

# Efficient Batch and Recursive Least Squares for Matrix Parameter Estimation

Brian Lai<sup>1</sup> and Dennis S. Bernstein<sup>1</sup>, *Life Fellow, IEEE*

**Abstract**—Traditionally, batch least squares (BLS) and recursive least squares (RLS) are used for identification of a vector of parameters which form a linear model. In some situations however, it is of interest to identify parameters in a matrix structure. In this case, a common approach is to transform the problem into standard vector form using the vectorization (vec) operator and the Kronecker product, known as vec-permutation. However, the use of the Kronecker product introduces extraneous zero terms in the regressor, resulting in unnecessary additional computational and space requirements. This letter derives matrix BLS and RLS formulations which, under mild assumptions, minimize the same cost as the vec-permutation approach. This new approach requires less computational complexity and space complexity than vec-permutation in both BLS and RLS identification. It is also shown that persistent excitation guarantees convergence to the true matrix parameters. This method can be used to improve computation time in the online identification of multiple-input, multiple-output systems for indirect adaptive model predictive control.

**Index Terms**—Identification, modeling, adaptive systems, MIMO systems.

## I. INTRODUCTION

LEAST squares based identification methods are foundational to systems and control theory, particularly identification, signal processing, and adaptive control [1], [2]. Batch least squares (BLS) and recursive least squares (RLS) are traditionally used to identify a vector of parameters in a linear measurement process [2], [3]. However, it may be of interest to identify parameters in a matrix structure, for example, in adaptive control of multiple-input, multiple-output (MIMO) systems [3], [4]. One approach is to use *vec-permutation* [5], a method which rewrites the linear measurement process such that the columns of the parameters to be identified are stacked into a vector. This is accomplished using the vectorization operator and Kronecker product, and is a straightforward solution for various situations [4], [6], [7], [8], [9], [10], [11], [12].

Manuscript received 8 March 2024; revised 4 May 2024; accepted 23 May 2024. Date of publication 5 June 2024; date of current version 27 June 2024. This work was supported by the NSF Graduate Research Fellowship under Grant DGE 1841052. Recommended by Senior Editor T. Oomen. (*Corresponding author: Brian Lai.*)

The authors are with the Department of Aerospace Engineering, University of Michigan at Ann Arbor, Ann Arbor, MI 48109 USA (e-mail: brianlai@umich.edu; dsbaero@umich.edu).

Digital Object Identifier 10.1109/LCSYS.2024.3409840

A significant drawback, however, is that the vec-permutation method increases the dimension of the linear measurement process by using the Kronecker product, introducing extraneous zero terms in the regressor (e.g., [4, eq. (15)]). This results in increased computational cost and storage requirements. Another approach is to apply standard least squares methods to separately identify the columns of the matrix of parameters [2, p. 102] but this method does not address what cost function is being minimized or the relationship to the vec-permutation approach. Other related methods including square root filtering [13], multiinnovations [14], and gradient-based methods [15] also do not address whether a least squares cost function is globally minimized or the relationship to standard least squares methods.

This letter derives a batch and recursive least squares algorithm for identification of matrix parameters which, under the assumption of independent residual error and parameter column weighting, minimizes the same cost function used in the vec-permutation approach. This method provides an  $\mathcal{O}(m^3)$  times improvement in computational complexity and an  $\mathcal{O}(m^2)$  times improvement in storage requirements over vec-permutation, where  $m \geq 1$  is the number of columns of the identified parameter matrix. We also show how persistent excitation guarantees convergence of the identified matrix parameters to true matrix parameters, which extends established results for identification of vector parameters [16], [17]. However, we show this improvement in computational complexity may come at the cost of performance if the columns of measurement noise are highly correlated. Finally, we show how this method can be used to significantly reduce computation time spent on online identification in predictive cost adaptive control (PCAC) [4].

## II. VEC-PERMUTATION LEAST SQUARES

Consider a measurement process of the form<sup>1</sup>

$$y_k = \phi_k \theta, \quad (1)$$

where  $k = 0, 1, 2, \dots$  is the time step,  $y_k \in \mathbb{R}^{p \times m}$  is the measurement at step  $k$ ,  $\phi_k \in \mathbb{R}^{p \times n}$  is the regressor at step  $k$ , and  $\theta \in \mathbb{R}^{n \times m}$  is a matrix of unknown parameters. Parameters  $\theta$  can be identified by minimizing the least squares cost function  $J_k: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ , defined as

<sup>1</sup>Note that since the measurement, regressor, and parameters are all matrices, the results of this letter can be easily extended to measurements processes of the form  $y_k = \theta \phi_k$  by rewriting as  $y_k^T = \phi_k^T \theta^T$  and identifying parameters  $\theta^T$ . For brevity, we leave the details to the reader.

$$J_k(\hat{\theta}) = \sum_{i=0}^k \text{vec}(y_i - \phi_i \hat{\theta})^T \bar{\Gamma}_i \text{vec}(y_i - \phi_i \hat{\theta}) + \text{vec}(\hat{\theta} - \theta_0)^T \bar{R} \text{vec}(\hat{\theta} - \theta_0), \quad (2)$$

where  $\text{vec}(\cdot)$  is the column stacking operator, positive-definite (and thus, by definition, symmetric)  $\bar{R} \in \mathbb{R}^{mn \times mn}$  is the regularization matrix,  $\theta_0 \in \mathbb{R}^{n \times m}$  is an initial estimate of  $\theta$ , and, for all  $k \geq 0$ , positive-definite  $\bar{\Gamma}_k \in \mathbb{R}^{mp \times mp}$  is the weighting matrix. If no estimate of the parameters  $\theta$  is known, common choices in practice for  $\theta_0$  are  $\theta_0 = 0$  or randomly sampled values. Note that while the regularization term  $\text{vec}(\hat{\theta} - \theta_0)^T \bar{R} \text{vec}(\hat{\theta} - \theta_0)$  results in a biased estimate, this term reduces variance and guarantees that the least squares cost (2) has a unique global minimizer which is useful when few data have been collected [17], [18, Sec. 7.3].

