

Article

# A Greedy Pursuit Hierarchical Iteration Algorithm for Multi-Input Systems with Colored Noise and Unknown Time-Delays

Ruijuan Du and Taiyang Tao \*

School of Intelligent Manufacturing, Nanyang Institute of Technology, Nanyang 473004, China; dujuan@nyist.edu.cn

\* Correspondence: tyt@nyist.edu.cn

**Abstract:** This paper focuses on the joint estimation of parameters and time delays for multi-input systems that contain unknown input delays and colored noise. A greedy pursuit hierarchical iteration algorithm is proposed, which can reduce the estimation cost. Firstly, an over-parameterized approach is employed to construct a sparse system model of multi-input systems even in the absence of prior knowledge of time delays. Secondly, the hierarchical principle is applied to replace the unknown true noise items with their estimation values, and a greedy pursuit search based on compressed sensing is employed to find key parameters using limited sampled data. The greedy pursuit search can effectively reduce the scale of the system model and improve the identification efficiency. Then, the parameters and time delays can be estimated simultaneously while considering the known orders and found locations of key parameters by utilizing iterative methods with limited sampled data. Finally, some simulations are provided to illustrate the effectiveness of the presented algorithm in this paper.

**Keywords:** large-scale systems; time delays; parameter estimation; system identification; hierarchical algorithm



**Citation:** Du, R.; Tao, T. A Greedy Pursuit Hierarchical Iteration Algorithm for Multi-Input Systems with Colored Noise and Unknown Time-Delays. *Algorithms* **2023**, *16*, 374. <https://doi.org/10.3390/a16080374>

Academic Editor: Christos D. Zaroliagis

Received: 5 July 2023

Revised: 28 July 2023

Accepted: 31 July 2023

Published: 4 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

System identification is a crucial process in modern control theory that involves determining the mathematical model of a system based on its input–output measurement data. The system identification is summarized and extended in references [1,2]. This modeling and estimation technique help engineers to estimate parameters that characterize system behavior, which can be used to design controllers for various applications. An efficient identification method is very important. The performance analysis of some algorithms for system identification is presented in [3], and some new identification methods based on multi-innovation theory are presented in [4]. Moreover, this technique has numerous practical applications across different fields such as aerospace engineering, robotics, and chemical processing plants. With the development of modern science and technology, the objects faced by system identification become increasingly complex. Those system models are defined as large-scale systems such as complex networks, neural networks, and artificial intelligence [5,6]. The parameter estimation of large-scale systems is an important research direction in system identification, where the reduced-order and decomposition are effective methods for identification of large-scale systems [7,8]. For instance, the neural network or artificial intelligence, which can be regarded as a class of typical large-scale systems, typically employ millions of interconnected neurons to simulate the brain. Similarly, the complex dynamic network control systems, characterized by a significant number of nodes and links, can also be considered as large-scale systems [9,10]. In conclusion, system identification plays an essential role in modern control theory by providing accurate mathematical models for complex systems. It allows us to predict outputs accurately while

improving overall performance and efficiency across various systems. However, the difficulty of identification will increase with the increasingly dimensionality of the large-scale systems, which is a challenge that has to be faced in the future [11].

Large-scale systems usually contain a large number of variables. When these systems have unknown input time delays, the model used to describe them often includes redundant information. The identification problem for large-scale systems is typically divided into two stages: first, test measures are taken to estimate the orders and time delays; second, some identification algorithms are used to estimate the parameters. Therefore, traditional methods require prior knowledge such as system orders and time delays before estimating parameters. For example, the expectation maximization approach is used to estimate the parameters when the system orders and time-delay are known [12], the Bayesian method is used to estimate the system when the time-delay is known [13], and so does reference [14]. The least squares (LS) algorithms are widely used in large-scale system identification due to their fast parameter convergence rate and high estimation accuracy. The hierarchical LS iterative method is used to estimate the input nonlinear system [15,16], and the recursive LS estimation method is applied to identify the output nonlinear systems [17]. However, it should be noted that applying these methods directly can be difficult because there may be coupling items within large-scale systems. It is necessary to decouple the system before parameter estimation. A common approach is to decompose large-scale systems into some multi-variable subsystems; among these subsystems, multiple-input multiple-output (MIMO) systems are a typical class of multi-variable system. The MIMO systems can be decomposed into several single-input single-output (SISO) subsystems using decomposition techniques. Then, each of those subsystems can be identified one by one with enough sampled data [18], which comes with a huge computational cost. Another alternative approach for decomposing MIMO systems is to break them down into several multiple-input single-output (MISO) systems.

The identification problem of MISO systems has been extensively studied, and a large number of effective research results are given. Along with the emergence of compressed sensing (CS) in recent years, reference [19] gives the relevant basic of CS theory, reference [20] mainly introduces the principle and application of CS, and reference [21] presents the convergence of an interesting orthogonal matching pursuit (OMP) algorithm. The CS theory provides a new idea and some useful methods for system identification. For MISO systems that contain unknown input time delays, the parameter vector to be identified of the system that is over-parameterized has high dimension and sparse characteristics since the information vector contains unknown input delays. The CS is an effective idea to solve the parameter estimation for sparse systems. Many scholars have carried out research in related fields and achieved a series of results. Inspired by the greedy search idea of the CS, some parameter identification algorithms for output error models were designed by combining the auxiliary model and the hierarchical identification principle. For MISO output error systems with unknown time-delays, reference [22] combines a high dimensional and sparse estimation model with an auxiliary identification model to present the auxiliary model iterative algorithm, which has less computational cost compared with the auxiliary model least squares iterative, and reference [23] gives an auxiliary gradient pursuit iterative algorithm using the gradient search principle to further reduce the amount of computation. An iterative algorithm has been applied to achieve the estimation of time delays and parameters for multivariate systems output error systems by combining basis pursuit and auxiliary models in [24]. For MISO finite impulse response systems with input time delays, reference [25] combines the gradient search and matching pursuit method to give a gradient pursuit iterative algorithm and realize the parameters and time-delay estimation, and the gradient pursuit iterative algorithm can reduce the computational complexity compared with the traditional method. For nonlinear systems, a block-oriented nonlinear system was parameterized into a linear identification model including a sparse parameter vector, and the parameter estimation algorithm was presented by combining the matching pursuit idea based on CS, the auxiliary model, and key variable separation

technology in [26]. For nonlinear systems containing colored noise in general, an OMP iterative method was presented to obtain the true parameters and time delays of an input Hammerstein nonlinear system in [27].

In this article, the kernel matrix is constructed to normalize the information matrix and satisfy the application conditions of compressed sensing, and a hierarchical iterative algorithm based on a greedy pursuit search idea is investigated to realize the parameters and time-delay estimation effectively at the same time [28]. The main innovations in this article are listed as follows.

- The multi-variable systems model is recast based on the framework of CS by using the hierarchical identification principle.
- The unknown true internal noise items of the recast sparse model in the presented algorithm are replaced by their estimation values according to the hierarchical principle.
- The presented algorithm constructs a kernel matrix to find the locations of key parameters and reduce the estimated dimension and computational cost by using greedy pursuit search, in which only limited sampled data are used.
- The parameters and time delays are estimated simultaneously by using the presented algorithm.

The remaining of this article is organized as follows. Section 2 introduces the multi-input system model with unknown input delays and colored noise. In Section 3, a greedy pursuit hierarchical iterative algorithm is given based on the over-parameterized sparse system model. In Section 4, some simulation experiments are given to show the effectiveness of the proposed method. Finally, Section 5 gives a summary of the content.

### 2. Systems Model

Consider the following systems shown in Figure 1.

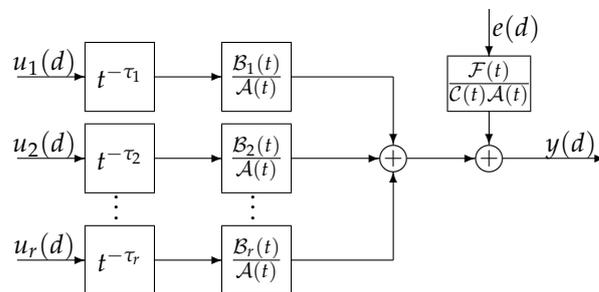


Figure 1. The multi-input systems with input time delays and colored noise.

The systems can be depicted as

$$\mathcal{A}(t)y(d) = \sum_{i=1}^r t^{-\tau_i} \mathcal{B}_i(t)u_i(d) + \frac{\mathcal{F}(t)}{\mathcal{C}(t)}e(d), \tag{1}$$

where \$u\_i(d)\$ and \$y(d)\$ are the available input and output sampled data, and \$e(d)\$ is the noise satisfying \$e(d) \sim N(0, \sigma^2)\$, respectively. \$\mathcal{A}(t), \mathcal{B}\_i(t), \mathcal{C}(t), \mathcal{F}(t)\$ are

$$\mathcal{A}(t) := 1 + \sum_{j=1}^{n_a} a_j t^{-j}, \mathcal{B}_i(t) := \sum_{j=1}^{n_{bi}} b_{ij} t^{-j}, \mathcal{C}(t) := 1 + \sum_{j=1}^{n_c} c_j t^{-j}, \mathcal{F}(t) := 1 + \sum_{j=1}^{n_f} f_j t^{-j}.$$

where \$n\_a, n\_{bi}, n\_c\$, and \$n\_f\$ are the system known orders and where the input time delay \$\tau\_i\$ of each channel is unknown.

