Permuted Linear Model for Header-Free Communication via Symmetric Polynomials

Xuming Song, Hayoung Choi, and Yuanming Shi
School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
Email: {songxm, hchoi, shiym}@shanghaitech.edu.cn

***Abstract*—We present a linear model with an unknown permutation matrix for header-free communication in massive Internet-of-Things (IoT) networks, thereby supporting low-latency communication. Besides the header-free communication application, the permuted linear model also has many applications in matching and correspondence estimation problems. To solve this permuted linear system, the key idea is to convert it into a system of polynomial equations via power sum symmetric polynomials. In our proposed method, specific assumptions (e.g., low-rankness and sparsity) about the permuted linear model is not needed. The closed form of the solution is established for a special case. Computational results show that the proposed method achieves good performance.**

I. Introduction

The upcoming fifth-generation (5G) system brings a central challenge that need to support short-packet transmission, especially for the IoT applications [1]. In the scenario of short-packet transmission, the size of the control information (metadata) is comparable to the size of payload, which significantly affects the overall efficiency of the transmission with respect to energy, latency, and bandwidth cost. Typically, the metadata consists of preamble (e.g. channel signaling) and header (e.g. the identity information about sensor node). To address this issue, lots of research efforts have been paid on channel signaling overhead reduction. Blind equalization and identification [2] is a bandwidth efficient solution by eliminating training data and maximizing channel capacity for true information transmission. Another way to handle the inefficiency is to remove the identity information for each user when it transmits the true information [3] (so-called header-free communication). Unfortunately, the absence of identity information makes the desired true information recovery problem highly intractable. Specifically, in the massive IoT scenario, we assume that the sensors' identity information is a critical part and forms the bulk of the communicated data. As mentioned in [4] and references therein, differential updates, spatial correlation, and multi-stage collection are the cases that sensors are used to periodically reconstruct the spatial field. Hence, a significant gain in communication procedure can be achieved by removing the identity information. For an linear sensing system which senses a spatial field, let $\boldsymbol{x}$ denote the representation of the field in the n-dimensional basis. The observation can be interpreted as the matrix multiplication between $\boldsymbol{x}$ and a unique sampling matrix corresponding to the location where the sample is taken. The absence of the identity information can be represented as an unknown permutation matrix $\boldsymbol{\Pi}$. The desired true information recovery problem can be expressed as the following permuted linear model:

$$\boldsymbol{y} = \boldsymbol{\Pi} \boldsymbol{A} \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a known sample matrix, $\boldsymbol{y} \in \mathbb{R}^m$ is an observation vector collected by the sensing system, $\boldsymbol{x} \in \mathbb{R}^n$ is an unknown parameter, and $\boldsymbol{\Pi} \in \mathbb{R}^{m \times m}$ is an unknown permutation matrix. It means that the system only has incomplete information about the order of entries in observation $\boldsymbol{y}$. In other words, we might have access to all the entries of $\boldsymbol{y}$ but do not know which entries correspond to which locations within the vector $\boldsymbol{y}$. More details about this permuted linear model for the header-free communication will be presented in Section II.

This permuted linear model also has other potential applications. Firstly, consider a well-known Simultaneous Location and Mapping (SLAM) problem [5]. The SLAM problem is a classical problem in robotics that the robot is viewed as a mobile sensor and its main task is sampling the unknown environment meanwhile locating each sample point. The absence of the location information can be represented as an unknown permutation matrix. Secondly, consider a time domain sampling problem with presence of clock jitter [6]. We may desire the frequency domain signal from the time domain samples, but it is difficult to associate sampled time domain observations to the correct time indices with clock jitter. This uncertainty of the time indices can be viewed as an unknown permutation matrix. Thirdly, in image processing, the pose and correspondence estimation problem [7] also have a similar structure with formula (1). The camera capture a 3D object by a 2D image and we want to match some key points between two spaces. The capture procedure can be modeled as an unknown linear transformation called pose, and the unknown permutation can be viewed as correspondence between points of the two spaces.