Using vec-permutation [5], (1) can be rewritten as

$$\bar{y}_k = \bar{\phi}_k \bar{\theta}, \quad (3)$$

where  $\bar{y}_k \in \mathbb{R}^{mp}$ ,  $\bar{\phi}_k \in \mathbb{R}^{mp \times mn}$ , and  $\bar{\theta} \in \mathbb{R}^{mn}$  are defined

$$\bar{y}_k \triangleq \text{vec}(y_k), \quad (4)$$

$$\bar{\phi}_k \triangleq I_m \otimes \phi_k, \quad (5)$$

$$\bar{\theta} \triangleq \text{vec}(\theta), \quad (6)$$

and where  $\otimes$  is the Kronecker product. Note, for all  $k \geq 0$  and  $\hat{\theta} \in \mathbb{R}^{n \times m}$  that  $\text{vec}(y_k - \phi_k \hat{\theta}) = \bar{y}_k - \bar{\phi}_k \text{vec}(\hat{\theta})$  and  $\text{vec}(\hat{\theta} - \theta_0) = \text{vec}(\hat{\theta}) - \bar{\theta}_0$ , where  $\bar{\theta}_0 \in \mathbb{R}^{mn}$  is defined as

$$\bar{\theta}_0 \triangleq \text{vec}(\theta_0). \quad (7)$$

It then follows that, for all  $k \geq 0$ , the cost function  $J_k$ , given in (2), can be rewritten as

$$J_k(\hat{\theta}) = \sum_{i=0}^k (\bar{y}_i - \bar{\phi}_i \text{vec}(\hat{\theta}))^T \bar{\Gamma}_i (\bar{y}_i - \bar{\phi}_i \text{vec}(\hat{\theta})) + (\text{vec}(\hat{\theta}) - \bar{\theta}_0)^T \bar{R} (\text{vec}(\hat{\theta}) - \bar{\theta}_0). \quad (8)$$

Propositions 1 gives the vec-permutation approach to minimize cost function  $J_k$ .

*Proposition 1:* For all  $k \geq 0$ , let  $\phi_k \in \mathbb{R}^{p \times n}$ , let  $y_k \in \mathbb{R}^{p \times m}$ , and let  $\bar{\Gamma}_k \in \mathbb{R}^{mp \times mp}$  be positive definite. Furthermore, let  $\bar{\theta}_0 \in \mathbb{R}^{n \times m}$  and let  $\bar{R} \in \mathbb{R}^{mn \times mn}$  be positive definite. Then, for all  $k \geq 0$ ,  $J_k: \mathbb{R}^n \rightarrow \mathbb{R}$ , defined in (2), has a unique minimizer, whose vectorization is denoted as  $\bar{\theta}_{k+1} \triangleq \text{vec}(\arg \min_{\hat{\theta} \in \mathbb{R}^{n \times m}} J_k(\hat{\theta}))$ , which is given by

$$\bar{\theta}_{k+1} = \bar{A}_k^{-1} \bar{b}_k, \quad (9)$$

where

$$\bar{A}_k \triangleq \bar{R} + \sum_{i=0}^k \bar{\phi}_i^T \bar{\Gamma}_i \bar{\phi}_i, \quad (10)$$

$$\bar{b}_k \triangleq \bar{R} \text{vec}(\theta_0) + \sum_{i=0}^k \bar{\phi}_i^T \bar{\Gamma}_i \bar{y}_i, \quad (11)$$

and where, for all  $k \geq 0$ ,  $\bar{y}_k \in \mathbb{R}^{mp}$  and  $\bar{\phi}_k \in \mathbb{R}^{mp \times mn}$  are defined in (4) and (5), respectively. Furthermore, for all  $k \geq 0$ ,  $\bar{\theta}_{k+1}$  is given recursively by

$$\bar{P}_{k+1}^{-1} = \bar{P}_k^{-1} + \bar{\phi}_k^T \bar{\Gamma}_k \bar{\phi}_k, \quad (12)$$

$$\bar{\theta}_{k+1} = \bar{\theta}_k + P_{k+1} \bar{\phi}_k^T \bar{\Gamma}_k (\bar{y}_k - \bar{\phi}_k \bar{\theta}_k). \quad (13)$$

where  $\bar{P}_0 \triangleq \bar{R}^{-1}$ , and, for all  $k \geq 0$ ,  $\bar{P}_k \in \mathbb{R}^{mn \times mn}$  is positive definite, and hence nonsingular. Moreover, for all  $k \geq 0$ ,  $\bar{P}_k$  can be expressed recursively as

$$\bar{P}_{k+1} = \bar{P}_k - \bar{P}_k \bar{\phi}_k^T (\bar{\Gamma}_k^{-1} + \bar{\phi}_k \bar{P}_k \bar{\phi}_k^T)^{-1} \bar{\phi}_k \bar{P}_k. \quad (14)$$