Introduce the internal noise item \$w(d)\$

$$w(d) := \frac{\mathcal{F}(t)}{\mathcal{C}(t)}e(d) = - \sum_{j=1}^{n_c} c_j w(d-j) + \sum_{j=1}^{n_f} f_j e(d-j) + e(d)$$

$$= \boldsymbol{\varphi}_n^T(d)\boldsymbol{\theta}_n + e(d), \tag{2}$$

where

$$\boldsymbol{\varphi}_n(d) := [-w(d-1), -w(d-2), \dots, -w(d-n_c), e(d-1), e(d-2), \dots, e(d-n_f)]^T \in \mathbb{R}^{n_c+n_f}, \tag{3}$$

$$\boldsymbol{\theta}_n := [c_1, c_2, \dots, c_{n_c}, f_1, f_2, \dots, f_{n_f}]^T \in \mathbb{R}^{n_c+n_f}. \tag{4}$$

The subscript  $n$  denotes noise.

Then, system model (1) can be rewritten as

$$y(d) = [1 - A(t)]y(d) + \sum_{i=1}^r t^{-\tau_i} \mathcal{B}_i(t)u_i(d) + w(d). \tag{5}$$

Since the time delay of each input channel is unknown, define the data regression length  $l$ , which is large enough to satisfy  $l \geq \max(\tau_i + n_{bi})$ . Define the over-parameterized vector  $\boldsymbol{\varphi}_s(t)$  and  $\boldsymbol{\theta}_s$  as

$$\boldsymbol{\varphi}_s(d) := [\boldsymbol{\varphi}_a^T(d), \boldsymbol{\varphi}_b^T(d)]^T \in \mathbb{R}^{n_a+lr}, \tag{6}$$

$$\boldsymbol{\theta}_s := [\boldsymbol{\theta}_a^T, \boldsymbol{\theta}_b^T]^T \in \mathbb{R}^{n_a+lr}, \tag{7}$$

where

$$\boldsymbol{\varphi}_a(d) := [-y(d-1), -y(d-2), \dots, -y(d-n_a)]^T \in \mathbb{R}^{n_a},$$

$$\boldsymbol{\varphi}_b(d) := [\boldsymbol{\varphi}_{b1}^T(x), \boldsymbol{\varphi}_{b2}^T(x), \dots, \boldsymbol{\varphi}_{br}^T(x)]^T \in \mathbb{R}^{lr},$$

$$\boldsymbol{\varphi}_{bi}(d) := [u_i(d-1), \dots, u_i(d-\tau_i), u_i(d-\tau_i-1), \dots, u_i(d-\tau_i-n_{bi}), \dots, u_i(d-l)]^T \in \mathbb{R}^l.$$

$$\boldsymbol{\theta}_a := [a_1, a_2, \dots, a_{n_a}]^T \in \mathbb{R}^{n_a},$$

$$\boldsymbol{\theta}_b := [\boldsymbol{\theta}_{b1}^T, \boldsymbol{\theta}_{b2}^T, \dots, \boldsymbol{\theta}_{br}^T]^T \in \mathbb{R}^{lr}, \boldsymbol{\theta}_{bi} := [\mathbf{0}_{\tau_i}, b_{i1}, b_{i2}, \dots, b_{in_{bi}}, \mathbf{0}_{l-\tau_i-n_{bi}}]^T \in \mathbb{R}^l, i = 1, 2, \dots, r.$$

The subscript  $s$  denotes the system, and  $\mathbf{0}_x$  denotes the zero block where the subscript means the number of zero elements.

Then, the following over-parameterized model can be indicated from Equation (5)

$$y(d) = \boldsymbol{\varphi}_s^T(d)\boldsymbol{\theta}_s + w(d) = \boldsymbol{\varphi}_s^T(d)\boldsymbol{\theta}_s + \boldsymbol{\varphi}_n^T(d)\boldsymbol{\theta}_n + e(d) = \boldsymbol{\varphi}^T(d)\boldsymbol{\theta} + e(d), \tag{8}$$

where

$$\boldsymbol{\varphi}(d) := [\boldsymbol{\varphi}_s^T(d), \boldsymbol{\varphi}_n^T(d)]^T \in \mathbb{R}^N, N := n_a + lr + n_c + n_f,$$

$$\boldsymbol{\theta} := [\boldsymbol{\theta}_s^T, \boldsymbol{\theta}_n^T]^T \in \mathbb{R}^N.$$

Consider the sampled data from  $d = 1$  to  $d = L$  and define  $\mathbf{Y}, \boldsymbol{\Phi}, \mathbf{E}$  as

$$\mathbf{Y} := [y(1), y(2), \dots, y(L)]^T \in \mathbb{R}^L,$$

$$\boldsymbol{\Phi} := [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \dots, \boldsymbol{\varphi}(L)]^T \in \mathbb{R}^{L \times N},$$

$$\mathbf{E} := [e(1), e(2), \dots, e(L)]^T \in \mathbb{R}^L.$$

The over-parameterized system model can be wrote as

$$\mathbf{Y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{E}. \tag{9}$$

Minimize the following cost function

$$J(\boldsymbol{\theta}) := \sum_{j=1}^L [y(j) - \boldsymbol{\varphi}^T(j)\boldsymbol{\theta}]^2 = \mathbf{E}^T \mathbf{E},$$

and the LS estimation value of the parameter vector will be obtained by using the LS principle

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{Y}. \tag{10}$$

**Remark 1.** It is worth noting that Equation (10) cannot estimate the parameter directly as  $\varphi_n(d)$  contains unmeasurable internal noise items  $w(d)$  and  $e(d)$ . In addition, the over-parameterized sparse system model has a large dimension  $N$ . Directly adopting the least squares method requires the sampled data to be much larger than the dimension  $N$  of the system model to achieve the satisfactory estimation accuracy. In some extreme cases, it will lead to matrix  $S = [\hat{\Phi}^T \hat{\Phi}]^{-1}$  singularity when sufficient sampled data cannot be obtained. A large number of samples and high-dimensional matrix operations will increase the estimation cost.

In order to achieve high efficiency of parameters and time-delay estimation, this paper drives a greedy pursuit hierarchical iterative algorithm for system model (1) using limited sampled data according to the sparse property of the over-parameterized system model.

### 3. Greedy Pursuit Hierarchical Iterative Parameter Estimation Algorithm

The sparsity property of the over-parameterized system model makes it possible to identify the multi-input system with limited sampled data according to the compressed sensing idea. System model (9) can be rewritten as

$$Y = \sum_{j=1}^N \phi_j \theta_j + E, \tag{11}$$

where  $\phi_j$  and  $\theta_j$  are the  $j$ -th column of  $\Phi$  and the corresponding parameter in  $\theta$ . According to Equation (7),  $\theta$  is sparse with a large number of zero elements and only contains a small number of non-zero key parameters. When the locations of the key parameters are found, the goal of reducing the dimensionality of the over-parameterized system model can be achieved. This provides the possibility of using limited sampling data to realize the parameters and time-delay estimation.

In order to meet the sparse recovery conditions, the information matrix  $\Phi$  should be normalized. Define the normalized transformation matrix  $T$  as

$$T := \begin{bmatrix} \Phi(1) & 0 & \cdots & 0 \\ 0 & \Phi(2) & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \Phi(N) \end{bmatrix}^{-1} \in \mathbb{R}^{N \times N} \tag{12}$$

where  $\Phi(j) = \sqrt{\sum_{i=1}^L \Phi_{ij}^2}$  denotes the modulus of the  $j$ th column of  $\Phi$ . Introducing  $T$  to Equation (9), the following can be obtained:

$$Y = \Psi \vartheta + E, \tag{13}$$

where  $\Psi = \Phi T$ ,  $\vartheta = T^{-1} \theta$ .

Let  $q$  be the inner iteration variable. Define the residual output as

$$r_q := Y - \Psi_{\Lambda_q} \vartheta_{\Lambda_q}. \tag{14}$$

where  $\Lambda_q$  is an index set that is composed of the locations of the found key nonzero parameters,  $\Psi_{\Lambda_q}$  is a sub-matrix of matrix  $\Psi$ , and  $\vartheta_{\Lambda_q}$  is a non-zero parameter vector. Both  $\Psi_{\Lambda_q}$  and  $\vartheta_{\Lambda_q}$  are indexed by  $\Lambda_q$ . The sub-matrix  $\Psi_{\Lambda_q}$  is composed of the indexed columns of the information matrix  $\Psi$ .