Based on this permuted linear model, there are two issues need to be addressed. One is to recover the unknown permutation matrix $\boldsymbol{\Pi}$, the other is to recover the unknown vector $\boldsymbol{x}$. Pananjady at al. [3] first considered the permutation matrix recovery problem in the linear regression model. They estab-

This work was partly supported by the National Nature Science Foundation of China under Grant No. 61601290, and the Shanghai Sailing Program under Grant No. 16YF1407700.

lished sharp conditions on signal-to-noise ratio, sample size m, and dimension n with respect to the exact or approximate recoverability of permutation matrix $\boldsymbol{\Pi}$ and showed that the maximum likelihood estimator of $\boldsymbol{\Pi}$ was NP-hard to compute. A similar problem, compressed sensing with unknown sensor permutation was researched in [8]. They studied a convex relaxation of the problem and proposed a branch-and-bound based algorithm to find the unknown permutation matrix $\boldsymbol{\Pi}$ under the condition that $\boldsymbol{x}$ has sparse structure. There are also some prior work on recovery problem for the unknown vector $\boldsymbol{x}$. In [9], band-limited signals reconstruction problem with unknown sample locations was investigated. In [10], the condition of exact recovery of $\boldsymbol{x}$ with random sample matrix $\boldsymbol{A}$ was established, but they failed to give a practical algorithm. [11] proposed a geometrical algorithm to recover the vector $\boldsymbol{x}$ and studied the uniqueness of the solution with specific sample matrix $\boldsymbol{A}$. In [12], a characterization of the minimax error rate for the multivariate version of this problem was provided based on denoising aspect, and also an exact recovery algorithm based on spectral theory was proposed for the noiseless case under a rank condition.

In this paper, we present a signal recovery problem in a massive IoT network that each senor node can communicate without identity information. An unknown permutation matrix introduced by the absence of identity information makes the signal recovery problem become hard. Inspired by the symmetric property of the symmetric polynomial, we use symmetric polynomials to remove the impact of the unknown permutation matrix which gives us a new perspective to handle this problem. Specifically, in order to solve this signal recovery problem more effectively, it is possible to convert it into a system of polynomial equations via power sum symmetric polynomials. Compared with the existing methods, our method do not need any additional assumptions about the permuted linear model, such as signal structure (i.e. sparsity) and rank condition (i.e. Rank($\boldsymbol{A}$) $\leq$ Rank($\boldsymbol{x}$)). Besides, our proposed method does not need to exhaustively search all the permutation matrix which is not practical in applications with large m. The computational results show that the time cost of our proposed method is not sensitive to the value m, and the performance is efficient when m is very large. It meets the characteristic of massive IoT network applications. We also derive the closed form solution of $\boldsymbol{x}$ when $n = 1$.

II. System Model and Preliminaries

A. System Model

We consider a massive IoT network with m sensor nodes s_i, $i \in \{1, 2, \cdots, m\}$ and a fusion center. One typical application of the IoT networks is to monitor the industrial environments, e.g., temperature, humidity, and pressure. We consider the scenarios, where each sensor node s_i takes the linear measurement [13] $y_i = \boldsymbol{A}_i\boldsymbol{x}$, where $y_i \in \mathbb{R}^1$ is the observation value at sensor node s_i, $\boldsymbol{A}_i^\top \in \mathbb{R}^n$ is a known sample vector, and $\boldsymbol{x} \in \mathbb{R}^n$ is an unknown parameter vector representing the environment. All the sensor nodes send the observations to the fusion center for further data processing. However, in massive IoT networks with short packet communications, it is typically the case that the number of bits of observations transmitted by the sensor node is exceeded by the number of bits transmitted to identify itself at the fusion center [4]. In order to reduce the communication overhead for identifying, head-free communication turns out to be a promising approach [3]. With the absence of identity information at the fusion center, the observed vector $\boldsymbol{y}$ can be written as follows:

$$\boldsymbol{\Pi}^\top \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{2}$$

where

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{e}_{\pi_1} \\ \boldsymbol{e}_{\pi_2} \\ \vdots \\ \boldsymbol{e}_{\pi_m} \end{bmatrix} \in \mathbb{R}^{m \times m}, \quad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_m \end{bmatrix} \in \mathbb{R}^{m \times n},$$

$$\text{and } \boldsymbol{y} = [y_{\pi_1}, y_{\pi_2}, \cdots, y_{\pi_m}]^\top \in \mathbb{R}^m.$$

