阵的Kruskal秩[8].

3 系统模型

本文考虑的多用户上行AF中继无线通信系统如图1所示, M个用户在R个中继节点的辅助下与配置了N根天线的基站进行通信, 其中所有用户和中继节点均配置单根天线. 考虑传输过程中存在路径损耗和阴影效应, 系统不考虑用户到基站的直接链路. 因此, 系统的传输路径为用户-中继信道$\boldsymbol{H}^{(1)} \in \mathbb{C}^{R\times M}$和中继-基站信道$\boldsymbol{H}^{(2)} \in \mathbb{C}^{N\times R}$. $\boldsymbol{H}^{(1)}$和$\boldsymbol{H}^{(2)}$中的元素均服从瑞利分布, 并且元素之间相互独立. 假设$\boldsymbol{H}^{(1)}$和$\boldsymbol{H}^{(2)}$为准静态信道模型, 即CSI在某个阶段内是保持不变的. 与文献[10—13]相同, 本文进一步假设中继工作在半双工模式下, 因此整个系统的传输过程可分为两个阶段. 在第一阶段, 所有用户发送长度为$L(L \geqslant M)$

的正交训练信号块 $\boldsymbol{S} \in \mathbb{C}^{M\times L}$ ($\boldsymbol{S}\boldsymbol{S}^{\mathrm{H}} = \boldsymbol{I}_{\mathrm{M}}$) 到 R 个中继节点. 此时, 中继端的接收信号表达式为

$$\boldsymbol{Y}^{(1)} = \boldsymbol{H}^{(1)}\boldsymbol{S} + \boldsymbol{V}^{(1)} \in \mathbb{C}^{R\times L}, \tag{5}$$

其中, $\boldsymbol{Y}^{(1)} = [\boldsymbol{y}_1^{(1)}, \cdots, \boldsymbol{y}_r^{(1)}, \boldsymbol{y}_R^{(1)}]^{\mathrm{T}}$, $\boldsymbol{y}_r^{(1)} \in \mathbb{C}^{1\times L}$ 为第 r 个中继节点的接收信号向量, $r = 1, \cdots, R$, $\boldsymbol{V}^{(1)} \in \mathbb{C}^{R\times L}$ 是中继节点处的加性高斯噪声矩阵. 在第二阶段, 中继节点采用对角放大因子矩阵组 $\{\boldsymbol{G}_k | k = 1, \cdots, K\}$ 对接收到的 $\boldsymbol{Y}^{(1)}$ 进行放大并转发到基站. 该阶段的总传输时间为 K 个时间块. 基站在第 K 个时间块内的接收信号可表示为

$$\begin{aligned}\boldsymbol{Y}_k^{(2)} &= \boldsymbol{H}^{(2)}\boldsymbol{G}_k\boldsymbol{H}^{(1)}\boldsymbol{S} + \boldsymbol{H}^{(2)}\boldsymbol{G}_k\boldsymbol{V}^{(1)} + \boldsymbol{V}_k^{(2)} \\ &= \boldsymbol{H}^{(2)}\boldsymbol{G}_k\boldsymbol{H}^{(2)}\boldsymbol{S} + \bar{\boldsymbol{V}}_k^{(2)} \in \mathbb{C}^{N\times L},\end{aligned} \tag{6}$$

其中, $\boldsymbol{V}_k^{(2)} \in \mathbb{C}^{N\times L}$ 表示基站的加性高斯噪声矩阵, $k = 1, \cdots, K$, $\bar{\boldsymbol{V}}_k^{(2)}$ 为基站第 K 个时间块内的接收信号有效噪声矩阵. 假设 $\boldsymbol{V}^{(1)}$ 和 $\boldsymbol{V}_k^{(2)}$ 中的元素都是均值为0方差为1的独立同分布的复高斯随机变量.

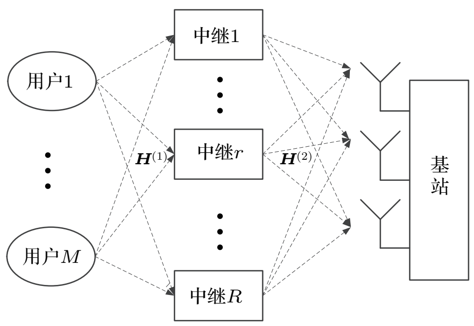

图1 上行多用户 AF 中继系统

Fig. 1. System model of the multi-user uplinks AF relay systems.

4 基于PARAFAC模型的信道估计方法

4.1 PARAFAC建模及信道可辨识性分析

为了不失一般性, 省略噪声项, 基站接收完 K 个时间块的信号后, 对 $\boldsymbol{Y}_k^{(2)}$, $k = 1, \cdots, K$, 进行堆栈处理, 并右乘 $\boldsymbol{S}^{\mathrm{H}}$, 有

$$\begin{aligned}\boldsymbol{X} &= \begin{bmatrix}\boldsymbol{Y}_1^{(2)} \\ \vdots \\ \boldsymbol{Y}_K^{(2)}\end{bmatrix}\boldsymbol{S}^{\mathrm{H}} = \begin{bmatrix}\boldsymbol{H}^{(2)}\boldsymbol{D}_1(\boldsymbol{F}) \\ \vdots \\ \boldsymbol{H}^{(2)}\boldsymbol{D}_K(\boldsymbol{F})\end{bmatrix}\boldsymbol{H}^{(1)} \\ &= (\boldsymbol{F} \odot \boldsymbol{H}^{(2)})(\boldsymbol{H}^{(1)\mathrm{T}})^{\mathrm{T}} \in \mathbb{C}^{KN\times M},\end{aligned} \tag{7}$$