At the next internal iteration, define the following cost function and minimize it to find location of one key parameter.

$$J(j) := \|r_q - \psi_j \vartheta_j\|_2, \tag{15}$$

which lead to  $\vartheta_j = \boldsymbol{\psi}_j \mathbf{r}_q / \|\boldsymbol{\psi}_j\|_2^2$ . Plugged back  $\vartheta_j$  into Equation (15), then we have

$$\begin{aligned} J(j) &= \min_{\vartheta_j} \|\boldsymbol{\psi}_j \vartheta_j - \mathbf{r}_q\|_2^2 = \left\| \frac{\boldsymbol{\psi}_j^T \mathbf{r}_q}{\|\boldsymbol{\psi}_j\|_2^2} \boldsymbol{\psi}_j - \mathbf{r}_q \right\|_2^2 \\ &= \|\mathbf{r}_q\|_2^2 - \frac{(\boldsymbol{\psi}_j^T \mathbf{r}_q)^2}{\|\boldsymbol{\psi}_j\|_2^2} = \|\mathbf{r}_q\|_2^2 - (\boldsymbol{\psi}_j^T \mathbf{r}_q)^2. \end{aligned} \tag{16}$$

where  $\boldsymbol{\psi}_j (j = 1, 2, \dots, N)$  is the column of the normalized matrix  $\boldsymbol{\Psi}$ . From Equation (16), the column most relevant to the last iteration residual  $\mathbf{r}_q$  can be found at next iteration. That is, the minimum problem is equivalent to the largest inner product (in absolute value) between the columns of  $\boldsymbol{\Psi}$  and  $\mathbf{r}_q$ . The location of found column can be indicated by the index

$$\lambda_{q+1} = \arg \max_{j=1,2,\dots,N} |\langle \mathbf{r}_q, \boldsymbol{\psi}_j \rangle|. \tag{17}$$

Then, the index set and the index matrix can be updated according to  $\Lambda_{q+1} := \Lambda_q \cup \lambda_{q+1}$ .

**Remark 2.** Because  $w(d)$  and  $e(d)$  in the matrix  $\boldsymbol{\Phi}$  are unknown noise items. The above steps from Equations (12)–(17) cannot be directly implemented since  $\boldsymbol{\Phi}$  contains some unknown items according to Equation (2).

Replace the true unmeasurable values  $w(d)$  and  $e(d)$  with their last iteration estimate by using the hierarchical principle [29]. Let  $k$  be the external iteration variable.  $\hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\theta}}_{k,s}, \hat{\boldsymbol{\theta}}_{k,n}$  are iteration estimates of  $\boldsymbol{\theta}, \boldsymbol{\theta}_s, \boldsymbol{\theta}_n$ , respectively. By replacing  $d$  with  $d - j$  in Equations (2) and (8), the following results can be obtained:

$$w(d - j) = \boldsymbol{\varphi}_n^T(d - j) \boldsymbol{\theta}_n + e(d - j), \tag{18}$$

$$y(d - j) = \boldsymbol{\varphi}_s^T(d - j) \boldsymbol{\theta}_s + w(d - j). \tag{19}$$

Then, the unknown noise items  $w(d - j), e(d - j)$  can be replaced by their last estimate  $\hat{w}_{k-1}(d - j), \hat{e}_{k-1}(d - j)$  based on Equation (3) and the hierarchical principle,

$$\hat{\boldsymbol{\varphi}}_{k,n}(d) = [-\hat{w}_{k-1}(d - 1), -\hat{w}_{k-1}(d - 2), \dots, -\hat{w}_{k-1}(d - n_c), \hat{e}_{k-1}(d - 1), \hat{e}_{k-1}(d - 2), \dots, \hat{e}_{k-1}(d - n_f)]^T \in \mathbb{R}^{n_e + n_f}. \tag{20}$$

Replace  $\boldsymbol{\varphi}_n^T(d - j), \boldsymbol{\theta}_n, \boldsymbol{\theta}_s$  with their iteration estimates  $\hat{\boldsymbol{\varphi}}_{k,n}^T(d - j), \hat{\boldsymbol{\theta}}_{k,n}, \hat{\boldsymbol{\theta}}_{k,s}$  in Equations (18) and (19); the estimation values of  $w(d - j)$  and  $e(d - j)$  at  $k$ -th iteration can be achieved by

$$\hat{w}_k(d - j) = y(d - j) - \boldsymbol{\varphi}_{k,s}^T(d - j) \hat{\boldsymbol{\theta}}_{k,s}, \tag{21}$$

$$\hat{e}_k(d - j) = \hat{w}_k(d - j) - \hat{\boldsymbol{\varphi}}_{k,n}^T(d - j) \hat{\boldsymbol{\theta}}_{k,n}. \tag{22}$$

According to Equation (20), the estimation of information matrix  $\hat{\boldsymbol{\Phi}}_k$  can be denoted at the  $k$ -th external iteration, where  $\hat{\boldsymbol{\varphi}}_k(d) = [\boldsymbol{\varphi}_s^T(d), \hat{\boldsymbol{\varphi}}_{k,n}^T(d)]^T$ .

Use  $\hat{\boldsymbol{\Phi}}_k$  to replace  $\boldsymbol{\Phi}$  in Equation (9). Construct the normalized transformation matrix  $\hat{T}$  using  $\hat{\boldsymbol{\Phi}}_k$  based on Equation (12), and construct  $\hat{\boldsymbol{\Psi}}_k$  according to  $\hat{\boldsymbol{\Phi}}_k$  and  $\hat{T}$ .

Then, Equations (14) and (15) can be expressed as

$$\mathbf{r}_{k,q} = \mathbf{Y} - \hat{\boldsymbol{\Psi}}_{k,\Lambda_q} \hat{\boldsymbol{\theta}}_{k,\Lambda_q}, \tag{23}$$

$$\lambda_{k,q} = \arg \max_{j=1,2,\dots,N} |\langle \mathbf{r}_{k,q-1}, \hat{\boldsymbol{\psi}}_j \rangle|, \tag{24}$$

where the double subscripts  $k, q$  denote that the key index at  $q$  inner iteration can be found using the  $k$ th external iteration updated data.

Applying the index set to update the normalization sub-matrix, we have

$$\hat{\Psi}_{k,\Lambda_q} = \hat{\Psi}_{k,\Lambda_{q-1}} \cup \hat{\psi}_{k,\lambda_q}, \quad \Lambda_q = \Lambda_{q-1} \cup \lambda_q. \tag{25}$$

Define a cost function as follows at the  $q$ th inner iteration to find the best parameter estimation value

$$J(\boldsymbol{\theta}_{k,\Lambda_q}) = \|\mathbf{r}_{k,q}\|^2 = \|\mathbf{Y} - \hat{\Psi}_{k,\Lambda_q} \boldsymbol{\theta}_{k,\Lambda_q}\|^2. \tag{26}$$

Minimize Equation (26), and the LS estimate can be obtained

$$\hat{\boldsymbol{\theta}}_{k,\Lambda_q} = [\hat{\Psi}_{k,\Lambda_q}^T \hat{\Psi}_{k,\Lambda_q}]^{-1} \hat{\Psi}_{k,\Lambda_q}^T \mathbf{Y}. \tag{27}$$

Note that  $\hat{\boldsymbol{\theta}}_{k,\Lambda_q} \in \mathbb{R}^q$ , using  $\Lambda_{k,q}$  and the transformation matrix  $\hat{\mathbf{T}}$ , the over-parameterized sparse parameter vector can be obtained.

Then, the steps of the greedy pursuit hierarchical iterative (GPHI) algorithm are listed as follows.

1. Define  $l$  and collect sampled data  $\{u_i(d), y(d): d = 1, 2, \dots, L\}$  to form  $\mathbf{Y}$ .
2. To initialize external iteration: let  $k = 1$ ,  $\hat{\boldsymbol{\theta}} = \mathbf{1}_N/p_0$ ,  $p_0 = 10^6$ ,  $\hat{w}_0(d), \hat{e}_0(d)$  be a random number, and give allowable error  $\epsilon$  and  $\epsilon$ .
3. Form  $\hat{\boldsymbol{\phi}}_{k,n}(d)$  by Equation (20), and  $\hat{\boldsymbol{\phi}}_k(d)$  by

$$\hat{\boldsymbol{\phi}}_k(d) = [\boldsymbol{\varphi}_s^T(d), \hat{\boldsymbol{\phi}}_{k,n}^T(d)]^T. \tag{28}$$

Using Equation (29), update  $\hat{\boldsymbol{\Phi}}_k$ , and using Equation (12) based on  $\hat{\boldsymbol{\Phi}}_k$ , construct  $\hat{\mathbf{T}}$ . Then, construct  $\hat{\Psi}_k$  by using  $\hat{\boldsymbol{\Phi}}_k$  and  $\hat{\mathbf{T}}$ .