Here, $\boldsymbol{x} \in \mathbb{R}^n$ is the unknown vector parameter to be recovered at the fusion center and $\boldsymbol{\Pi}$ is an unknown permutation matrix. Specifically, π_j denote the image of an element j under the permutation $\boldsymbol{\Pi}$, and $\boldsymbol{e}_{\pi_j}^\top \in \mathbb{R}^m$ is the basis vector whose entries are all zeros except π_j-th entry equals to 1. Our goal is to efficiently recover $\boldsymbol{x}$ with the observation $\boldsymbol{y}$ and measurement matrix $\boldsymbol{A}$. Note that (2) can be rewritten as (1). In this paper, we mainly consider the noiseless case and further provide a preliminary idea to handle the noisy case.

B. Preliminaries about Symmetric Polynomial

Here, we present the basic concepts about symmetric polynomials. Let $\mathbb{N}$ be the set of nonnegative integers. The set of all polynomials in $x_1, \ldots, x_n$ with real coefficients is denoted $\mathbb{R}[x_1, \ldots, x_n]$. A polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$ is *symmetric* if $f(x_{\pi_1}, \ldots, x_{\pi_n}) = f(x_1, \ldots, x_n)$ for every possible permutation $x_{\pi_1}, \ldots, x_{\pi_n}$ of the variables $x_1, \ldots, x_n$. Given variables $x_1, \ldots, x_n$, we define the *elementary symmetric functions* $\sigma_1, \ldots, \sigma_n \in \mathbb{R}[x_1, \ldots, x_n]$ by the formulas

$$\begin{aligned}
\sigma_0(x_1, \ldots, x_n) &= 1, \\
\sigma_1(x_1, \ldots, x_n) &= x_1 + \cdots + x_n, \\
&\vdots \\
\sigma_r(x_1, \ldots, x_n) &= \sum_{1 \leq i_1 < i_2 < \cdots < i_r \leq n} x_{i_1} x_{i_2} \cdots x_{i_r}, \\
&\vdots \\
\sigma_n(x_1, \ldots, x_n) &= x_1 x_2 \cdots x_n.
\end{aligned}$$

That is, σ_r is the sum of all monomials that are products of r distinct variables. The *power sum symmetric polynomial of degree* k in n variables $x_1, \ldots, x_n$, written as p_k, is the sum of all k-th powers of the variables, i.e.,

$$p_k(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i^k.$$

It is well known that every symmetric polynomial in $\mathbb{R}[x_1, \dots, x_n]$ can be written as a polynomial in the power sums $p_1, \dots, p_n$ [14]. So, every elementary symmetric function can be written in terms of power sums, and vice versa.

Newton's identities, also known as the *Newton-Girard formulae* provide relations between power sums and elementary symmetric polynomials as follows:

$$\begin{aligned} \sigma_1 &= p_1 \\ 2\sigma_2 &= \sigma_1 p_1 - p_2 \\ &\vdots \\ k\sigma_k &= \sum_{i=1}^{k} (-1)^{i-1} \sigma_{k-i} p_i \end{aligned}$$

for all nonzero $k \in \mathbb{N}$. These equations allow to recursively express the σ_k in terms of the $p_1, \dots, p_k$. Conversely, p_k can be recursively expressed in term of the $\sigma_1, \dots, \sigma_k$. For more information, please refer to [15], [16]. These identities relate to sums of powers of roots of a polynomial with the coefficients of the polynomial. An explicit expression for elementary symmetric functions $\sigma_1, \dots, \sigma_n$ in terms of sums $p_1, \dots, p_n$ and, vise versa, are given by well-known *Waring formulas* [17].