其中, $\boldsymbol{X}$ 是基站在 K 个时间块后接收的总信号; $\boldsymbol{F} \in \mathbb{C}^{K\times R}$ 是放大因子组构成的矩阵; $\boldsymbol{D}_k(\boldsymbol{F})$ 表示矩阵 $\boldsymbol{F}$ 的第 k 行元素所构成的对角阵, 有 $\boldsymbol{D}_k(\boldsymbol{F}) = \boldsymbol{G}_k$. 根据文献 [7], (7) 式是三维PARAFAC信号模型 $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{N\times M\times K}$ 的一个矩阵展开式, $\boldsymbol{H}^{(1)}$, $\boldsymbol{H}^{(2)}$ 和 $\boldsymbol{F}$ 可视为张量模型 $\boldsymbol{\mathcal{X}}$ 的三个加载矩阵. N, M 和 K 是 $\boldsymbol{\mathcal{X}}$ 的三个维度, 分别对应的是空间维度 (接收天线、发送天线) 和时间维度 (时间分集). 因此, $\boldsymbol{\mathcal{X}}$ 是一个包含了空-时接收信号的三维张量模型. 该三维张量模型对应的分解式为

$$\begin{aligned}&\boldsymbol{\mathcal{X}}(n, m, k) \\ &= \sum_{r=1}^{R} \boldsymbol{H}^{(2)}(n, r)\boldsymbol{F}(k, r)\boldsymbol{H}^{(1)}(m, r).\end{aligned} \tag{8}$$

由前文可知, (7) 式还可以表示为如下的矩阵展开式:

$$\boldsymbol{Y} = (\boldsymbol{H}^{(1)\mathrm{T}} \odot \boldsymbol{F})\boldsymbol{H}^{(2)\mathrm{T}} \in \mathbb{C}^{MK\times N}. \tag{9}$$

由第2节内容可知, $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 分别是信号张量模型 $\boldsymbol{\mathcal{X}}$ 沿时间方向 (K) 和空间方向 (M) 得到的矩阵展开式. 本文考虑 $N = M = R$ 的系统配置, 且 $\boldsymbol{H}^{(1)}$ 和 $\boldsymbol{H}^{(2)}$ 是随机矩阵, 可知 $\boldsymbol{H}^{(1)}$ 和 $\boldsymbol{H}^{(2)}$ 具有满 Kruskal 秩 [8]. 同时假设 $\boldsymbol{F}$ 有满 Kruskal 秩, 根据式 PARAFAC 分解唯一性定理 [7,8], 当满足

$$\min(R, K) \geqslant 2 \tag{10}$$

时, 三维张量 $\boldsymbol{\mathcal{X}}$ 的分解具有唯一性, 即加载矩阵 $\boldsymbol{H}^{(1)}$, $\boldsymbol{H}^{(2)}$ 和 $\boldsymbol{F}$ 是可辨识的.

4.2 Levenberg-Marquardt 拟合算法

根据 (7) 和 (9) 式, 可采用传统的交替最小二乘 (alternating least squares, ALS) 算法实现对信道 $\boldsymbol{H}^{(1)}$ 和 $\boldsymbol{H}^{(2)}$ 的联合估计. 然而, ALS 算法对加载矩阵的要求较为敏感. 当加载矩阵的列之间存在高度的共线性时, 会使得构成的张量模型带有 "病态", 进而大大降低了 ALS 算法的收敛速度 [14]. 由于 LM [15] 算法可以实现二次收敛并能很好地解决 "病态" 张量模型的收敛问题, 该方法已经成功应用到多种不同的张量模型拟合过程中 [14,16,17]. 因此, 本节将采用 LM 算法对构造的 PARAFAC 信号模型进行拟合, 实现两跳信道的联合估计. 下面给出具体的实现过程.

首先, 令张量 $\hat{\boldsymbol{\mathcal{X}}} \in \mathbb{C}^{N\times M\times K}$ 为接收信号张量 $\boldsymbol{\mathcal{X}}$ 的估计结果. 令 φ 为两者误差, 作为估计的代价

函数, 即

$$\varphi=\frac{1}{2}\|\hat{\boldsymbol{\mathcal{X}}}-\boldsymbol{\mathcal{X}}\|_F^2. \tag{11}$$

对 $\boldsymbol{H}^{(1)}$ 和 $\boldsymbol{H}^{(2)\mathrm{T}}$ 进行向量化操作后, 按顺序存储在全局向量 $\boldsymbol{p}\in\mathbb{C}^{Q\times1}$ 中, 有

$$\boldsymbol{p}=[\boldsymbol{p}_{\boldsymbol{H}^{(1)}}^{\mathrm{T}},\boldsymbol{p}_{\boldsymbol{H}^{(2)}}^{\mathrm{T}}]^{\mathrm{T}}, \tag{12}$$

其中, $Q=(N+M)R$. 由 (12) 式可知, $\boldsymbol{p}$ 可拆分为两个独立的子向量, 每个子向量包含了不同链路的信道状态信息. 分别定义 $\boldsymbol{x}(\boldsymbol{p})\in\mathbb{C}^{F\times1}$, $\tilde{\boldsymbol{x}}\in\mathbb{C}^{F\times1}$ 和 $\boldsymbol{r}(\boldsymbol{p})\in\mathbb{C}^{F\times1}$ 为接收信号张量 $\boldsymbol{\mathcal{X}}$ 的估计向量表达式、加噪向量表达式和估计误差向量, 则

$$\boldsymbol{r}(\boldsymbol{p})=\boldsymbol{x}(\boldsymbol{p})-\tilde{\boldsymbol{x}}, \tag{13}$$

其中, $F=NMK$. 此时, 代价函数 (11) 式转变为

$$\varphi(\boldsymbol{p})=\frac{1}{2}|\boldsymbol{x}(\boldsymbol{p})-\tilde{\boldsymbol{x}}|^2=\frac{1}{2}\boldsymbol{r}^{\mathrm{H}}(\boldsymbol{p})\boldsymbol{r}(\boldsymbol{p}). \tag{14}$$