*Proof:* Since  $J_k: \mathbb{R}^n \rightarrow \mathbb{R}$ , defined in (2), is a standard least squares cost with vector parameters, this result follows directly from [3]. ■

Equations (9) through (13) give the batch least squares solution using vec-permutation while (12) through (14) give the recursive least squares solution using vec-permutation. An inefficiency with this method is that the Kronecker product in (5) introduces extraneous zero terms in  $\bar{\phi}_k$  when  $m > 1$ , resulting in a sparse and higher dimensional regressor matrix. However, the results of Proposition 1 cannot be simplified since the regularization matrix  $\bar{R}$  and weighting matrices  $\bar{\Gamma}_k$  are not necessarily sparse.

### III. COLUMN-BY-COLUMN LEAST SQUARES

To simplify the vec-permutation approach, we make the assumption that there exist positive-definite  $R_1, \dots, R_m \in \mathbb{R}^{n \times n}$  such that  $\bar{R}$  is block diagonal of the form

$$\bar{R} = \text{diag}(R_1, \dots, R_m). \quad (15)$$

Furthermore, we assume that, for all  $k \geq 0$ , there exist positive-definite  $\Gamma_{1,k}, \dots, \Gamma_{m,k} \in \mathbb{R}^{p \times p}$  such that  $\bar{\Gamma}_k$  is block diagonal of the form

$$\bar{\Gamma}_k = \text{diag}(\Gamma_{1,k}, \dots, \Gamma_{m,k}). \quad (16)$$

This corresponds to independent weighting of the columns of the residual error,  $y_i - \phi_i \hat{\theta}$ ,  $i = 0, \dots, k$ , and of the regularization term,  $\hat{\theta} - \theta_0$ , in (2). Then, for all  $k \geq 0$  and  $\hat{\theta} \in \mathbb{R}^{n \times m}$ , (2) can be rewritten as

$$J_k(\hat{\theta}) = \sum_{j=0}^m J_{j,k}(\hat{\theta}_j), \quad (17)$$

where, for all  $j = 1, \dots, m$ ,  $J_{j,k}: \mathbb{R}^n \rightarrow \mathbb{R}$  is defined as

$$J_{j,k}(\hat{\theta}_j) = \sum_{i=0}^k (y_{j,i} - \phi_i \hat{\theta}_j)^T \Gamma_{j,i} (y_{j,i} - \phi_i \hat{\theta}_j) + (\hat{\theta}_j - \theta_{j,0})^T R_j (\hat{\theta}_j - \theta_{j,0}). \quad (18)$$

where the vectors  $y_{1,k}, \dots, y_{m,k} \in \mathbb{R}^p$ ,  $\theta_{1,0}, \dots, \theta_{m,0} \in \mathbb{R}^n$ , and  $\hat{\theta}_1, \dots, \hat{\theta}_m \in \mathbb{R}^n$  are the  $m$  columns of  $y_k$ ,  $\theta_0$ , and  $\hat{\theta}$ , respectively. In particular,

$$y_k \triangleq [y_{1,k} \quad \dots \quad y_{m,k}], \quad (19)$$

$$\theta_0 \triangleq [\theta_{1,0} \quad \dots \quad \theta_{m,0}], \quad (20)$$

$$\hat{\theta} \triangleq [\hat{\theta}_1 \quad \dots \quad \hat{\theta}_m]. \quad (21)$$

Propositions 2 show that, under assumptions (15) and (16), the cost function  $J_k$ , given by (17), can be minimized by separately updating the columns of the parameter estimate. We call this the *column-by-column* approach.

*Proposition 2:* For all  $k \geq 0$ , let  $\phi_k \in \mathbb{R}^{p \times n}$ , let  $y_k \in \mathbb{R}^{p \times m}$ , and let  $\Gamma_{1,k}, \dots, \Gamma_{m,k} \in \mathbb{R}^{p \times p}$  be positive definite.

Furthermore, let  $\bar{\theta}_0 \in \mathbb{R}^{n \times m}$  and let  $R_1, \dots, R_m \in \mathbb{R}^{n \times n}$  be positive definite. Then, for all  $k \geq 0$ ,  $J_k: \mathbb{R}^n \rightarrow \mathbb{R}$ , defined in (17), has a unique minimizer, whose columns are denoted as

$$[\theta_{1,k+1} \quad \dots \quad \theta_{m,k+1}] \triangleq \arg \min_{\hat{\theta} \in \mathbb{R}^{n \times m}} J_k(\hat{\theta}), \quad (22)$$

which, for all  $j = 1, \dots, m$ , are given by

$$\theta_{j,k+1} = A_{j,k}^{-1} b_{j,k}, \quad (23)$$

where

$$A_{j,k} \triangleq R_j + \sum_{i=0}^k \phi_i^T \Gamma_{j,i} \phi_i, \quad (24)$$

$$b_{j,k} \triangleq R_j \theta_{j,0} + \sum_{i=0}^k \phi_i^T \Gamma_{j,i} y_{j,i}, \quad (25)$$