III. Main Results

In this section, we first show how to convert the permuted linear model (1) into a system of polynomial equations. Based on this transformation, a closed form for $n = 1$ is derived. We also give a preliminary idea that how to handle the noisy case.

Denote the set of all $m \times m$ permutation matrices as $\mathcal{P}_m$. Let $\boldsymbol{x}$ denote the column vector of the variables $x_1, \dots, x_n$, i.e.,

$$\boldsymbol{x} = \begin{bmatrix} x_1, & \cdots, & x_n \end{bmatrix}^\top.$$

Lemma 1. *If $\boldsymbol{x} \in \mathbb{R}^n$ is a solution to $\boldsymbol{y} = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x}$ for given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{\Pi} \in \mathcal{P}_m$, then $\boldsymbol{x}$ is a solution to the polynomial equations,*

$$p_k(\boldsymbol{A}\boldsymbol{x}) = p_k(\boldsymbol{y}) \quad \text{for each } k \in \mathbb{N}, \tag{3}$$

where p_k is the power sum symmetric polynomial of degree k.

Proof. Suppose that $\boldsymbol{x}$ is a solution to $\boldsymbol{y} = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x}$ for some $\boldsymbol{\Pi} \in \mathcal{P}_m$. Since the power sum p_k is a symmetric function, the function value is independent of the arrangement of entries of $\boldsymbol{y}$. Thus, $p_k(\boldsymbol{A}\boldsymbol{x}) = p_k(\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x}) = p_k(\boldsymbol{y})$ for all $k \in \mathbb{N}$. □

That is to say, the set of all solutions to the system of polynomials equations (3) includes all solutions to the permuted linear system $\boldsymbol{y} = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x}$ for any permutation matrix $\boldsymbol{\Pi} \in \mathcal{P}_m$.

Lemma 2. *If column vectors $\boldsymbol{y} = [y_1, \dots, y_m]^\top$, $\hat{\boldsymbol{y}} = [\hat{y}_1, \dots, \hat{y}_m]^\top \in \mathbb{R}^m$ satisfy that*

$$p_k(\hat{\boldsymbol{y}}) = p_k(\boldsymbol{y}) \quad \text{for all } k = 1, \dots, m, \tag{4}$$

then there exists a permutation matrix $\boldsymbol{\Pi} \in \mathcal{P}_m$ such that $\hat{\boldsymbol{y}} = \boldsymbol{\Pi}\boldsymbol{y}$.

Proof. Recall that a monic polynomial is a univariate polynomial in which the leading coefficient (the nonzero coefficient of highest degree) is equal to 1. Let f and $\hat{f}$ be degree m monic polynomial functions whose roots consist of entries of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, respectively. Since $p_k(\hat{\boldsymbol{y}}) = p_k(\boldsymbol{y})$, $k = 1, \dots, m$, by Newton's identities it follows that $\sigma_k(\hat{\boldsymbol{y}}) = \sigma_k(\boldsymbol{y})$, $k = 1, \dots, m$. So, the coefficients of two polynomials f and $\hat{f}$ are identical. Thus, the roots of f and $\hat{f}$ are identical, i.e., $\{\hat{y}_1, \dots, \hat{y}_m\} = \{y_1, \dots, y_m\}$. Therefore, $\hat{\boldsymbol{y}} = \boldsymbol{\Pi}\boldsymbol{y}$ for some $\boldsymbol{\Pi} \in \mathcal{P}_m$. □

Denote the set of solutions of (1) as

$$\mathcal{S} = \bigcup_{\boldsymbol{\Pi} \in \mathcal{P}_m} \{\boldsymbol{x} \in \mathbb{R}^n \ | \boldsymbol{y} = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x}\}.$$

Theorem 1. *For each $j \in \mathbb{N}$, let $\mathcal{S}_j := \{\boldsymbol{x} \in \mathbb{R}^n : p_k(\boldsymbol{A}\boldsymbol{x}) = p_k(\boldsymbol{y}),\ k = 1, \dots, j\}$. Then, for any given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{y} \in \mathbb{R}^m$, it holds that*

$$\mathcal{S}_1 \supseteq \cdots \supseteq \mathcal{S}_m = \mathcal{S}_{m+1} = \cdots = \mathcal{S}.$$