对 $\varphi(\boldsymbol{p})$ 求关于 $\boldsymbol{p}$ 的一阶偏导数, 有

$$\boldsymbol{g}(\boldsymbol{p})=\varphi'(\boldsymbol{p})=\boldsymbol{J}^{\mathrm{H}}(\boldsymbol{p})\boldsymbol{r}(\boldsymbol{p}), \tag{15}$$

其中, $\boldsymbol{g}(\boldsymbol{p})$ 为 $\varphi(\boldsymbol{p})$ 关于 $\boldsymbol{p}$ 的复梯度向量 [14,16], Jacobian 矩阵 $\boldsymbol{J}(\boldsymbol{p})\in\mathbb{C}^{L\times Q}$ 是 $\boldsymbol{r}(\boldsymbol{p})$ 的一阶偏导数, 并有如下表达式,

$$\boldsymbol{J}(\boldsymbol{p})=\frac{\partial\boldsymbol{r}(\boldsymbol{p})}{\partial\boldsymbol{p}}=\frac{\partial\boldsymbol{x}(\boldsymbol{p})}{\partial\boldsymbol{p}}. \tag{16}$$

为了使得 $\varphi(\boldsymbol{p})$ 最小, LM 算法 [15] 的基本思想是在每一次迭代的过程中更新一次步长 $\Delta\boldsymbol{p}$, 使得以下等式成立:

$$(\boldsymbol{J}^{(i)\mathrm{H}}(\boldsymbol{p})\boldsymbol{J}^{(i)}(\boldsymbol{p})+\lambda^{(i)}\boldsymbol{I})\Delta\boldsymbol{p}^{(i)}=-\boldsymbol{g}^{(i)}(\boldsymbol{p}), \tag{17}$$

$$\boldsymbol{p}^{(i+1)}=\boldsymbol{p}^{(i)}+\Delta\boldsymbol{p}^{(i)}, \tag{18}$$

其中, λ 是一个恒大于 0 的阻尼系数, 以保证在每一次更新中标准方程式 $(\boldsymbol{J}^{\mathrm{H}}(\boldsymbol{p})\boldsymbol{J}(\boldsymbol{p})+\lambda\boldsymbol{I})$ 是非奇异和正定的, 从而保证了 $\Delta\boldsymbol{p}$ 陡降方向的正确性.

从 (17) 和 (18) 式可知, 采用 LM 算法对未知矩阵进行更新, 就需要首先得到 Jacobian 矩阵和梯度向量. 根据 PARAFAC 模型的特殊结构, 可得如下求解过程.

对 (7) 和 (9) 式进行向量化操作, 有

$$\begin{cases}\boldsymbol{x}(\boldsymbol{p})=\mathrm{vec}(\boldsymbol{X})\\ =(\boldsymbol{I}_{\mathrm{M}}\otimes(\boldsymbol{F}\odot\boldsymbol{H}^{(2)}))\boldsymbol{p}_{\boldsymbol{H}^{(1)}}\in\mathbb{C}^{MKN\times1},\\ \boldsymbol{y}(\boldsymbol{p})=\mathrm{vec}(\boldsymbol{Y})\\ =(\boldsymbol{I}_{\mathrm{N}}\otimes(\boldsymbol{H}^{(1)\mathrm{T}}\odot\boldsymbol{F}))\boldsymbol{p}_{\boldsymbol{H}^{(2)}}\in\mathbb{C}^{NMK\times1},\end{cases} \tag{19}$$

其中, $\boldsymbol{p}_{\boldsymbol{H}^{(1)}}=\mathrm{vec}(\boldsymbol{H}^{(1)})\in\mathbb{C}^{MR\times1}$, $\boldsymbol{p}_{\boldsymbol{H}^{(2)}}=\mathrm{vec}(\boldsymbol{H}^{(2)\mathrm{T}})\in\mathbb{C}^{RN\times1}$, $\mathrm{vec}(\cdot)$ 表示向量化操作. 两者关系如下:

$$\boldsymbol{x}(\boldsymbol{p})=\boldsymbol{\Pi}\boldsymbol{y}(\boldsymbol{p}). \tag{20}$$

其中, $\boldsymbol{\Pi}$ 为置换矩阵, 并有如下表达式,

$$\boldsymbol{\Pi}=\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N}\boldsymbol{e}_m^{(M)}\boldsymbol{e}_n^{(N)\mathrm{T}}\otimes\boldsymbol{e}_k^{(K)}\boldsymbol{e}_m^{(M)\mathrm{T}}\\ \otimes\boldsymbol{e}_n^{(N)}\boldsymbol{e}_k^{(K)\mathrm{T}}, \tag{21}$$

$\boldsymbol{e}_{l_1}^{(L_1)}, l_1=1,\cdots,L_1$ 为欧氏空间 $\mathbb{R}^{L_1}$ 的第 l_1 列, $L_1\in\{M,N,K\}$. 此时, 根据 (16) 式有

$$\begin{cases}\boldsymbol{J}(\boldsymbol{p}_{\boldsymbol{H}^{(1)}})=(\boldsymbol{I}_{\mathrm{M}}\otimes(\boldsymbol{F}\odot\boldsymbol{H}^{(2)})),\\ \boldsymbol{J}(\boldsymbol{p}_{\boldsymbol{H}^{(2)}})=\boldsymbol{\Pi}(\boldsymbol{I}_{\mathrm{N}}\otimes(\boldsymbol{H}^{(1)\mathrm{T}}\odot\boldsymbol{F})).\end{cases} \tag{22}$$