4. Begin the internal iteration. Let  $q = 1$ ,  $\mathbf{r}_{k,0} = \mathbf{Y}$ , and  $\Lambda_0 = \emptyset$ ,  $\hat{\Psi}_{k,\Lambda_0} = \emptyset$ .
  - (a) Find  $\lambda_q$  by Equation (24).
  - (b) Update  $\Lambda_q$  and  $\hat{\Psi}_{k,\Lambda_q}$  by Equation (25).
  - (c) Compute the parameter estimate  $\hat{\boldsymbol{\theta}}_{k,\Lambda_q}$  by Equation (27). Update residual  $\mathbf{r}_q$  by Equation (23).
  - (d) If  $\|\hat{\boldsymbol{\theta}}_{k,\Lambda_q} - \hat{\boldsymbol{\theta}}_{k,\Lambda_{q-1}}\| \leq \epsilon$ , obtain  $\hat{\boldsymbol{\theta}}_{k,\Lambda_q}$ ; otherwise, increase  $q$  by 1 and go to step (a).
5. Recover parameter estimate  $\hat{\boldsymbol{\theta}}_k$  by

$$\hat{\boldsymbol{\theta}}_{k,\Lambda_q} \in \mathbb{R}^p \xrightarrow{\Lambda_q} \hat{\boldsymbol{\theta}}_k \in \mathbb{R}^N, \hat{\boldsymbol{\theta}}_k = \hat{\mathbf{T}} \hat{\boldsymbol{\theta}}_k. \tag{29}$$

Update noise estimates  $\hat{w}_k(d-j), \hat{e}_k(d-j)$  by Equations (21) and (22), and update the noise estimation vectors  $\hat{\boldsymbol{\phi}}_{k,n}(d), \hat{\boldsymbol{\phi}}_k(d)$  using Equations (20) and (29),  $\hat{\boldsymbol{\Phi}}_k$  and  $\hat{\Psi}_k$  by

$$\hat{\boldsymbol{\Phi}}_k = [\hat{\boldsymbol{\phi}}_k(1), \hat{\boldsymbol{\phi}}_k(2), \dots, \hat{\boldsymbol{\phi}}_k(L)]^T, \hat{\Psi}_k = \hat{\boldsymbol{\Phi}}_k \hat{\mathbf{T}}. \tag{30}$$

6. If  $\|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k-1}\| \leq \epsilon$ , complete the iteration stage and receive the final estimate  $\hat{\boldsymbol{\theta}}$ ; otherwise, let  $k = k + 1$  and turn to step 4.

The recovered parameter vector  $\hat{\boldsymbol{\theta}}$  has  $r + 1$  zero blocks. With prior knowledge  $l$  and orders  $n_a, n_{b_i}, n_c, n_f$ , time-delay estimates will be received by

$$\begin{aligned} \hat{\tau}_1 &= n_1, \\ \hat{\tau}_i &= n_j - (l - \hat{\tau}_{i-1} - n_{b_j}), \end{aligned} \tag{31}$$

where  $n_j(j = 2, 3, \dots, r)$  is the number of zero items in each zero block.

### 4. Simulation Experiments

**Experiment 1.** Consider a multi-input system as follows:

$$\begin{aligned}
 \mathcal{A}(t)y(d) &= \sum_{i=1}^5 t^{\tau_i} \mathcal{B}_i(t)u_i(d) + \frac{\mathcal{F}(t)}{\mathcal{C}(t)}e(d), \\
 \mathcal{A}(t) &= 1 + a_1t^{-1} + a_2t^{-2} = 1 - 0.80t^{-1} + 0.60t^{-2}, \\
 \mathcal{B}_1(t) &= b_{11}t^{-1} + b_{12}t^{-2} = 2.00t^{-1} - 1.20t^{-2}, \\
 \mathcal{B}_2(t) &= b_{21}t^{-1} + b_{22}t^{-2} = -1.80t^{-1} - 0.90t^{-2}, \\
 \mathcal{B}_3(t) &= b_{31}t^{-1} + b_{32}t^{-2} = 1.00t^{-1} + 0.50t^{-2}, \\
 \mathcal{C}(t) &= 1 + c_1t^{-1} = 1 + 0.80t^{-1}, \\
 \mathcal{F}(t) &= 1 + f_1t^{-1} = 1 - 0.40t^{-1}. \\
 \tau_1 &= 6, \tau_2 = 15, \tau_3 = 20.
 \end{aligned}$$

Let  $l = 30$ ; the true value of the parameter vector to be estimated is

$$\theta = [-0.80, 0.60, \mathbf{0}_6, 2.00, -1.20, \mathbf{0}_{37}, -1.80, -0.90, \mathbf{0}_{33}, 1.00, 0.50, \mathbf{0}_8, 0.80, -0.40]^T \in \mathbb{R}^N.$$

The  $\{u_i(d)\}$  and  $\{y(d)\}$  are measurable. The  $e(d) \sim \mathcal{N}(0, \sigma^2)$  is noise, and the noise variances are taken to be  $\sigma^2 = 0.50^2$  and  $\sigma^2 = 1.00^2$ , respectively. Take the sampled data length  $L = 1000$ . The dimension of  $\theta$  is  $N := n_a + lr + n_c + n_f = 94$ , where the sparsity is

$$K := n_a + \sum_{i=1}^5 n_{bi} + n_c + n_f = 10.$$

Use the first 200 sampled data to estimate this experiment by applying the GPHI algorithm, and the data from  $d = 200$  to  $d = 500$  are taken to verify the effectiveness of the obtained results. Define parameter estimation error as  $\delta := \|\hat{\theta}_k - \theta\| / \theta \times 100\%$ .

Noise variance is a commonly utilized metric for quantifying noise in measurement data, with higher values indicating reduced accuracy and reliability of the measurements. The impact of noise variance on parameter estimation accuracy cannot be overlooked. To evaluate the robustness of the GPHI algorithm against noise interference, the different noise variances  $\sigma^2 = 0.50^2$  and  $\sigma^2 = 1.00^2$  are applied.

The parameter estimates and estimation error  $\delta$  versus iterative times  $k$  are shown in Table 1 and Figure 2. From Figure 2, it can be seen that the estimation errors tend to decrease as the iterative times increase under different noise variances. The parameter estimates versus iterative times  $k$  are shown in Figures 3–6.

**Table 1.** The estimated parameter values and estimation errors of Experiment 1 with  $L = 200$  ( $\sigma^2 = 0.50^2, \sigma^2 = 1.00^2$ ).

$\sigma^2$	$k$	$a_1$	$a_2$	$b_{11}$	$b_{12}$	$b_{21}$	$b_{22}$	$b_{31}$	$b_{32}$	$c_1$	$f_1$	$\delta\%$
0.50 <sup>2</sup>	1	-0.7107	0.5090	1.9694	-1.0090	-1.8753	-1.0219	1.0790	0.5366	0.0000	0.0000	27.7831
	2	-0.8717	0.6218	1.9681	-1.3473	-1.8160	-0.7010	1.0268	0.3843	0.9248	0.0000	15.8395
	3	-0.7982	0.5895	1.9816	-1.1991	-1.7996	-0.8780	1.0396	0.4889	0.8053	0.0000	12.1355
	5	-0.8052	0.5905	1.9772	-1.2174	-1.7890	-0.8756	1.0339	0.4876	0.7704	-0.2763	3.9074
	8	-0.7905	0.5818	1.9615	-1.1770	-1.7950	-0.9001	1.0299	0.4972	0.7374	-0.4265	2.5213
1.00 <sup>2</sup>	1	-0.5319	0.3512	1.9279	-0.6311	-1.9520	-1.3414	1.1347	0.6812	0.0000	0.0000	40.4417
	2	-1.1028	0.6513	1.9611	-1.7779	-1.8306	-0.1742	1.0516	0.0000	1.1758	0.0000	42.0247
	3	-0.7517	0.6354	1.9523	-1.1405	-1.8269	-0.9556	1.1100	0.5334	0.6873	0.0000	16.0285
	5	-0.8796	0.6296	1.9260	-1.3454	-1.8156	-0.7463	1.0452	0.4368	0.9477	0.0000	17.9049
	8	-0.8218	0.5847	1.9238	-1.2333	-1.7917	-0.8439	1.0565	0.4663	0.7521	-0.4519	4.0108
True values		-0.8000	0.6000	2.0000	-1.2000	-1.8000	-0.9000	1.0000	0.5000	0.8000	-0.4000	

From Figures 3–6, it can be concluded that the estimated values gradually approach the true values with the increase in iterative times under different noise variances.

When noise variances are  $\sigma^2 = 0.50^2$  and  $\sigma^2 = 1.00^2$ , the true outputs and estimated outputs versus iterative times  $k$  are given in Figures 7 and 8, and a comparison of  $\delta$  and the different sampled data length is shown in Table 2. When the system is disturbed by different noises, the estimated outputs can better match the true outputs and the bias of them is small—see Figures 7 and 8.

With  $\sigma^2 = 0.50^2$  and  $L = 200$ , the locations of the estimated key non-zero parameters are shown in Table 3.

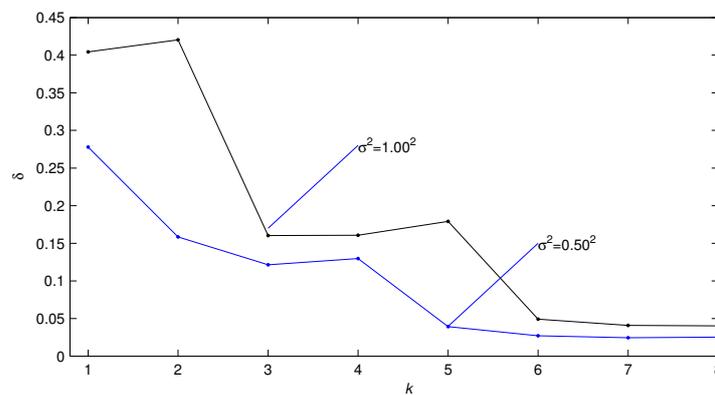


Figure 2. The estimation errors  $\delta$  versus  $k$  of Experiment 1 with  $L = 200$  ( $\sigma^2 = 0.50^2, \sigma^2 = 1.00^2$ ).