and where, for all  $k \geq 0$ ,  $y_{j,k} \in \mathbb{R}^p$  is defined in (19) and  $\theta_{j,0} \in \mathbb{R}^n$  is defined (20). Furthermore, for all  $k \geq 0$  and  $j = 1, \dots, m$ ,  $\theta_{j,k+1} \in \mathbb{R}^n$  is given recursively by

$$P_{j,k+1}^{-1} = P_{j,k}^{-1} + \phi_k^T \Gamma_{j,k} \phi_k, \quad (26)$$

$$\theta_{j,k+1} = \theta_{j,k} + P_{j,k+1} \phi_k^T \Gamma_{j,k} (y_{j,k} - \phi_k \theta_{j,k}). \quad (27)$$

where  $P_{j,0} \triangleq R_j^{-1}$ , and, for all  $k \geq 0$  and  $j = 1, \dots, m$ ,  $P_{j,k} \in \mathbb{R}^{n \times n}$  is positive definite, and hence nonsingular. Moreover, for all  $k \geq 0$  and  $j = 1, \dots, m$ ,  $P_{j,k}$  can be expressed recursively as

$$P_{j,k+1} = P_{j,k} - P_{j,k} \phi_k^T (\Gamma_{j,k}^{-1} + \phi_k P_{j,k} \phi_k^T)^{-1} \phi_k P_{j,k}. \quad (28)$$

*Proof:* Note that, for all  $j = 1, \dots, m$ ,  $J_{j,k}$  is a function of only  $\hat{\theta}_j$  and is a standard least squares cost with vector parameters. It then follows from [3] that, for all  $j = 1, \dots, m$ ,  $J_{j,k}$  has a unique minimizer, denoted as  $\theta_{j,k+1} \triangleq \arg \min_{\hat{\theta}_j \in \mathbb{R}^n} J_{j,k}(\hat{\theta}_j)$ , which is given by (23). It also follows from [3] that (24) through (28) hold. Finally, it follows from (17) and (21) that

$$\begin{aligned} & \arg \min_{\hat{\theta} \in \mathbb{R}^{n \times m}} J_k(\hat{\theta}) \\ &= \left[ \arg \min_{\hat{\theta}_1 \in \mathbb{R}^n} J_{1,k}(\hat{\theta}_1) \quad \dots \quad \arg \min_{\hat{\theta}_m \in \mathbb{R}^n} J_{m,k}(\hat{\theta}_m) \right] \end{aligned}$$

and (22) follows.  $\blacksquare$

An advantage of the column-by-column approach versus vec-permutation is that no Kronecker product is used, implying that no sparse matrices are introduced. Note that the column-by-column approach can also be derived by applying standard least squares methods to the  $m$  columns of (1) [2, p. 102]. However, our derivation further shows the connection to vec-permutation and how the column-by-column approach implicitly implies independent weighting to the columns of the residual error and parameter regularization.

#### IV. MATRIX UPDATE LEAST SQUARES

Finally, we make the stronger assumption that there exists positive-definite  $R \in \mathbb{R}^{n \times n}$  such that  $\bar{R}$  is block diagonal of the form

$$\bar{R} = \text{diag}(R, \dots, R) = I_m \otimes R. \quad (29)$$

Furthermore, we assume that, for all  $k \geq 0$ , there exist positive-definite  $\Gamma_k \in \mathbb{R}^{p \times p}$  such that  $\bar{\Gamma}_k$  is block diagonal of the form

$$\bar{\Gamma}_k = \text{diag}(\Gamma_k, \dots, \Gamma_k) = I_m \otimes \Gamma_k. \quad (30)$$

This corresponds to independent and identical weighting of the columns of the residual error,  $y_i - \phi_i \hat{\theta}$ ,  $i = 0, \dots, k$ , and of the regularization term,  $\hat{\theta} - \theta_0$ , in (2). Then, it follows from Lemma 1 that, for all  $k \geq 0$ , (2) can be rewritten as

$$J_k(\hat{\theta}) = \text{tr} \left[ \sum_{i=0}^k (y_i - \phi_i \hat{\theta})^T \Gamma_i (y_i - \phi_i \hat{\theta}) + (\hat{\theta} - \theta_0)^T R (\hat{\theta} - \theta_0) \right]. \quad (31)$$

Propositions 3 show that, under assumptions (29) and (30), the cost function  $J_k$ , given by (31), can be minimized as a single matrix equation. We call this the *matrix update* approach.

*Proposition 3:* For all  $k \geq 0$ , let  $\phi_k \in \mathbb{R}^{p \times n}$ , let  $y_k \in \mathbb{R}^{p \times m}$ , and let  $\Gamma_k \in \mathbb{R}^{p \times p}$  be positive definite. Furthermore, let  $\theta_0 \in \mathbb{R}^{n \times m}$  and let  $R \in \mathbb{R}^{n \times n}$  be positive definite. Then, for all  $k \geq 0$ ,  $J_k: \mathbb{R}^n \rightarrow \mathbb{R}$ , defined in (31), has a unique minimizer, denoted as  $\theta_{k+1} \triangleq \arg \min_{\hat{\theta} \in \mathbb{R}^{n \times m}} J_k(\hat{\theta})$ , which is given by