根据 $\boldsymbol{p}$ 的可拆分结构可知, $\boldsymbol{J}=[\boldsymbol{J}_{\boldsymbol{H}^{(1)}},\boldsymbol{J}_{\boldsymbol{H}^{(2)}}]$, 其中 $\boldsymbol{J}$, $\boldsymbol{J}_{\boldsymbol{H}^{(1)}}$ 和 $\boldsymbol{J}_{\boldsymbol{H}^{(2)}}$ 分别是表示 $\boldsymbol{J}(\boldsymbol{p})$, $\boldsymbol{J}(\boldsymbol{p}_{\boldsymbol{H}^{(1)}})$ 和 $\boldsymbol{J}(\boldsymbol{p}_{\boldsymbol{H}^{(2)}})$. 此时, 矩阵 $\boldsymbol{J}^{\mathrm{H}}\boldsymbol{J}$ 有如下结构:

$$\boldsymbol{J}^{\mathrm{H}}\boldsymbol{J}=\begin{bmatrix}\boldsymbol{J}_{\boldsymbol{H}^{(1)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(1)}} & \boldsymbol{J}_{\boldsymbol{H}^{(1)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(2)}}\\ \boldsymbol{J}_{\boldsymbol{H}^{(2)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(1)}} & \boldsymbol{J}_{\boldsymbol{H}^{(2)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(2)}}\end{bmatrix}. \tag{23}$$

该矩阵可以利用已有的子矩阵 $\boldsymbol{J}_{\boldsymbol{H}^{(1)}}$ 和 $\boldsymbol{J}_{\boldsymbol{H}^{(2)}}$ 进行分块计算得到:

$$\begin{cases}\boldsymbol{J}_{\boldsymbol{H}^{(1)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(1)}}\\ =\boldsymbol{I}_{\mathrm{M}}\otimes((\boldsymbol{F}\odot\boldsymbol{H}^{(2)})^{\mathrm{H}}(\boldsymbol{F}\odot\boldsymbol{H}^{(2)})),\\ \boldsymbol{J}_{\boldsymbol{H}^{(2)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(2)}}\\ =\boldsymbol{I}_{\mathrm{N}}\otimes((\boldsymbol{H}^{(1)\mathrm{T}}\odot\boldsymbol{F})^{\mathrm{H}}(\boldsymbol{H}^{(1)\mathrm{T}}\odot\boldsymbol{F})),\\ \boldsymbol{J}_{\boldsymbol{H}^{(1)}}^{\mathrm{H}}\boldsymbol{J}_{\boldsymbol{H}^{(2)}}\\ =(\boldsymbol{I}_{\mathrm{M}}\otimes(\boldsymbol{F}\odot\boldsymbol{H}^{(2)}))^{\mathrm{H}}\boldsymbol{\Pi}(\boldsymbol{I}_{\mathrm{N}}\otimes(\boldsymbol{H}^{(1)\mathrm{T}}\odot\boldsymbol{F})).\end{cases} \tag{24}$$

同样, (15) 式可以等效为如下形式:

$$\boldsymbol{g}=\begin{bmatrix}\varphi'(\boldsymbol{p}_{\boldsymbol{H}^{(1)}})\\ \varphi'(\boldsymbol{p}_{\boldsymbol{H}^{(2)}})\end{bmatrix}=\begin{bmatrix}\boldsymbol{g}_{\boldsymbol{H}^{(1)}}\\ \boldsymbol{g}_{\boldsymbol{H}^{(2)}}\end{bmatrix}. \tag{25}$$

通过 (19) 式, 代价函数 (14) 式可以进一步等效为如下两种形式:

$$\begin{cases}\varphi=\dfrac{1}{2}\|\tilde{\boldsymbol{x}}-(\boldsymbol{I}_{\mathrm{M}}\otimes(\boldsymbol{F}\odot\boldsymbol{H}^{(2)}))\boldsymbol{p}_{\boldsymbol{H}^{(1)}}\|^2,\\ \varphi=\dfrac{1}{2}\|\tilde{\boldsymbol{y}}-(\boldsymbol{I}_{\mathrm{N}}\otimes(\boldsymbol{H}^{(1)\mathrm{T}}\odot\boldsymbol{F}))\boldsymbol{p}_{\boldsymbol{H}^{(2)}}\|^2.\end{cases} \tag{26}$$

将(26)式代入(25)式, 则

$$\begin{cases} \boldsymbol{g}_{\boldsymbol{H}^{(1)}} &= \boldsymbol{J}_{\boldsymbol{H}^{(1)}}^{\mathrm{H}} \boldsymbol{J}_{\boldsymbol{H}^{(1)}} \boldsymbol{p}_{\boldsymbol{H}^{(1)}} \\ &\quad -(\boldsymbol{I}_{\mathrm{M}} \otimes (\boldsymbol{F} \odot \boldsymbol{H}^{(2)}))^{\mathrm{H}} \tilde{\boldsymbol{x}}, \\ \boldsymbol{g}_{\boldsymbol{H}^{(2)}} &= \boldsymbol{J}_{\boldsymbol{H}^{(2)}}^{\mathrm{H}} \boldsymbol{J}_{\boldsymbol{H}^{(2)}} \boldsymbol{p}_{\boldsymbol{H}^{(2)}} \\ &\quad -(\boldsymbol{I}_{\mathrm{N}} \otimes (\boldsymbol{H}^{(1)\mathrm{T}} \odot \boldsymbol{F}))^{\mathrm{H}} \tilde{\boldsymbol{y}}. \end{cases} \tag{27}$$

最后, 根据计算得到的$\boldsymbol{J}$和$\boldsymbol{g}$, 在每次迭代过程中对λ进行更新, 并采用Cholesky分解[16]并回代的方式对标准方程(17)求逆, 从而得到每一次迭代过程中的估计参数. 由于$\lambda > 0$, 标准方程(17)始终是一个对称正定的非奇异矩阵, 从而确保了$\Delta\boldsymbol{p}$是正确的下降方向. 当λ较大时, (17)式转化为$\Delta\boldsymbol{p} \simeq -(1/\lambda)\boldsymbol{g}$, $\Delta\boldsymbol{p}$是最陡下降方向上的一个短步长, 从而保证了算法的快速收敛; 当λ较小时, (17)式又近似转化为高斯-牛顿更新, 从而保证了算法的局部二次收敛. 目前已有多种方法对λ的具体取值进行研究, 本文采用具有代表性的文献[18]的方法对λ进行更新.

综上所述, 所提PARAFAC-LM算法具体流程如下.