That is, solving the permuted linear model (1) *is equivalent to solving the following system of polynomial equations*

$$p_k(\boldsymbol{A}\boldsymbol{x}) = p_k(\boldsymbol{y}),\ k = 1, \dots, m. \tag{5}$$

Proof. Clearly, $\mathcal{S}_i \subseteq \mathcal{S}_j$ for $i \geq j$. If $\boldsymbol{x}$ is a solution of $\boldsymbol{y} = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x}$ for some $\boldsymbol{\Pi} \in \mathcal{P}_m$, then by Lemma 1 it follows that $\boldsymbol{x} \in \mathcal{S}_j$ for all $j \in \mathbb{N}$. Thus $\mathcal{S} \subseteq \mathcal{S}_j$ for all $j \in \mathbb{N}$. If $\boldsymbol{x} \in \mathcal{S}_m$, then by Lemma 2 it follows that $\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ for some $\boldsymbol{\Pi} \in \mathcal{P}_m$. □

Note that there are possibly more than one solution with the corresponding permutation matrix. Even though $\boldsymbol{x}$ is unique, the corresponding permutation matrices may possibly not be unique. The unique recoverability of $\boldsymbol{x}$ for the permuted linear model (1) was studied in [10]. Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be a given matrix with i.i.d. random entries drawn from an arbitrary continuous probability distribution f over $\mathbb{R}$. If $m \geq 2n$, then with probability 1, $\boldsymbol{x}$ can be uniquely recovered. Hence, if m is sufficiently larger than n, by Theorem 1, so the system of the polynomial equations (5) has a unique solution. Computational results show that the number of polynomial equations which is required to get the unique solution may be much smaller then m, which is in fact $n+1$ (see Fig. 2). Since $\dim(\mathcal{S}_1) = n-1$, we guess that it holds $\dim(\mathcal{S}_j) - 1 = \dim(\mathcal{S}_{j+1})$ for $j = 1, \dots, n-1$, implying $\dim(\mathcal{S}_n) = 0$, which means $\mathcal{S}_n$ has finitely many points. Then an additional polynomial equation possibly makes $\mathcal{S}_{n+1}$ have the unique solution. However, the mathematical proof is still open. Here, $\dim(\mathcal{S}_j)$ is the degree of the affine Hilbert polynomial of the corresponding ideal. For more information, please refer to [14].

Example 1. *Consider the problem*

$$\begin{bmatrix} -3 \\ 41 \\ -18 \\ 29 \\ 20 \end{bmatrix} = \prod \begin{bmatrix} -5 & 2 \\ 7 & 1 \\ -2 & 5 \\ 6 & -4 \\ 10 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{6}$$

Taking p_1 on the both side of the equation, we have

$$\begin{aligned} & p_1(\boldsymbol{y}) = p_1(\boldsymbol{\Pi A x}) = p_1(\boldsymbol{A x}) \\ & \Rightarrow 69 = 16x_1 + 5x_2, \end{aligned} \tag{7}$$

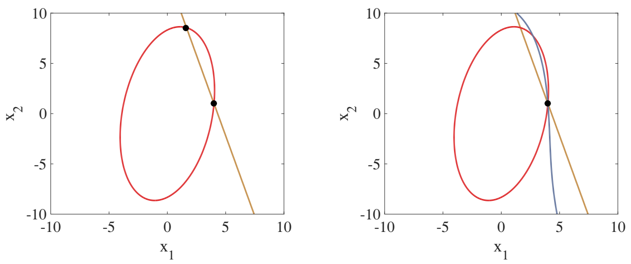
which has infinitely many solutions. So, we take p_2 on the both side to have the following polynomial equation.

$$\begin{aligned} & p_2(\boldsymbol{y}) = p_2(\boldsymbol{\Pi A x}) = p_2(\boldsymbol{A x}) \\ & \Rightarrow 3255 = 214x_1^2 - 54x_1x_2 + 47x_2^2. \end{aligned} \tag{8}$$