$$\theta_{k+1} = A_k^{-1} b_k, \quad (32)$$

where

$$A_k \triangleq R + \sum_{i=0}^k \phi_i^T \Gamma_i \phi_i, \quad (33)$$

$$b_k \triangleq R \theta_0 + \sum_{i=0}^k \phi_i^T \Gamma_i y_i. \quad (34)$$

Furthermore, for all  $k \geq 0$ ,  $\theta_{k+1} \in \mathbb{R}^{n \times m}$  is given recursively by

$$P_{k+1}^{-1} = P_k^{-1} + \phi_k^T \Gamma_k \phi_k, \quad (35)$$

$$\theta_{k+1} = \theta_k + P_{k+1} \phi_k^T \Gamma_k (y_k - \phi_k \theta_k). \quad (36)$$

where  $P_0 \triangleq R^{-1}$  and, for all  $k \geq 0$ ,  $P_k \in \mathbb{R}^{n \times n}$  is positive definite, hence nonsingular. Moreover, for all  $k \geq 0$ ,  $P_{k+1}$  can be expressed recursively as

$$P_{k+1} = P_k - P_k \phi_k^T (\Gamma_k^{-1} + \phi_k P_k \phi_k^T)^{-1} \phi_k P_k. \quad (37)$$

*Proof:* Note that (31) can be rewritten as (17) where, for all  $k \geq 0$  and  $j = 1, \dots, m$ ,  $\Gamma_{j,k} = \Gamma_k$  and  $R_j = R$ . It then follows from Proposition 2 that  $J_k$  has a unique minimizer given by  $\theta_{k+1} = [A_k^{-1} b_{1,k} \quad \dots \quad A_k^{-1} b_{m,k}] = A_k^{-1} [b_{1,k} \quad \dots \quad b_{m,k}]$ , where  $A_k \in \mathbb{R}^{n \times n}$  is defined in (33) and  $b_{1,k}, \dots, b_{m,k}$  are defined in (25). Finally, note that  $b_k = [b_{1,k} \quad \dots \quad b_{m,k}]$ , yielding (32). Moreover, it also follows from Proposition 2 that, for all  $k \geq 0$ ,  $P_{k+1}^{-1} = P_k^{-1} + \phi_k^T \Gamma_k \phi_k$ , and  $\theta_{j,k+1} = \theta_{j,k} + P_{k+1} \phi_k^T \Gamma_k (y_{j,k} - \phi_k \theta_{j,k})$ , where, for all  $j = 1, \dots, m$ ,  $\theta_{j,k} \in \mathbb{R}^n$  and  $y_{j,k} \in \mathbb{R}^p$  are the  $j$ 'th columns of  $\theta_k$  and  $y_k$ , respectively. Combining columns yields (36) and applying matrix inversion lemma to (35) yields (37).  $\blacksquare$

Similarly, while the matrix update approach can be derived by applying standard least squares methods to the  $m$  columns of (1) and combining columns [2, p. 103], our derivation shows how the matrix update approach implicitly implies independent and identical weighting to the columns of the residual error and parameter regularization.

TABLE I  
BATCH LEAST SQUARES SUMMARY AND COMPUTATIONAL COMPLEXITIES FOR  $N$  MEASUREMENTS

	Algorithm	Comp. Complexity (No Assumptions)	Comp. Complexity ( $N \gg n \geq p$ )	Comp. Complexity ( $N \gg p \geq n$ )
Vec-Permutation	(9), (10), (11)	$\mathcal{O}(\max\{Npnm^3 \max\{p, n\}, n^3m^3\})$	$\mathcal{O}(Npn^2m^3)$	$\mathcal{O}(Np^2nm^3)$
Column-by-Column	(23), (24), (25)	$\mathcal{O}(\max\{Npnm \max\{p, n\}, n^3m\})$	$\mathcal{O}(Npn^2m)$	$\mathcal{O}(Np^2nm)$
Matrix Update	(32), (33), (34)	$\mathcal{O}(\max\{Np \max\{n, m\} \max\{p, n\}, n^3\})$	$\mathcal{O}(Npn \max\{n, m\})$	$\mathcal{O}(Np^2 \max\{n, m\})$

TABLE II  
RECURSIVE LEAST SQUARES SUMMARY AND COMPUTATIONAL/SPACE COMPLEXITIES PER STEP

	Algorithm ( $n \gg p$ )	Comp. Complexity ( $n \gg p$ )	Algorithm ( $p \geq n$ )	Comp. Complexity ( $p \geq n$ )	Number of Parameters in Memory
Vec-Permutation	(14), (13)	$\mathcal{O}(pn^2m^3)$	(12), (13)	$\mathcal{O}(p^2nm^3)$	$n^2m^2 + nm$
Column-by-Column	(28), (27)	$\mathcal{O}(pn^2m)$	(26), (27)	$\mathcal{O}(p^2nm)$	$n^2m + nm$
Matrix Update	(37), (36)	$\mathcal{O}(pn \max\{n, m\})$	(35), (36)	$\mathcal{O}(p^2 \max\{n, m\})$	$n^2 + nm$

### A. Convergence of Matrix Update RLS

It is well-known that in standard RLS, the parameter estimate vector converges to the vector of true parameters if the sequence of regressors  $(\phi_k)_{k=0}^{\infty}$  is persistently exciting [16], [17]. Theorem 1 extends this result to matrix RLS. To begin, we extend the definition of persistent excitation (PE) from [1, p. 64] to the case of matrix regressors and nonuniform weight.<sup>2</sup>

*Definition 1:*  $(\phi_k)_{k=0}^{\infty} \subset \mathbb{R}^{p \times n}$  with weight  $(\Gamma_k)_{k=0}^{\infty}$  is persistently exciting (PE) if

$$C \triangleq \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=0}^{k-1} \phi_i^T \Gamma_i \phi_i \in \mathbb{R}^{n \times n} \quad (38)$$

exists and is positive definite.