步骤1 初始化$\boldsymbol{p}^{(1)}$, $\lambda^{(1)}$和$\boldsymbol{r}(\boldsymbol{p}^{(1)})$, 令$i=1$, $up=1$, $v=2$.

步骤2 若$up=1$, 计算$\boldsymbol{J}^{(i)}$和$\boldsymbol{g}^{(i)}$.

步骤3 计算$\Delta\boldsymbol{p}^{(i)} = -(\boldsymbol{J}^{(i)\mathrm{H}}\boldsymbol{J}^{(i)} + \lambda^{(i)}\boldsymbol{I})^{-1}\boldsymbol{g}^{(i)}$.

步骤4 更新$\boldsymbol{p}^{(i+1)} = \boldsymbol{p}^{(i)} + \Delta\boldsymbol{p}^{(i)}$.

步骤5 根据(12)式计算$\hat{\boldsymbol{H}}_{\mathrm{new}}^{(1)}$, $\hat{\boldsymbol{H}}_{\mathrm{new}}^{(2)}$.

步骤6 根据(26)式计算$\varphi^{(\mathrm{new})}(\boldsymbol{p}^{(i+1)})$.

步骤7 计算增益比[14]:

$$\rho = \frac{\varphi(\boldsymbol{p}^{(i)}) - \varphi^{(\mathrm{new})}(\boldsymbol{p}^{(i+1)})}{\varphi(\boldsymbol{p}^{(i)}) - \hat{\varphi}(\boldsymbol{p}^{(i)})},$$

其中

$$\begin{aligned} &\varphi(\boldsymbol{p}^{(i)}) - \hat{\varphi}(\boldsymbol{p}^{(i)}) \\ =& -\frac{1}{2}(\boldsymbol{J}^{(i)}\Delta\boldsymbol{p}^{(i)})^{\mathrm{H}}\boldsymbol{r}(\boldsymbol{p}^{(i)}) - \frac{1}{2}\boldsymbol{r}(\boldsymbol{p}^{(i)})^{\mathrm{H}}\boldsymbol{J}^{(i)}\Delta\boldsymbol{p}^{(i)} \\ & -\frac{1}{2}\|\boldsymbol{J}^{(i)}\Delta\boldsymbol{p}^{(i)}\|^2. \end{aligned}$$

步骤8 更新λ[18]:

若$\rho \geqslant 0$, 则$\boldsymbol{p}^{(i+1)}$更新正确, 令$\lambda^{(i+1)} = \lambda^{(i)}\max(1/3, 1-(2\rho-1)^3)$, $v=2$, $\hat{\boldsymbol{H}}_{(i)}^{(1)} = \hat{\boldsymbol{H}}_{\mathrm{new}}^{(1)}$, $\hat{\boldsymbol{H}}_{(i)}^{(2)} = \hat{\boldsymbol{H}}_{\mathrm{new}}^{(2)}$, $\varphi(\boldsymbol{p}^{(i)}) = \varphi^{(\mathrm{new})}(\boldsymbol{p}^{(i+1)})$, $up=1$; 否则, $\boldsymbol{p}^{(i+1)}$更新失败, 需要重新更新$\boldsymbol{p}^{(i+1)}$, 令$up=0$, $\lambda^{(i+1)} = v\lambda^{(i)}, v=2v$.

步骤9 $i=i+1$.

步骤10 回到步骤2直到程序收敛.

在第i次迭代中, 所提PARAFAC-LM算法的估计误差为

$$\phi^{(i)} = \|\boldsymbol{X} - (\boldsymbol{F} \odot \hat{\boldsymbol{H}}_{(i)}^{(2)})\hat{\boldsymbol{H}}_{(i)}^{(1)}\|_F, \tag{28}$$

其中, $\hat{\boldsymbol{H}}_{(i)}^{(1)}$和$\hat{\boldsymbol{H}}_{(i)}^{(2)}$分别是第$i$次迭代过程中关于$\boldsymbol{H}^{(1)}$和$\boldsymbol{H}^{(2)}$的估计结果. 若满足条件$|\phi^{(i)} - \phi^{(i-1)}| \leqslant \varepsilon$ (例如$\varepsilon = 10^{-6}$), 即可认为所提PARAFAC-LM算法达到收敛. 由于假设系统设计矩阵$\boldsymbol{F}$已知, LM算法结束后得到的估计结果$\hat{\boldsymbol{H}}^{(1)}$和$\hat{\boldsymbol{H}}^{(2)}$仅存在尺度模糊, 可通过规范化的方法[19,20]消除该尺度模糊. 所提PARAFAC-LM算法计算复杂度通过复乘法次数分析得到, 在每次迭代过程中所需要的总复乘法次数近似为$\mathcal{O}\left(\frac{1}{3}(MR+NR)^3\right)$. 因此, 所提算法总的计算复杂度取决于算法达到收敛时迭代的次数. 在后续的讨论中, 我们将会看到根据系统设计矩阵$\boldsymbol{F}$的不同选取, 所提算法均只需要少量的迭代次数就能够达到收敛.

5 仿真结果与讨论

仿真中, 假设用户数、中继数和基站天线数目均为4, 导频信号的长度$L=4$, 即$N=M=R=L=4$. 与文献[13]相同, 本文假设用户的发射功率P_S和中继的发送功率P_R相同. CSI的估计性能由归一化均方误差(normalized mean square error, NMSE)衡量, 定义为: $\mathrm{NMSE}^{\boldsymbol{H}^{(l)}} = \|\hat{\boldsymbol{H}}^{(l)} - \boldsymbol{H}^{(l)}\|_F^2/\|\boldsymbol{H}^{(l)}\|_F^2$, 其中$l \in \{1,2\}$. 所有仿真结果均由蒙特卡罗仿真2000次平均得到.