The system of two polynomial equations have two distinct solutions, which is still not unique (see Fig. 1(a)). One more power sum polynomial, p_3, provides the following equation

$$\begin{aligned} & p_3(\boldsymbol{y}) = p_3(\boldsymbol{\Pi A x}) = p_3(\boldsymbol{A x}) \\ & 95451 = 1426x_1^3 + 225x_1^2x_2 + 129x_1x_2^2 + 71x_2^3. \end{aligned} \tag{9}$$

Using simple algebra one can find the unique solution to the system of three polynomial equations, which is $x_1 = 4, x_2 = 1$ (see Fig. 1(b)).



(a) $\mathcal{S}_2$ (b) $\mathcal{S}_3$

Fig. 1. There are two distinct intersections for two curves (7) and (8), but only one intersection for three curves (7), (8), (9).

Now we find the explicit form of the system of polynomial equations. For $\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, denote $|\boldsymbol{\alpha}| := \alpha_1 + \ldots + \alpha_n$. Consider $\boldsymbol{y} = \boldsymbol{\Pi A x}$, where

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \ \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \ \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

Taking p_k on the both sides of the equation $\boldsymbol{y} = \boldsymbol{\Pi A x}$, we have the system of polynomial equations is of the form

$$b_k = \sum_{|\boldsymbol{\alpha}|=k} c_{\boldsymbol{\alpha}} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad \text{for } k \in \mathbb{N},$$

where

$$b_k = \sum_{j=1}^m y_j^k, \quad c_{\boldsymbol{\alpha}} = \sum_{j=1}^m k! \frac{a_{j1}^{\alpha_1} a_{j2}^{\alpha_2} \cdots a_{jn}^{\alpha_n}}{\alpha_1! \alpha_2! \cdots \alpha_n!}.$$

For instance, assume that $\boldsymbol{y} = \boldsymbol{\Pi A x}$ has a solution, $\boldsymbol{x} \in \mathbb{R}^1$, for given $\boldsymbol{A} \in \mathbb{R}^{m \times 1}$ and $\boldsymbol{y} \in \mathbb{R}^m$, and an unknown permutation matrix $\boldsymbol{\Pi} \in \mathcal{P}_m$. Then the explicit form of the solution is

$$x = \frac{\sum_{i=1}^m y_i}{\sum_{i=1}^m a_{i1}},$$

provided that $\sum_{i=1}^m a_{i1} \neq 0$. However, it is possible that the denominator is zero. Since $\sum_{i=1}^m a_{i1}^2 \neq 0$ for $\boldsymbol{A} \neq \boldsymbol{0}$, alternatively, one can have

$$|x| = \left(\frac{\sum_{i=1}^m y_i^2}{\sum_{i=1}^m a_{i1}^2} \right)^{\frac{1}{2}}.$$

Moreover, it is possible to find closed forms of the system of polynomials for $n = 2, 3, 4$. However, by Abel-Ruffini Theorem there are no closed forms for $n \geq 5$.

For the noisy case $\boldsymbol{y} = \boldsymbol{\Pi A x} + \boldsymbol{w}$, if the noise $\boldsymbol{w}$ is i.i.d. Gaussian, we can find the Maximum Likelihood Estimator (MLE) using least squares method by minimizing $\|\boldsymbol{y} - \boldsymbol{\Pi A x}\|_2$ as follows:

$$(\widehat{\boldsymbol{\Pi}}, \hat{\boldsymbol{x}}) = \underset{\boldsymbol{\Pi} \in \mathcal{P}_m, \ \boldsymbol{x} \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{\Pi A x}\|_2^2. \tag{10}$$

Taking p_k on the both sides of the equation $\boldsymbol{y} - \boldsymbol{w} = \boldsymbol{\Pi A x}$:

$$\sum_{j=1}^m (y_j - w_j)^k = \sum_{j=1}^m (a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jn}x_n)^k.$$