*Theorem 1:* Let  $\theta, \theta_0 \in \mathbb{R}^{n \times m}$  and let  $R \in \mathbb{R}^{n \times n}$  be positive definite. For all  $k \geq 0$ , let  $\phi_k \in \mathbb{R}^{p \times n}$ , let  $y_k \in \mathbb{R}^{p \times m}$  be given by (1), let  $\Gamma_k \in \mathbb{R}^{p \times p}$  be positive definite, and let  $P_k \in \mathbb{R}^{n \times n}$  and  $\theta_k \in \mathbb{R}^{n \times m}$  be given by (35) and (36), respectively. Assume that  $(\phi_k)_{k=0}^{\infty}$  with weight  $(\Gamma_k)_{k=0}^{\infty}$  is PE, and define  $C \in \mathbb{R}^{n \times n}$  by (38). Then,

$$\lim_{k \rightarrow \infty} k(\theta_k - \theta) = C^{-1}R(\theta_0 - \theta). \quad (39)$$

*Proof:* Note that

$$\begin{aligned} \theta_k &= \left( R + \sum_{i=0}^{k-1} \phi_i^T \Gamma_i \phi_i \right)^{-1} \left( R\theta_0 + \sum_{i=0}^{k-1} \phi_i^T \Gamma_i y_i \right) \\ &= \left( R + \sum_{i=0}^{k-1} \phi_i^T \Gamma_i \phi_i \right)^{-1} \left[ R(\theta_0 - \theta) + \left( R + \sum_{i=0}^{k-1} \phi_i^T \Gamma_i \phi_i \right) \theta \right] \\ &= \left( R + \sum_{i=0}^{k-1} \phi_i^T \Gamma_i \phi_i \right)^{-1} R(\theta_0 - \theta) + \theta. \end{aligned}$$

Hence, it follows that

$$\begin{aligned} \lim_{k \rightarrow \infty} k(\theta_k - \theta) &= \lim_{k \rightarrow \infty} \left( \frac{1}{k} R + \frac{1}{k} \sum_{i=0}^{k-1} \phi_i^T \Gamma_i \phi_i \right)^{-1} R(\theta_0 - \theta) \\ &= C^{-1}R(\theta_0 - \theta). \quad \blacksquare \end{aligned}$$

<sup>2</sup>Note that [1] considers a measurement process  $y_k = \phi_k^T \theta$  while we consider  $y_k = \phi_k \theta$ . As such, PE is defined in [1] using the limit  $\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=0}^{k-1} \phi_i \phi_i^T$  whereas we consider  $\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=0}^{k-1} \phi_i^T \phi_i$ .

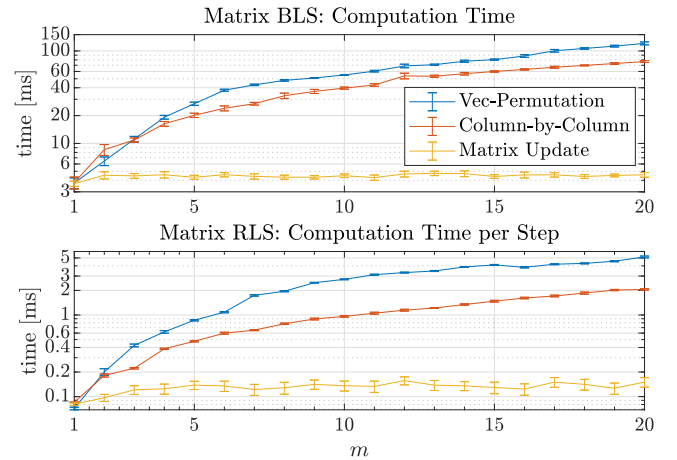


Fig. 1. Consider the measurement process (1) with  $p = 10$ ,  $n = 50$ , and  $1 \leq m \leq 20$ . Batch least squares (top) shows computation time with  $N = 100$  data points, averaged over 10 trials. Recursive least squares (bottom) shows computation time per step, averaged over 100 trials. Error bars show the 95% confidence intervals.

### V. COMPUTATIONAL COMPLEXITY AND PERFORMANCE TRADEOFF

Next, we study the computational complexities of vec-permutation, column-by-column, and matrix update least squares. For simplicity, we assume it takes  $\mathcal{O}(nmp)$  arithmetic operations to multiply an  $(n \times m)$  matrix with a  $(m \times p)$  matrix,  $\mathcal{O}(n^3)$  arithmetic operations to invert an  $(n \times n)$  matrix, and  $\mathcal{O}(nm)$  arithmetic operations to add two  $(n \times m)$  matrices.