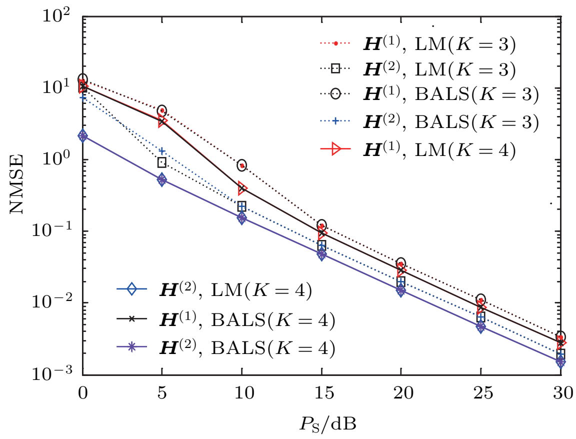

图2 (网刊彩色) 当$\boldsymbol{F}$为DFT矩阵时, 两种算法的信道NMSE性能比较

Fig. 2. (color online) When $\boldsymbol{F}$ is a DFT matrix, NMSE performance comparison.

当中继放大因子矩阵 $\boldsymbol{F}$ 为离散傅里叶变换 (discrete Fourier transform, DFT) 矩阵时, 图2为所提方法与BALS算法[12]在不同 K 值下估计性能的比较示意图. 由图2可知, 所提LM算法与BALS算法的估计结果随着 P_S 的增加而逐渐变好, 两者估计出的CSI有着近乎相同的估计精度. 由图2还可知, 与 $K=3$ 相比, 在 $K=4$ 时两种算法都有着更好的估计结果, 这是因为时间分集的增加使得系统获得了更多的信道相关知识, 从而提升了估计性能.

图3为 $K=3$, 中继放大因子矩阵 $\boldsymbol{F}$ 为随机矩阵时所提方法与BALS算法性能的比较结果示意图. 与图2类似, 图3中两种算法的估计精度曲线几乎是重合的. 图3还表明, 与 $\boldsymbol{F}$ 为DFT矩阵相比较, $\boldsymbol{F}$ 为随机矩阵时信道估计性能有所下降. 表1给出了本次实验中所提LM算法与BALS算法达到收敛所需迭代次数和CPU占用时间(单位: s). 从表1可以看到, 随着 P_S 值的增加, 两种算法的迭代次数逐渐降低. 表1还表明, 在中高 P_S 值时 (大于0 dB), 所提LM算法的迭代次数远小于BALS算法, 此时所提LM算法的CPU占用时间要少于BALS算法. 以 $P_S=30$ dB为例, BALS算法平均需要94次迭代才能收敛, 占用的CPU时间为0.0274 s; 而所提LM 算法只需要13次迭代就能够实现收敛, 占用的CPU时间仅为0.0089 s, 在收敛速度方面具有明显的优势.

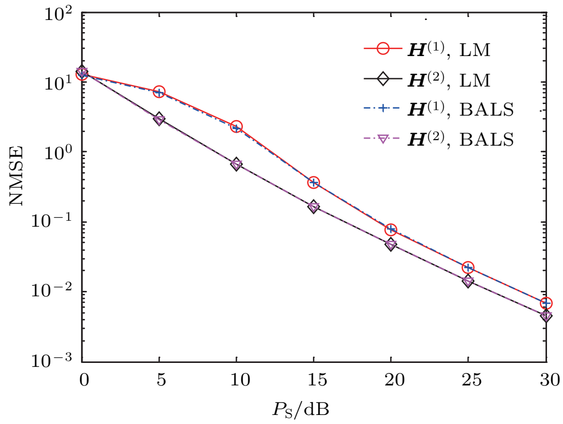

图3 (网刊彩色) 当 $\boldsymbol{F}$ 为随机矩阵时, 两种算法的信道NMSE性能比较

Fig. 3. (color online) When $\boldsymbol{F}$ is a random matrix, NMSE performance comparison.

表1 当 $\boldsymbol{F}$ 为随机矩阵时, 算法性能比较

Table 1. When $\boldsymbol{F}$ is a random matrix, mean number of iterations and CPU times till convergence by the LM and the BALS algorithms.

		P_S/dB						
		0	5	10	15	20	25	30
迭代次数	LM	42	34	23	17	14	13	13
	BALS	123	122	109	95	95	92	94
CPU 占用时间/s	LM	0.0373	0.0285	0.0177	0.0118	0.0096	0.0091	0.0089
	BALS	0.0325	0.0320	0.0297	0.0278	0.0237	0.0262	0.0274

下面分析天线相关性对所提LM算法性能的影响. $\boldsymbol{H}^{(1)}$ 和 $\boldsymbol{H}^{(2)}$ 为相关衰落信道模型的定义如下[21]:

$$\boldsymbol{H}^{(l)}=\boldsymbol{R}_{r_l}^{1/2}\boldsymbol{H}_w^{(l)}\boldsymbol{R}_{t_l}^{\mathrm{T}/2}, \tag{29}$$

其中, $l\in\{1,2\}$; $\boldsymbol{R}_{r_l}=\boldsymbol{R}_{r_l}^{1/2}\boldsymbol{R}_{r_l}^{\mathrm{H}/2}$, $\boldsymbol{R}_{t_l}=\boldsymbol{R}_{t_l}^{1/2}\boldsymbol{R}_{t_l}^{\mathrm{H}/2}$ 分别是 $\boldsymbol{H}_w^{(l)}$ 的发射和接收相关系数矩阵. 为了便于分析, 令 $\boldsymbol{R}_{r_l}=\boldsymbol{R}_{t_l}$, $\boldsymbol{R}_{r_l}(x,y)=\rho^{|x-y|}$, ρ 是一个小于1的归一化相关系数. $\boldsymbol{H}_w^{(l)}$ 中的元素是满足均值为0方差为1的独立同分布复高斯随机变量. $\boldsymbol{F}$ 为随机生成矩阵. 图4为所提方法在 $K=4$, 相关系数分别为 $\rho=0.2$(弱相关信道) 和 $\rho=0.8$(强相关信道) 情况下估计结果示意图. 图4表明, 随着 P_S 的增大, 不同相关衰落信道的NMSE均减小. 图4还表明, 对于信道 $\boldsymbol{H}^{(1)}$ 和 $\boldsymbol{H}^{(2)}$, 弱相关衰落信道的NMSE要好于强相关衰落信道情况下的估计结果. 这是因为信道矩阵的列之间相关性上升造成了拟合精度的降低, 从而对估计结果的准确性产生了影响.