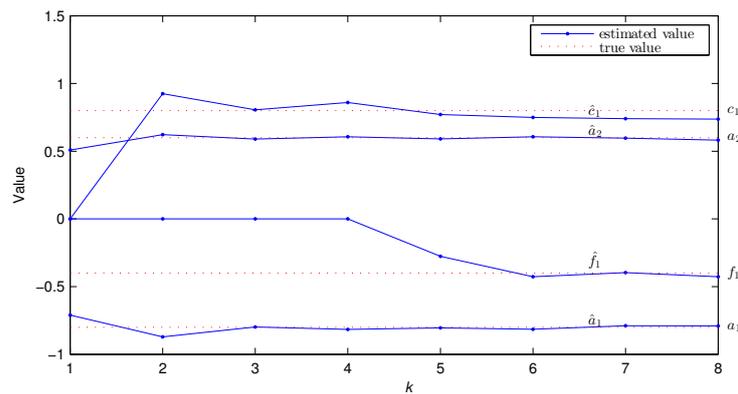


Figure 3. The estimated parameter values  $\hat{a}_1, \hat{a}_2, \hat{c}_1, \hat{f}_1$  versus  $k$  of Experiment 1 with  $L = 200$  and  $\sigma^2 = 0.50^2$ .

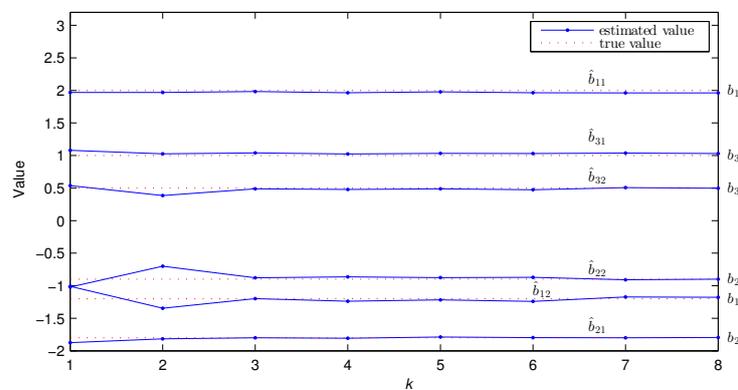
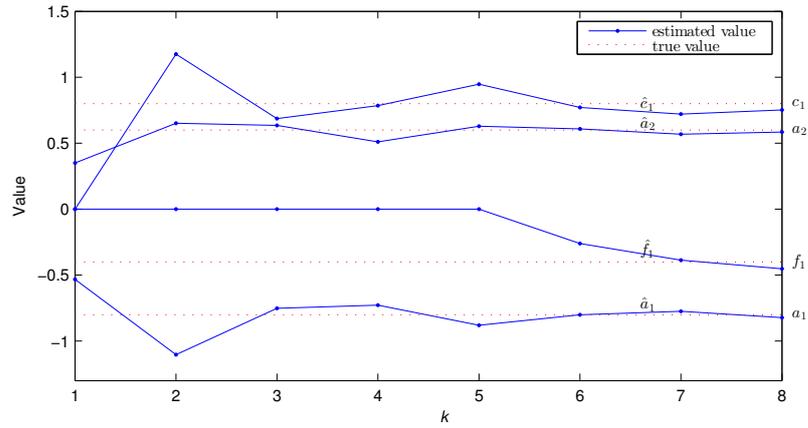
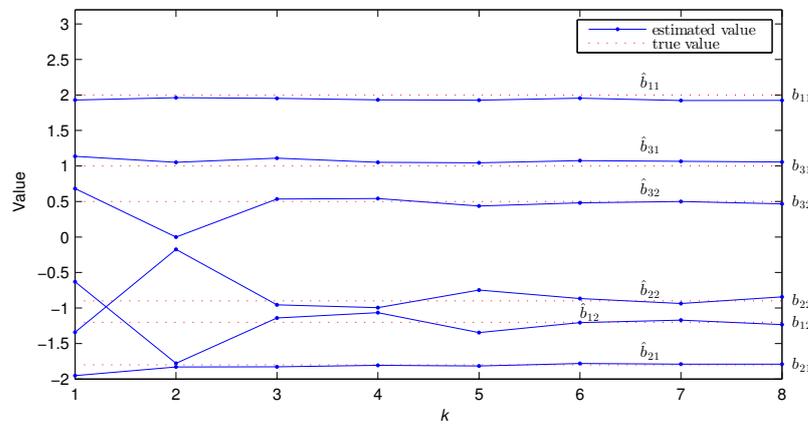


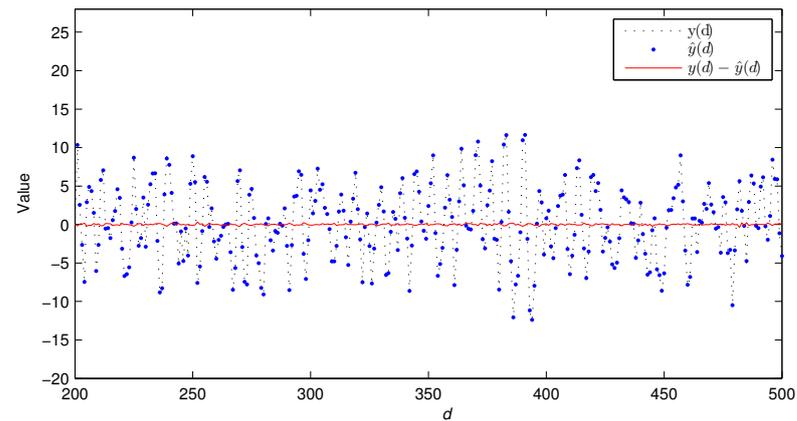
Figure 4. The estimated parameter values  $\hat{b}_{11}, \hat{b}_{12}, \hat{b}_{21}, \hat{b}_{22}, \hat{b}_{31}, \hat{b}_{32}$  versus  $k$  of Experiment 1 with  $L = 200$  and  $\sigma^2 = 0.50^2$ .



**Figure 5.** The estimated parameter values  $\hat{a}_1, \hat{a}_2, \hat{c}_1, \hat{f}_1$  versus  $k$  of Experiment 1 with  $L = 200$  and  $\sigma^2 = 1.00^2$ .



**Figure 6.** The estimated parameter values  $\hat{b}_{11}, \hat{b}_{12}, \hat{b}_{21}, \hat{b}_{22}, \hat{b}_{31}, \hat{b}_{32}$  versus  $k$  of Experiment 1 with  $L = 200$  and  $\sigma^2 = 1.00^2$ .



**Figure 7.** The true outputs, estimated outputs, and their bias of Experiment 1 with  $\sigma^2 = 0.50^2$ .

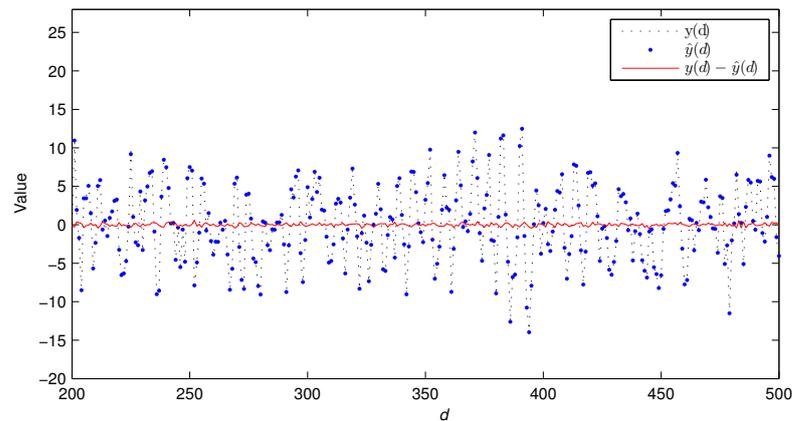


Figure 8. The true outputs, estimated outputs, and their bias of Experiment 1 with  $\sigma^2 = 1.00^2$ .

Table 2. The estimation errors with different  $L$  and  $\sigma^2$ .

Sampled Data Length $L$	400	500	600	700	800	1000
Estimation error $\delta(\%)(\sigma^2 = 0.50^2)$	2.3931	2.591	2.0535	1.6165	1.9868	2.2789
Estimation error $\delta(\%)(\sigma^2 = 1.00^2)$	6.1781	5.9393	4.1059	4.3916	3.6913	4.5132

Table 3. The locations of the estimated key non-zero parameters of Experiment 1 with  $L = 200$  and  $\sigma^2 = 0.50^2$ .