So the system of polynomial equations has the form

$$b_k + h_k(\boldsymbol{w}) = \sum_{|\boldsymbol{\alpha}|=k} c_{\boldsymbol{\alpha}} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad \text{for } k \in \mathbb{N}, \tag{11}$$

where

$$h_k(\boldsymbol{w}) := h_k(w_1, \ldots, w_m) = \sum_{j=1}^m \sum_{\ell=1}^k (-1)^k \binom{k}{\ell} y_j^{k-\ell} w_j^k,$$

which is a polynomial in $\mathbb{R}[w_1, \ldots, w_m]$. Then the least squares optimization problem (10) can be converted into the polynomial optimization problem as follows:

$$\begin{aligned} \hat{\boldsymbol{x}} = & \underset{\boldsymbol{x} \in \mathbb{R}^n, \ \boldsymbol{w} \in \mathbb{R}^m}{\operatorname{argmin}} \|\boldsymbol{w}\|_2 \\ \text{s.t.} & \sum_{|\boldsymbol{\alpha}|=k} c_{\boldsymbol{\alpha}} x_1^{\alpha_1} \cdots x_n^{\alpha_n} - h_k(\boldsymbol{w}) = b_k \quad \text{for } k \in \mathbb{N}. \end{aligned} \tag{12}$$

The polynomial optimization problem and its computational algorithms have been extensively studied in [18] and references therein.

IV. Computational Results

Although the brute force algorithm in [10] can be used to solve the permuted linear system [10], it is impractical when m is very large. We have already shown that a permuted linear model can be converted into a system of polynomial equations. Assume that the permuted linear system (1) has a unique solution. Our proposed algorithm is presented as follows.

(i) Construct polynomial equations: $p_k(\boldsymbol{y}) = p_k(\boldsymbol{A x})$ for $k \leftarrow 1, 2, \ldots, n$.
(ii) Solve this system of polynomial equations.
(iii) If the solution is unique, stop.
(iv) Otherwise, $k \leftarrow k + 1$, check if solutions satisfies one additional equation $p_k(\boldsymbol{A x}) = p_k(\boldsymbol{y})$. Then perform (iii).

Although there are various methods for solving a system of polynomial equations, only homotopy continuation method is used in this paper. There are several numerical tools for homotopy continuation method, such as PHCpack, Hom4PS and Bertini. In this paper, a MATLAB interface to the numerical homotopy continuation tool, called Bertini, is used to solve the system of polynomial equations numerically [19]. In Fig. 2, it shows that $n+1$ number of the equations is enough to get the unique solution for $n = 2, 3, \ldots, 6$ and $m = 3n$, respectively. In Fig. 3, the time cost is calculated with respect to m when $n = 3$. For lager m, the brute force approach is impractical. So, we only compare our proposed approach with the brute force approach when $m = 6, 7, \ldots, 11$. We also do simulations for very large m, and it shows that our proposed approach performs well and the time cost is linear increase with respect to m.

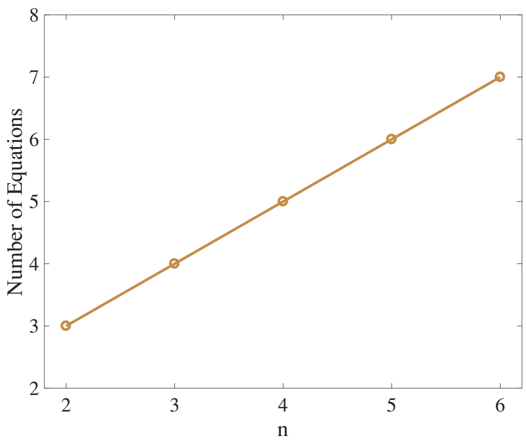


Fig. 2. The number of equations needed to get the unique solution.