The computational complexities of the batch and recursive least squares methods are shown in Tables I and II respectively. Note that for the RLS methods, columns 2 and 4 of Table II show the most efficient implementation depending on the dimensions  $n$  and  $p$ . Note that for both BLS and RLS, column-by-column and matrix update result in an  $\mathcal{O}(m^2)$  and  $\mathcal{O}(m^3)$  times improvement in computational complexity over vec-permutation, respectively. Moreover, for RLS, column-by-column and matrix update result in an  $\mathcal{O}(m)$  and  $\mathcal{O}(m^2)$  times improvement in space complexity over vec-permutation, respectively. Next, Figure 1 shows numerical testing of BLS and RLS with vec-permutation, column-by-column, and matrix



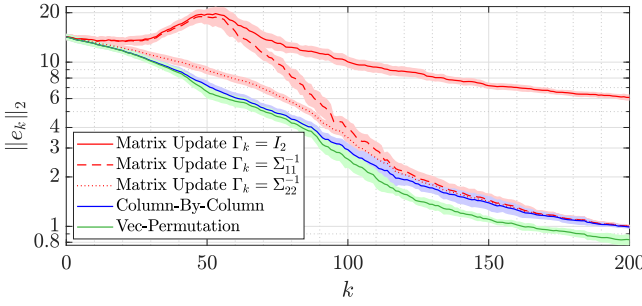


Fig. 2. Example 1: Parameter estimation error  $\|e_k\|_2$  over 200 steps for matrix update, column-by-column, and vec-permutation recursive least squares with 95% confidence intervals over 10 trials highlighted.

update. We consider the measurement process (1) with  $p = 10$ ,  $n = 50$ , and  $1 \leq m \leq 20$ . For larger values of  $m$ , we see significantly faster computation time for matrix update over vec-permutation and column-by-column.

While matrix update least squares offers a significant improvement in computational cost for large values of  $m$ , the following example shows that there may be a sacrifice in performance if there is prior knowledge that measurement noise has highly correlated columns. While not shown in this example, there may also be performance sacrifice if there is prior knowledge that columns of the parameters are highly correlated.

*Example 1:* Consider the measurement process (1) with  $p = 2$ ,  $m = 2$ , and  $n = 100$ . We consider 10 independent trials where, for each trial, the two columns of the parameters  $\theta$  are i.i.d. sampled from the Gaussian distribution  $\mathcal{N}(0_{n \times 1}, I_n)$ . For each trial, for all  $k \geq 0$ , the two rows of the regressor,  $\phi_k$  are i.i.d. sampled from  $\mathcal{N}(0_{n \times 1}, I_n)$ , and the measurement  $y_k$  is given by  $y_k = \phi_k \theta + v_k$ , where  $v_k \in \mathbb{R}^{p \times m}$  is the measurement noise and  $\text{vec}(v_k)$  is i.i.d. sampled from  $\mathcal{N}(0_{pm \times 1}, \Sigma)$ , where

$$\Sigma \triangleq \begin{bmatrix} \Sigma_{11} & 9.9 \mathbf{1}_{2 \times 2} \\ 9.9 \mathbf{1}_{2 \times 2} & \Sigma_{22} \end{bmatrix}, \quad \Sigma_{11} \triangleq \begin{bmatrix} 1 & 0.99 \\ 0.99 & 1 \end{bmatrix} \quad (40)$$

and  $\Sigma_{22} \triangleq 100 \Sigma_{11}$ . In this setup, the second column of the measurement noise has higher variance than that of the first column, and any two elements of the measurement noise are highly correlated with a correlation coefficient of 0.99.

We compare the performance of vec-permutation, column-by-column, and matrix update RLS. We set regularization terms as  $\theta_0 = 0_{nm \times 1}$  and  $\bar{R} = I_{nm}$ ,  $\theta_{j,0} = 0_{n \times 1}$  and  $R_j = I_n$   $j = 1, 2$ , and  $\theta_0 = 0_{n \times m}$  and  $R = I_n$ , respectively. For vec-permutation, we let  $\bar{\Gamma}_k = \Sigma^{-1}$ , giving the minimum variance estimator. For column-by-column, we let  $\Gamma_{1,k} = \Sigma_{11}^{-1}$  and  $\Gamma_{2,k} = \Sigma_{22}^{-1}$ . Finally, for matrix update, we consider the three choices  $\Gamma_k = I_2$ ,  $\Gamma_k = \Sigma_{11}^{-1}$ , and  $\Gamma_k = \Sigma_{22}^{-1}$ .

Figure 2 shows  $\|e_k\|_2$ , the error at step  $k$ , defined as  $\|\hat{\theta}_k - \bar{\theta}\|_2$ ,  $\|[\theta_{1,k} \ \theta_{2,k}] - \theta_k\|_2$ , and  $\|\theta_k - \theta\|_2$  for vec-permutation, column-by-column, and matrix update RLS, respectively. The highlighted region gives the 95% confidence interval. Note how identification performance of matrix update least squares varies considerable based on the choice of  $\Gamma_k$ . Vec-permutation performs similarly to column-by-column when  $k \leq n = 100$  but gives slightly better performance asymptotic once  $k > n$ .