Parameter	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_{11}$	$\hat{b}_{12}$	$\hat{b}_{21}$	$\hat{b}_{22}$	$\hat{b}_{31}$	$\hat{b}_{32}$	$\hat{c}_1$	$\hat{f}_1$
Location	1	2	9	10	48	49	83	84	93	94

According to Table 3 and Equation (31), the time delays can be estimated

$$\begin{aligned}
 \hat{\tau}_1 &= n_1 = 6, \\
 \hat{\tau}_2 &= n_2 - (l - \hat{\tau}_1 - n_{b2}) = 37 - (30 - 6 - 2) = 15, \\
 \hat{\tau}_3 &= n_3 - (l - \hat{\tau}_2 - n_{b3}) = 33 - (30 - 15 - 2) = 20.
 \end{aligned}
 \tag{32}$$

Experiment 2. Consider the other multi-input system,

$$\begin{aligned}
 A(t)y(d) &= \sum_{i=1}^5 t^{\tau_i} B_i(t)u_i(d) + \frac{F(t)}{C(t)}e(d), \\
 A(t) &= 1 + a_1t^{-1} + a_2t^{-2} = 1 + 0.60t^{-1} + 0.4t^{-2}, \\
 B_1(t) &= b_{11}t^{-1} + b_{12}t^{-2} = 2.0t^{-1} - 1.3t^{-2}, \\
 B_2(t) &= b_{21}t^{-1} + b_{22}t^{-2} = 1.5t^{-1} - 0.9t^{-2}, \\
 B_3(t) &= b_{31}t^{-1} + b_{32}t^{-2} = -1.0t^{-1} + 0.5th^{-2}, \\
 B_4(t) &= b_{41}t^{-1} + b_{42}t^{-2} = -1.2t^{-1} + 0.6t^{-2}, \\
 B_5(t) &= b_{51}t^{-1} + b_{52}t^{-2} = 1.0t^{-1} - 0.8t^{-2}, \\
 C(t) &= 1 + c_1t^{-1} = 1 + 0.7t^{-1}, \\
 F(t) &= 1 + f_1t^{-1} = 1 - 0.4t^{-1}. \\
 \tau_1 &= 9, \tau_2 = 23, \tau_3 = 15, \tau_4 = 30, \tau_5 = 17.
 \end{aligned}$$

Let the data regression length  $l = 50$ . The true parameter vector to be identified is

$$\theta = [0.60, 0.40, \mathbf{0}_9, 2.00, -1.30, \mathbf{0}_{62}, 1.50, -0.90, \mathbf{0}_{40}, -1.00, 0.50, \mathbf{0}_{63}, -1.20, 0.60, \mathbf{0}_{35}, 1.00, -0.80, \mathbf{0}_{31}, 0.70, -0.40]^T \in \mathbb{R}^N.$$

The dimension of sparse parameter vector is  $N := n_a + lr + n_c + n_f = 254$ , where the number of key non-zero parameters is  $K := n_a + \sum_{i=1}^5 n_{bi} + n_c + n_f = 14$ . In this simulation, take the length of measurable input and output data  $L = 1000$ . The data from  $d = 1$  to  $d = 250$  are used to realize the parameters and time-delay estimation. The remaining data from  $d = 250$  to  $d = 500$  are used to contrast the bias between the true outputs and estimated outputs. The other constraints are similar with those in Experiment 1.

To show the advantages of the GPHI algorithm, the errors  $\delta$  of the LSI and GPHI algorithms versus the iteration times  $k$  are depicted in Figure 9, where sampled data length  $L$  is 250 and noise variance is  $\sigma^2 = 0.50^2$ . From Figure 9, it can be found that the GPHI algorithm can achieve higher parameter estimation accuracy with limited sampled data.

When the noise variances are  $\sigma^2 = 0.50^2, \sigma^2 = 1.00^2$ , apply the GPHI algorithm. The errors  $\delta$  versus  $k$  are shown in Table 4 and Figure 10, and the parameter estimates versus  $k$  are given in Figures 11–14. Under different noise variances, the algorithm can achieve effective estimation accuracy of parameter estimation, and the parameter estimates approach the true values with the increase in the iterative times. This shows that the GPHI algorithm has certain robustness from Figures 10–14.

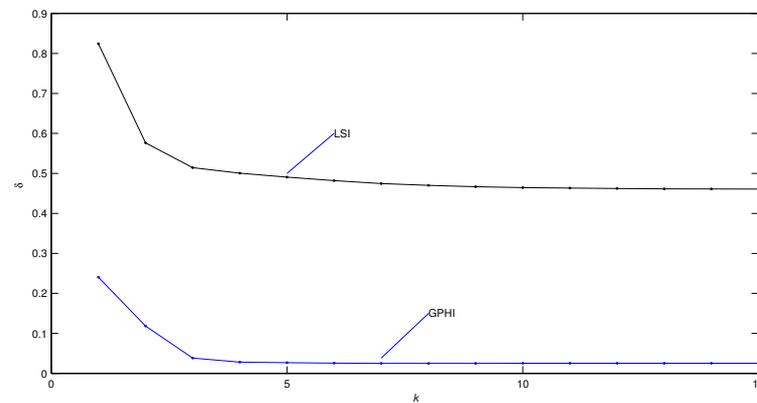


Figure 9. The estimation errors of the LSI and GPHI algorithms versus  $k$  of Experiment 2 ( $L = 250, \sigma^2 = 0.50^2$ ).

Table 4. The estimated parameter values of parameter and errors of Experiment 2 with  $L = 250$  ( $\sigma^2 = 0.50^2, \sigma^2 = 1.00^2$ ).

$\sigma^2$	$k$	$a_1$	$a_2$	$b_{11}$	$b_{12}$	$b_{21}$	$b_{22}$	$b_{31}$	$b_{32}$	$b_{41}$	$b_{42}$	$b_{51}$	$b_{52}$	$c_1$	$f_1$	$\delta\%$
0.50 <sup>2</sup>	1	0.7043	0.4193	1.9900	-1.1505	1.4756	-0.7222	-0.9542	0.3833	-1.2275	0.5137	0.9985	-0.6643	0.0000	0.0000	24.0379
	2	0.5958	0.3868	1.9909	-1.3471	1.4534	-0.8632	-0.9999	0.4732	-1.1926	0.5949	1.0018	-0.7892	0.8209	0.0000	11.8417
	3	0.5702	0.3731	1.9940	-1.4102	1.4709	-0.9009	-1.0007	0.5398	-1.1864	0.6208	0.9846	-0.8065	0.6720	-0.3408	3.8284
	5	0.5937	0.3974	1.9985	-1.3644	1.4680	-0.8740	-1.0119	0.5066	-1.1964	0.5860	0.9948	-0.7824	0.6427	-0.4213	2.6627
	8	0.5971	0.3944	1.9958	-1.3538	1.4675	-0.8662	-1.0131	0.5092	-1.1959	0.5802	0.9964	-0.7793	0.6478	-0.4059	2.4836
1.00 <sup>2</sup>	1	1.1487	0.5378	1.9923	-0.3923	1.5176	0.0000	-0.9743	0.0000	-1.1962	0.0000	0.9823	0.0000	0.0000	0.0000	68.7931
	2	0.6610	0.3569	1.9489	-1.2599	1.4150	-0.7717	-1.0095	0.3747	-1.2543	0.5356	0.9832	-0.7020	0.7977	0.0000	15.6424
	3	0.5342	0.3111	1.9775	-1.5040	1.4303	-0.9078	-0.9906	0.5777	-1.1896	0.6985	1.0027	-0.8310	0.7349	-0.2909	7.8178
	5	0.5616	0.3767	2.0078	-1.4832	1.4422	-0.9073	-1.0159	0.5403	-1.2081	0.6244	0.9810	-0.8144	0.6377	-0.4610	5.8361
	8	0.6041	0.3931	1.9989	-1.3938	1.4410	-0.8344	-1.0317	0.5114	-1.2076	0.5711	0.9894	-0.7675	0.6391	-0.4123	4.0277
True values		0.6000	0.4000	2.0000	-1.3000	1.5000	-0.9000	-1.0000	0.5000	-1.2000	0.6000	1.0000	-0.8000	0.7000	-0.4000	

Apply the data from  $d = 250$  to  $d = 500$  to verify the outputs with different noise variances. The true outputs, estimated outputs, and their bias are shown in Figures 15 and 16, respectively. Figures 15 and 16 show that the estimated output values can match the true values of the system well with different noise variances.

The errors  $\delta$  versus different sampled data length  $L$  are shown in Table 5 to verify the validity of estimation accuracy under limited sampled data length. With  $\sigma^2 = 0.50^2$  and  $L = 250$ , the key non-zero parameter location estimations are given in Table 6.

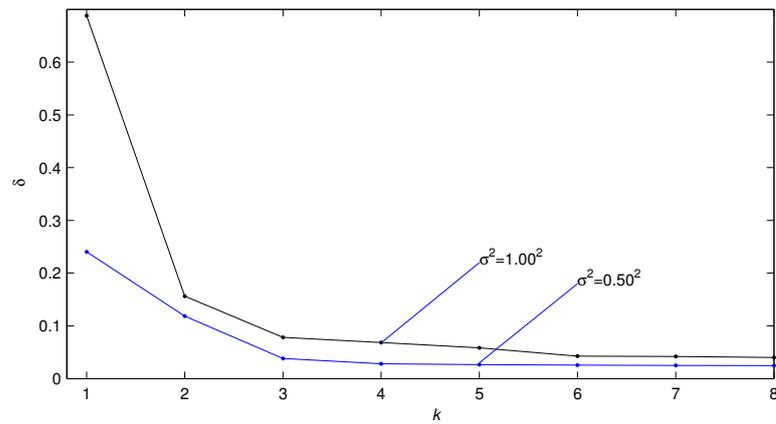


Figure 10. The estimation errors  $\delta$  versus  $k$  of Experiment 2 with  $L = 250$  and  $(\sigma^2 = 0.50^2, \sigma^2 = 1.00^2)$ .