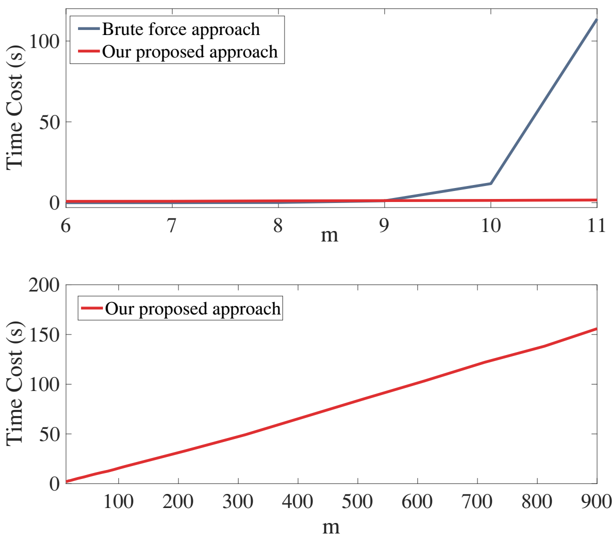


Fig. 3. The computational time cost with respect to m.

V. Conclusion

In this paper, we presented a permuted linear model for header-free communication in massive IoT networks. We converted the permuted linear model into a system of polynomial equations for efficiently solving it. This transformation also brings new perspective and potential opportunities to handle the permuted linear model. Our proposed method does not need specific assumptions about the permuted linear model and can be widely extended to other applications. The empirical results showed that our method also works efficiently when m is large compared with the brute force algorithm.

References

[1] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sept., 2016.

[2] Z. Ding and Y. Li, *Blind equalization and identification*. CRC press, 2001.

[3] A. Pananjady, M. J. Wainwright, and T. A. Courtade, "Linear regression with shuffled data: Statistical and computational limits of permutation recovery," *IEEE Trans. Inf. Theory*, to appear, 2017.

[4] L. Keller, M. J. Siavoshani, C. Fragouli, K. Argyraki, and S. Diggavi, "Joint identity-message coding," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1083–1093, Sept., 2010.

[5] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.

[6] A. v. Balakrishnan, "On the problem of time jitter in sampling," *IEEE Trans. Inf. Theory*, vol. 8, no. 3, pp. 226–236, Apr., 1962.

[7] P. David, D. Dementhon, R. Duraiswami, and H. Samet, "Softposit: Simultaneous pose and correspondence determination," *Int. J. Comput. Vision*, vol. 59, no. 3, pp. 259–284, 2004.

[8] V. Emiya, A. Bonnefoy, L. Daudet, and R. Gribonval, "Compressed sensing with unknown sensor permutation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, May, 2014, pp. 1040–1044.

[9] P. Marziliano and M. Vetterli, "Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3462–3471, Dec., 2000.

[10] J. Unnikrishnan, S. Haghighatshoar, and M. Vetterli, "Unlabeled sensing: Solving a linear system with unordered measurements," in *Proc. 53rd Annual Allerton Conf. Commun., Control, Comput. (Allerton)*, Sept., 2015, pp. 786–793.

[11] G. Elhami, A. Scholefield, B. B. Haro, and M. Vetterli, "Unlabeled sensing: Reconstruction algorithm and theoretical guarantees," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, Mar., 2017, pp. 4566–4570.

[12] A. Pananjady, M. J. Wainwright, and T. A. Courtade, "Denoising linear models with permuted data," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June, 2017, pp. 446–450.

[13] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Dec., 2008.

[14] D. A. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer-Verlag New York, Inc., 2007.

[15] J. A. Eidswick, "A proof of newton's power sum formulas," *Am. Math. Monthly*, vol. 75, no. 4, pp. 396–397, Apr., 1968.

[16] D. G. Mead, "Newton's identities," *Am. Math. Monthly*, vol. 99, no. 8, pp. 749–751, Oct., 1992.

[17] H. W. Gould, "The Girarad-Waring power sum formulas for symmetric functions and fibonacci sequences," *Fibonacci Quart*, vol. 37, no. 2, pp. 135–140, May, 1999.

[18] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[19] D. J. Bates, A. J. Newell, and M. Niemerg, "Bertinilab: A matlab interface for solving systems of polynomial equations," *Numer. Algorithms*, vol. 71, no. 1, pp. 229–244, Jan., 2016. [Online]. Available: https://doi.org/10.1007/s11075-015-0014-6