最后, 给出当加载矩阵中存在近似共线性相关列情况下, 所提LM算法与BALS算法性能的比较. 假设放大因子矩阵 $\boldsymbol{F}$ 的选取为

$$\boldsymbol{F}=\begin{bmatrix}1 & \cos\alpha & 0 & 0\\ 0 & \sin\alpha & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}. \tag{30}$$

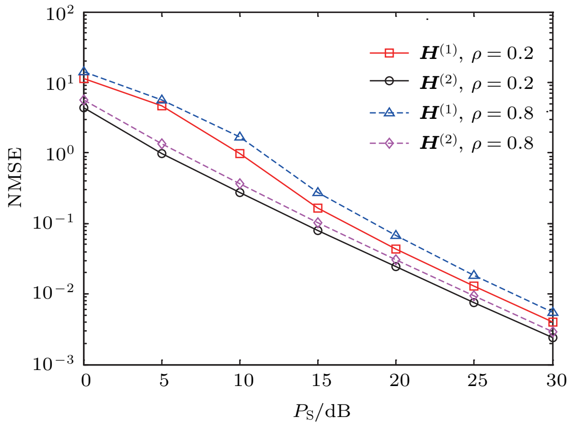

图4 (网刊彩色) 不同相关系数下, 信道NMSE性能

Fig. 4. (color online) NMSE versus P_S for correlated channels.

在(30)式中, α的取值决定了$\boldsymbol{F}$的前两列的共线性程度. 当$\alpha = \pi/60 \simeq 0.052$时, $\boldsymbol{F}$的前两列可以看作是近似共线. 图5为$P_S = 30$ dB 和 $K = 4$时, 两种算法接收信号NMSE和迭代次数关系示意图. 其中, 接收信号NMSE定义为$\text{NMSE}^{\boldsymbol{\mathcal{X}}} = \|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}^{(i)}\|_F^2/\|\boldsymbol{\mathcal{X}}\|_F^2$. 从图5可知, 接收信号的NMSE在BALS算法中需要1000次左右才能收敛, 而所提LM算法仅需20次即可达到收敛. 此时, 所提算法在收敛次数上优势明显. 表2给出了不同P_S情况下, 所提LM算法与BALS算法达到收敛所需迭代次数和CPU占用时间(单位: s). 表2表明, 所提LM算法不仅收敛次数方面表现稳定, 并且在各P_S值处占用的CPU时间均远小于BALS算法. 以$P_S = 30$ dB为例, 两种算法收敛的迭代次数分别为20和880, CPU占用时间分别为0.0182 s和0.2739 s, 到达收敛时LM算法的CPU占用时间还不到BALS算法的十分之一. 因此, 当加载矩阵存在共线性相关列时, 所提LM算法无论是收敛次数还是收敛时间上均优于BALS算法, 并且优势极为明显.

因此, 结合上述分析可知: 系统设计矩阵$\boldsymbol{F}$的改变, 会对BALS算法性能产生较大的影响; 而本文所提LM算法则表现稳健, 受$\boldsymbol{F}$改变的影响较小. 这也是本文采用LM算法来实现信道估计的根本原因.

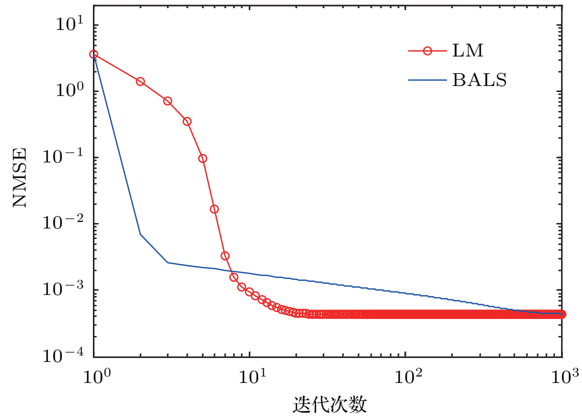

图5 (网刊彩色) 迭代次数与信号NMSE的关系

Fig. 5. (color online) NMSE versus iterations.

表2 当$\boldsymbol{F}$包含近似共线性相关列时, 算法性能比较

Table 2. When $\boldsymbol{F}$ is a almost collinear matrix, mean number of iterations and CPU times till convergence by the LM and the BALS algorithms.

		P_S/dB						
		0	5	10	15	20	25	30
迭代次数	LM	20	19	19	20	20	21	20
	BALS	959	797	736	729	771	821	880
CPU 占用时间/s	LM	0.021	0.0198	0.0185	0.0195	0.0197	0.0209	0.0182
	BALS	0.3043	0.2557	0.2343	0.2326	0.2417	0.256	0.2739

6 结 论

本文针对多用户上行AF中继系统, 提出了一种能够快速收敛的信道估计方法. 该方法通过在基站构造具有平行因子结构的张量模型, 并采用LM算法实现了系统中用户-中继信道和中继-基站信道的独立估计. 与已有BALS算法相比, 所提方法具有相似的估计精度. 当放大因子矩阵为随机或者存在近似共线性列时, 该方法表现稳定, 均能够快速实现收敛. 因此, 对于多用户上行AF中继系统, 所提方法具有一定的研究价值. 下一步的工作重点将是降低每次迭代过程中LM算法的计算复杂度.