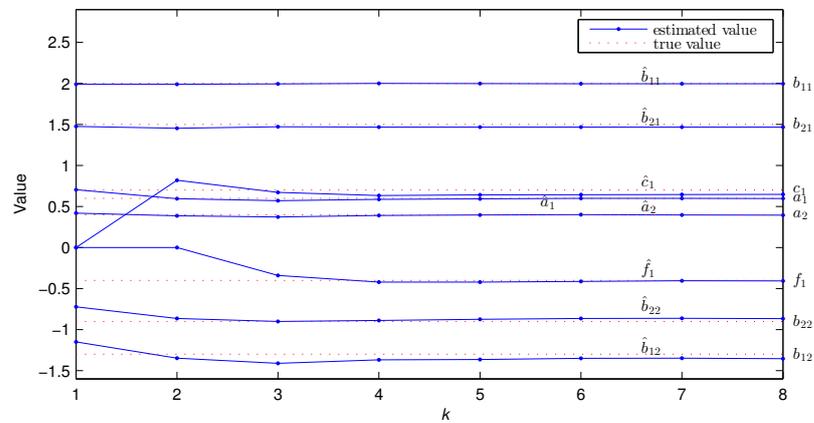


Figure 11. The estimated parameter values  $\hat{a}_1, \hat{a}_2, \hat{b}_{11}, \hat{b}_{12}, \hat{b}_{21}, \hat{b}_{22}, \hat{c}_1, \hat{f}_1$  versus  $k$  of Experiment 2 with  $L = 250$  and  $\sigma^2 = 0.50^2$ .

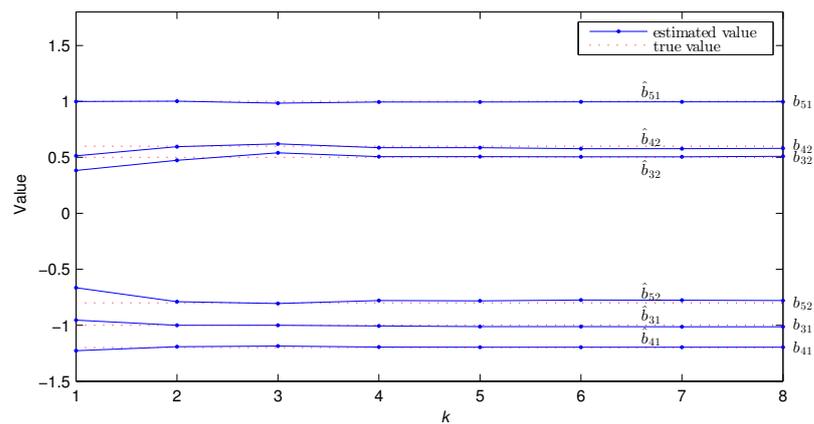
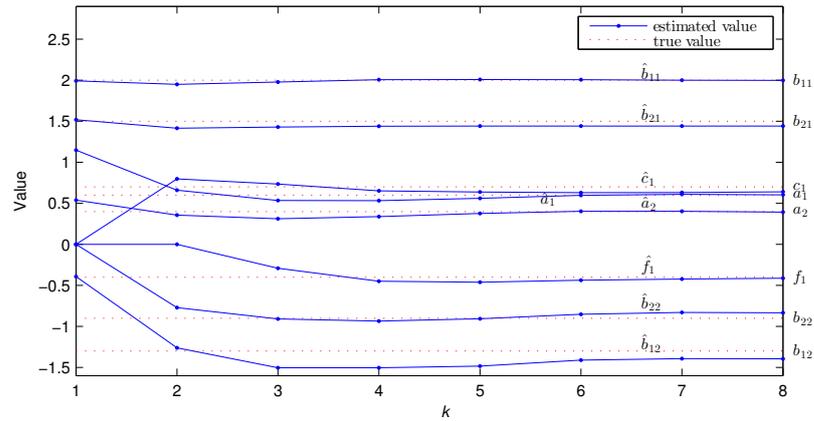
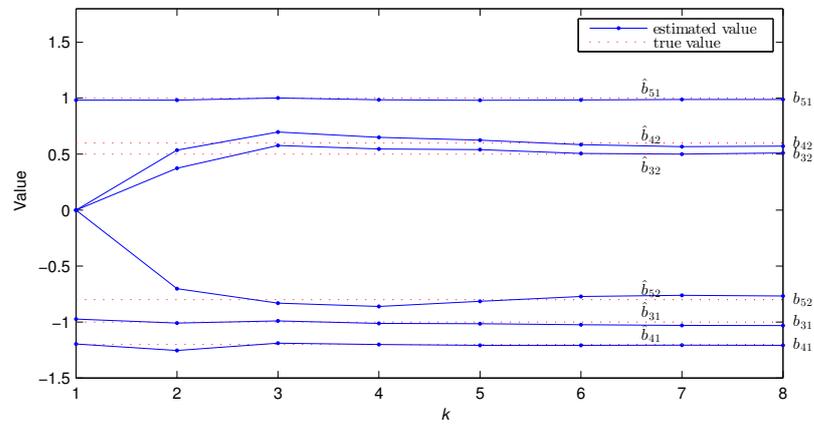


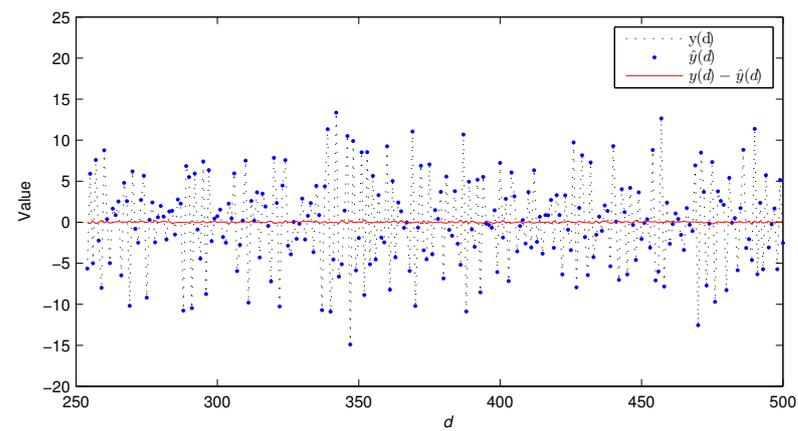
Figure 12. The estimated parameter values  $\hat{b}_{31}, \hat{b}_{32}, \hat{b}_{41}, \hat{b}_{42}, \hat{b}_{51}, \hat{b}_{52}$  versus  $k$  of Experiment 2 with  $L = 250$  and  $\sigma^2 = 0.50^2$ .



**Figure 13.** The estimated parameter values  $\hat{a}_1, \hat{a}_2, \hat{b}_{11}, \hat{b}_{12}, \hat{b}_{21}, \hat{b}_{22}, \hat{c}_1, \hat{f}_1$  versus  $k$  of Experiment 2 with  $L = 250$  and  $\sigma^2 = 1.00^2$ .



**Figure 14.** The estimated parameter values  $\hat{b}_{31}, \hat{b}_{32}, \hat{b}_{41}, \hat{b}_{42}, \hat{b}_{51}, \hat{b}_{52}$  versus  $k$  of Experiment 2 with  $L = 250$  and  $\sigma^2 = 1.00^2$ .



**Figure 15.** The true outputs, estimated outputs, and their bias of Experiment 2 with  $\sigma^2 = 0.50^2$ .

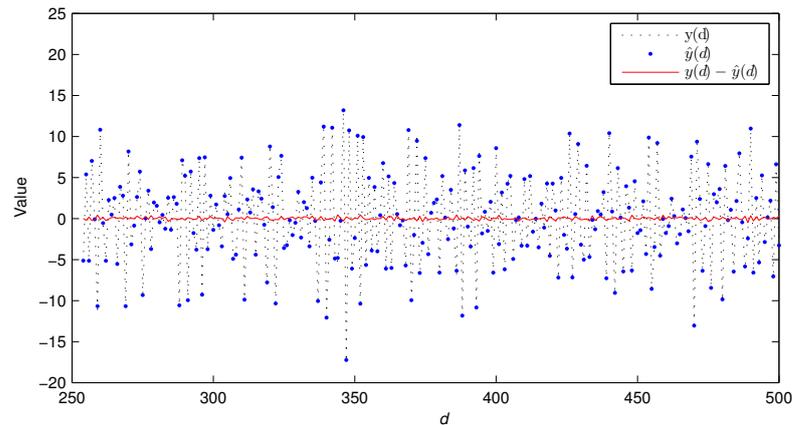


Figure 16. The true outputs, estimated outputs, and their bias of Experiment 2 with  $\sigma^2 = 1.00^2$ .

Table 5. The parameter estimation errors with different  $L$  and  $\sigma^2$ .

Sampled Data Length $L$	300	400	500	600	700	800	1000
Estimation error $\delta(\%)(\sigma^2 = 0.50^2)$	2.3844	1.7831	1.9832	2.5122	2.4627	2.388	1.8807
Estimation error $\delta(\%)(\sigma^2 = 1.00^2)$	3.641	3.2164	4.055	4.7375	4.5741	4.1934	3.2887

Table 6. The locations of the estimated key non-zero parameters of Experiment 2 with  $\sigma^2 = 0.50^2$  and  $L = 250$ .

Parameter	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_{11}$	$\hat{b}_{12}$	$\hat{b}_{21}$	$\hat{b}_{22}$	$\hat{b}_{31}$	$\hat{b}_{32}$	$\hat{b}_{41}$	$\hat{b}_{42}$	$\hat{b}_{51}$	$\hat{b}_{52}$	$\hat{c}_1$	$\hat{f}_1$
Location	1	2	12	13	76	77	118	119	183	184	220	221	253	254

According to Table 6 and Equation (31), the time delays of Experiment 2 can be estimated

$$\begin{aligned}
 \hat{\tau}_1 &= n_1 = 9, \\
 \hat{\tau}_2 &= n_2 - (l - \hat{\tau}_1 - n_{b2}) = 62 - (50 - 9 - 2) = 23, \\
 \hat{\tau}_3 &= n_3 - (l - \hat{\tau}_2 - n_{b3}) = 40 - (50 - 23 - 2) = 15, \\
 \hat{\tau}_4 &= n_4 - (l - \hat{\tau}_3 - n_{b4}) = 63 - (50 - 15 - 2) = 30, \\
 \hat{\tau}_5 &= n_5 - (l - \hat{\tau}_4 - n_{b5}) = 35 - (50 - 30 - 2) = 17.
 \end{aligned}
 \tag{33}$$

Some results can be obtained from Figures 2–16 and Tables 1–6.

- With the increase in noise variance, the error of estimated parameter vector becomes bigger, but the estimation accuracy is still higher. This shows that the GPHI algorithm has certain robustness—see Figures 2 and 9 and Tables 1 and 4.
- Compared with the traditional LSI algorithm, the GPHI algorithm can use the limited sampled data to achieve higher parameter estimation accuracy—see Figure 9.
- With different noise variances, the parameter estimates converge near the true values as the iterative times increase, which shows that GPHI algorithm has certain robustness—see Figures 3–6 and 11–14.
- The estimation accuracy of using limited sampled data is close to that of using large sampled data. This shows that the GPHI algorithm can realize parameter estimation using limited sampled data—see Tables 2 and 5.
- The estimated outputs can match the true outputs well—see Figures 7, 8, 15 and 16.
- The GPHI algorithm can accurately find the locations of key non-zero parameters, which shows that the GPHI algorithm can estimate the parameters and time delays simultaneously—see Tables 3 and 6 and Equations (32) and (33).

## 5. Conclusions

This paper solves the problem of the joint estimation of parameters and time delays of a class of multi-input systems with colored noise by using limited sampled data. A hierarchical compressed sensing framework is established, and an efficient greedy pursuit hierarchical iterative algorithm is provided. Since the over-parameterized system model is sparse, the locations of key non-zero parameters can be found by using the greedy search method to reduce estimation cost, and the parameter estimates can be identified by using the iterative method. The time delays can be estimated based on prior knowledge combined with the structure of parameter estimates. Simulation examples demonstrate that the given algorithm can efficiently estimate parameters and time delays simultaneously.

**Author Contributions:** Conceptualization, T.T.; methodology, R.D. and T.T.; software, R.D.; validation, T.T.; analysis, R.D.; writing—original draft preparation, T.T.; writing—review and editing, T.T. and R.D.; and supervision, T.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Science and Technology Project of Henan Province (No. 202102210297), the Key Research Project of Henan Higher Education Institutions (No. 21A413006), and the Science and Technology Project of Nanyang (No. JCQY2021012).

**Data Availability Statement:** Not applicable.

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

## References

1. Ljung, L. *System Identification: Theory for The User, Information and System Science Series*; Tsinghua University Press: Beijing, China, 2002.
2. Ding, F. *System Identification—New Theory and Methods*; Science Press: Beijing, China, 2013.
3. Ding, F. *System Identification—Performance Analysis for Identification Methods*; Science Press: Beijing, China, 2014.
4. Ding, F. *System Identification—Multi-Innovation Identification Theory and Methods*; Science Press: Beijing, China, 2016.
5. Xu, J.; Tao, Q.; Li, Z.; Xi, X.; Suykens, J.A.; Wang, S. Efficient hinging hyperplanes neural network and its application in nonlinear system identification. *Automatica* **2020**, *116*, 108906. [\[CrossRef\]](#)
6. Zhang, C.M.; Lu, Y. Study on artificial intelligence: The state of the art and future prospects. *J. Ind. Inf. Integr.* **2021**, *23*, 00224. [\[CrossRef\]](#)
7. Chen, J.; Huang, B.; Gan, M.; Chen, P. A novel reduced-order algorithm for rational model based on Arnoldi process and Krylov subspace. *Automatica* **2021**, *129*, 109663. [\[CrossRef\]](#)
8. Li, Y.; Yu, K. Adaptive fuzzy decentralized sampled-data control for large-scale nonlinear systems. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 1809–1822. [\[CrossRef\]](#)
9. Skarding, J.; Gabrys, B.; Musial, K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access* **2021**, *9*, 79143–79168. [\[CrossRef\]](#)
10. Liu, Y.A.; Tang, G.S.D.; Liu, Y.F.; Kong, Q.K.; Wang, J. Extended dissipative sliding mode control for nonlinear networked control systems via event-triggered mechanism with random uncertain measurement. *Appl. Math. Comput.* **2021**, *396*, 125901. [\[CrossRef\]](#)
11. Zhang, X.; Liu, Q.Y.; Ding, F.; Alsaedi, A.; Hayat, T. Recursive identification of bilinear time-delay systems through the redundant rule. *J. Frankl. Inst.* **2020**, *357*, 726–747. [\[CrossRef\]](#)
12. Gu, Y.; Liu, J.; Li, X.; Chou, Y.; Ji, Y. State space model identification of multirate processes with time-delay using the expectation maximization. *J. Frankl. Inst.* **2019**, *356*, 1623–1639. [\[CrossRef\]](#)
13. Fei, Q.L.; Ma, J.X.; Xiong, W.L.; Guo, F. Variational Bayesian identification for bilinear state space models with Markov-switching time delays. *Int. J. Robust Nonlinear Control* **2020**, *30*, 7478–7495. [\[CrossRef\]](#)
14. Ding, F.; Wang, X.H.; Mao, L.; Xu, L. Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering. *Digit. Signal Process.* **2017**, *62*, 211–223. [\[CrossRef\]](#)
15. Ding, F.; Ma, H.; Pan, J.; Yang, E.F. Hierarchical gradient and least squares-based iterative algorithms for input nonlinear output-error systems using the key term separation. *J. Frankl. Inst.* **2021**, *358*, 5113–5135. [\[CrossRef\]](#)
16. Ding, F.; Liu, X.M.; Hayat, T. Hierarchical least squares identification for feedback nonlinear equation-error systems. *J. Frankl. Inst.* **2020**, *357*, 2958–2977. [\[CrossRef\]](#)
17. Ding, F.; Wang, X.H.; Chen, Q.J.; Xiao, Y.S. Recursive least squares parameter estimation for a class of output nonlinear systems based on the model decomposition. *Circ. Syst. Signal Process.* **2016**, *35*, 3323–3338. [\[CrossRef\]](#)
18. Chen, T.; Ohlsson, H.; Ljung, L. On the estimation of transfer functions, regularizations and Gaussian processes—Revisited. *Automatica* **2012**, *48*, 1525–1535.
19. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306.
20. Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [\[CrossRef\]](#)
21. Tropp, J.A. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory* **2004**, *50*, 2231–2242. [\[CrossRef\]](#)

22. Liu, Y.J.; You, J.Y.; Ding, F. Iterative identification for multiple-input systems based on auxiliary model-orthogonal matching pursuit. *Control Decis.* **2019**, *34*, 787–792.
23. You, J.Y.; Liu, Y.J.; Chen, J.; Ding, F. Iterative identification for multiple-input systems with time-delays based on greedy pursuit and auxiliary model. *J. Frankl. Inst.* **2019**, *356*, 5819–5833.
24. You, J.Y.; Liu, Y.J. Iterative identification for multivariable systems with time-delays based on basis pursuit de-noising and auxiliary model. *Algorithms* **2018**, *11*, 180. [[CrossRef](#)]
25. Tao, T.Y.; Wang, B.; Wang, X.H. Parameter and time delay estimation algorithm based on gradient pursuit for multi-input C-ARMA systems. *Control Decis.* **2022**, *37*, 2085–2090.
26. Wang, D.Q.; Li, L.W.; Ji, Y.; Yan, Y. Model recovery for Hammerstein systems using the auxiliary model based orthogonal matching pursuit method. *Appl. Math. Model.* **2017**, *54*, 537–550. [[CrossRef](#)]
27. Mao, Y.W.; Ding, F.; Liu, Y.J. Parameter estimation algorithms for Hammerstein time-delay systems based on the orthogonal matching pursuit scheme. *IET Signal Process.* **2017**, *11*, 265–274. [[CrossRef](#)]
28. Troop, J.A. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inf. Theory* **2006**, *52*, 1030–1051. [[CrossRef](#)]
29. Ding, F. *System Identification—Iterative Search Principle and Identification Methods*; Science Press: Beijing, China, 2018.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.