参考文献

[1] Sanguinetti L, D'Amico A A, Rong Y 2012 *IEEE J. Sel. Areas Commun.* **30** 1331

[2] Hammerstrom I, Wittneben A 2007 *IEEE Trans. Wirel. Commun.* **6** 2798

[3] Rong Y 2010 *IEEE Commun. Lett.* **14** 390

[4] Munoz M O, Vidal J, Agustin A 2007 *IEEE Trans. Signal Process* **55** 2593

[5] Zhou J, Jiang H, Hisakazu K, Shao G F 2014 *Acta Phys. Sin.* **63** 140506 (in Chinese) [周杰, 江浩, 菊池久和, 邵根富 2014 物理学报 **63** 140506]

[6] Ma L, Liu S Z, Qiao G 2015 *Acta Phys. Sin.* **64** 154304 (in Chinese) [马璐, 刘凇佐, 乔钢 2015 物理学报 **64** 154304]

[7] Sidiropoulos N D, Giannakis G B, Bro R 2000 *IEEE Trans. Signal Process* **48** 810

[8] Kruskal J B 1977 *Linear Algebra. Appl.* **18** 95

[9] Xiao H L, Ouyang S, Nie Z P 2009 *Acta Phys. Sin.* **58** 3685 (in Chinese) [肖海林, 欧阳缮, 聂在平 2009 物理学报 **58** 3685]

[10] de Almeida A L F, Fernandes C A, Da Costa D 2013 *IEEE Signal Process. Lett.* **20** 697

[11] Du J H, Yuan C W, Hu Z W, Lin H Y 2015 *IEEE Commun. Lett.* **19** 1961

[12] Rong Y, Khandaker M R, Xiang Y 2012 *IEEE Trans. Wirel. Commun.* **11** 2224

[13] Du J H, Yuan C W, Zhang J B 2015 *IET Commun.* **9** 737

[14] De Almeida A L F, Favier G, Ximenes L R 2013 *IEEE Trans. Signal Process* **61** 1895

[15] Marquardt D 1963 *SIAM J. Appl. Math.* **11** 431

[16] Nion D, De Lathauwer L 2008 *IEEE Trans. Signal Process* **56** 5567

[17] Tomasi G, Bro R 2006 *Comp. Stat. Data Anal.* **50** 1700

[18] Madsen K, Nielsen H B, Tingleff O 2004 *Methods for Non-Linear Least Squares Problems* (2nd Ed.) (Copenhagen: Tech. Univ. Denmark) pp24–27

[19] Du J H, Yuan C W, Tian P, Lin H Y 2016 *IET Commun.* **10** 995

[20] Ximenes L R, Favier G, De Almeida A L F, Silva Y C 2014 *IEEE Trans. Signal Process* **62** 3604

[21] Shiu D, Foschini G, Gans M J, Kahn J 2000 *IEEE Trans. Commun.* **48** 502

物 理 学 报 Acta Phys. Sin. Vol. 65, No. 21 (2016) 210201

A fast algorithm with convergence for channel estimation in multi-user uplink amplify-and-forward relay system*

Lin He-Yun[1)] Yuan Chao-Wei[1)†] Du Jian-He[2)]

1) (*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*)

2) (*School of Information and Engineering, Communication University of China, Beijing 100024, China*)

(Received 21 June 2016; revised manuscript received 27 July 2016)

Abstract

Recently, tensor models (or multi-way arrays) play a vital role in many applications, such as wireless communication systems, blind source separation, machine learning, signal (audio, image, speech) processing, chemometrics, data mining, arithmetic complexity, environmental sciences, etc. Parallel factor (PARAFAC) analysis, also known as canonical polyadic decomposition, is a common name for low rank decomposition of tensors. A traditional way to fit the PARAFAC model is the alternating least squares (ALS) algorithm, which can transform a nonlinear optimization problem into some independent linear least squares problems. However, the ALS scheme for computing the decomposition of the tensor is known to converge slowly if one or some modes include nearly collinear columns. Particularly, if the collinearity is presented in all modes, the ALS will end in a “convergence bottleneck”.

Hence, it is necessary to develop a robust and fast algorithm to compute the decomposition of the tensor. In this paper, a novel channel estimation algorithm using the Levenberg Marquardt (LM) method based on a third-order tensor model is presented in a multi-user uplink amplify-and-forward (AF) relay system. As the relay nodes all operate with half-duplex mode to aid the transmission, the overall transmission period is partitioned into two transmission sub-processes. In the first transmission sub-process, the users transmit channel training sequence to the relay nodes. This stage requires time block once. During the second transmission sub-process, a set of diagonal amplifying factor matrices are utilized by the relay nodes to amplify the received data. Then, the relay nodes transmit each of the amplified data to the base station. This stage requires time blocks K times. With the help of the channel training sequence and the relay amplifying factor matrices, the received data at the base station can be stacked up into a third-order PARAFAC model. And then based on this tensor model an LM channel estimation algorithm is proposed to provide the individual channel state information of both user-to-relay and relay-to-base station channel links. As the channel sequence is transmitted by the users only once, the proposed scheme has a higher spectral efficiency than the case that the channel sequence is transmitted K times by the users. Numerical experiments are shown to demonstrate the efficacy of the proposed LM channel estimation algorithm.

The results are as follows. Firstly, the LM approach has the same channel estimation performance as the bilinear alternating least-squares method. Secondly, the proposed estimator yields much faster convergence speed when the relay amplifying factor matrix is a random matrix or a highly collinear one. Finally, the proposed scheme performs well in both independent identically distributed channels and correlated channels scenarios, which means that the proposed channel estimator can provide the robust and reliable feature for multi-user uplinks AF relay systems.

Keywords: tensor model, channel estimation, Levenberg Marquardt algorithm, amplify-and-forward relay

PACS: 02.10.Xm, 84.40.Ua, 43.60.Gk **DOI:** 10.7498/aps.65.210201

* Project supported by the National High Technology Research and Development Program of China (Grant Nos. 2015AA01A705, 2014AA01A701) and the Science Project of Communication University of China (Grant No. 3132016XNG1618).

† Corresponding author. E-mail: yuancw2000@bupt.edu.cn