## VI. APPLICATION TO ONLINE IDENTIFICATION FOR INDIRECT ADAPTIVE MODEL PREDICTIVE CONTROL

Finally, a useful application of this letter is efficient online identification for adaptive model predictive control. Consider a MIMO input-output system of the form

$$y_k = - \sum_{i=1}^{\hat{n}} F_i y_{k-i} + \sum_{i=0}^{\hat{n}} G_i u_{k-i}, \quad (41)$$

where  $k \geq 0$  is the time step,  $\hat{n}$  is the model order,  $u_k \in \mathbb{R}^m$  is the control,  $y_k \in \mathbb{R}^p$  is the measurement, and  $F_1, \dots, F_{\hat{n}} \in \mathbb{R}^{p \times p}$  and  $G_0, \dots, G_{\hat{n}} \in \mathbb{R}^{p \times m}$  are the system coefficient matrices to be estimated. A model of the form (41) is identified online in the indirect adaptive model predictive control scheme: predictive cost adaptive control (PCAC) [4]. For all  $k \geq 0$ , the system coefficient matrices are estimated by minimizing the cost function  $J_k: \mathbb{R}^{p \times \hat{n}(m+p)+m} \rightarrow \mathbb{R}$ , defined as

$$J_k(\hat{\theta}) = \sum_{i=0}^k z_i^T(\hat{\theta}) z_i(\hat{\theta}) + \text{vec}(\hat{\theta} - \theta_0)^T \bar{P}_0^{-1} \text{vec}(\hat{\theta} - \theta_0), \quad (42)$$

where  $z_k: \mathbb{R}^{p \times \hat{n}(m+p)+m} \rightarrow \mathbb{R}^p$  is defined

$$z_k(\hat{\theta}) \triangleq y_k + \sum_{i=1}^{\hat{n}} \hat{F}_i y_{k-i} - \sum_{i=0}^{\hat{n}} \hat{G}_i u_{k-i}, \quad (43)$$

$\hat{\theta} \in \mathbb{R}^{p \times \hat{n}(m+p)+m}$  are the coefficients to be estimated, defined

$$\hat{\theta} \triangleq [\hat{F}_1 \ \dots \ \hat{F}_{\hat{n}} \ \hat{G}_0 \ \dots \ \hat{G}_{\hat{n}}], \quad (44)$$

and where  $\theta_0 \in \mathbb{R}^{p \times \hat{n}(m+p)+m}$  is an initial guess of the coefficients and  $\bar{P}_0 \in \mathbb{R}^{[\hat{n}p(m+p)+mp] \times [\hat{n}p(m+p)+mp]}$  is positive definite. Note that, for all  $k \geq 0$ ,  $z_k(\hat{\theta})$  can be written as

$$z_k(\hat{\theta}) = y_k - \hat{\theta} \phi_k, \quad (45)$$

where  $\phi_k \in \mathbb{R}^{\hat{n}(m+p)+m}$  is defined as

$$\phi_k \triangleq [-y_{k-1}^T \ \dots \ -y_{k-\hat{n}}^T \ u_k^T \ \dots \ u_{k-\hat{n}}^T]^T. \quad (46)$$

Further defining  $\bar{\phi}_k \in \mathbb{R}^{p \times \hat{n}p(m+p)+mp}$  as

$$\bar{\phi}_k \triangleq \phi_k^T \otimes I_p, \quad (47)$$

it follows that  $z_k(\hat{\theta})$  can be written as

$$z_k(\hat{\theta}) = y_k - \bar{\phi}_k \text{vec}(\hat{\theta}) \quad (48)$$

where  $\text{vec}(\hat{\theta}) \in \mathbb{R}^{\hat{n}p(m+p)+mp}$  is the vectorization of  $\hat{\theta}$ . Using (48), we derive the identification algorithm used in [4].

*Proposition 4:* For all  $k \geq 0$ , let  $u_k \in \mathbb{R}^m$ ,  $y_k \in \mathbb{R}^p$ . Furthermore, let  $\theta_0 \in \mathbb{R}^{p \times \hat{n}(m+p)+m}$  and let  $\bar{P}_0 \in \mathbb{R}^{[\hat{n}p(m+p)+mp] \times [\hat{n}p(m+p)+mp]}$  be positive definite. Then, for all  $k \geq 0$ ,  $J_k$ , defined in (42), has a unique global minimizer, denoted

$$\theta_{k+1} \triangleq \arg \min_{\hat{\theta} \in \mathbb{R}^{p \times \hat{n}(m+p)+m}} J_k(\hat{\theta}), \quad (49)$$

which is given by

$$\bar{P}_{k+1} = \bar{P}_k - \bar{P}_k \bar{\phi}_k^T (I_p + \bar{\phi}_k \bar{P}_k \bar{\phi}_k^T)^{-1} \bar{\phi}_k \bar{P}_k, \quad (50)$$

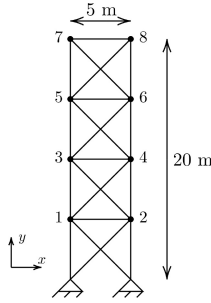


Fig. 3. Flexible truss structure from [10] with nodes labeled.

TABLE III

TRUSS EX. COMPUTATION TIME PER STEP: MEAN  $\pm$  STD. DEVIATION

	Vec-Permutation	Matrix RLS	Change
ID Time per Step	(5.9 $\pm$ 0.5) ms	(0.14 $\pm$ 0.03) ms	-97.6 %
Total Time per Step	(28 $\pm$ 6) ms	(22 $\pm$ 6) ms	-21.4 %

$$\text{vec}(\theta_{k+1}) = \text{vec}(\theta_k) + \bar{P}_{k+1} \bar{\phi}_k^T (y_k - \bar{\phi}_k \text{vec}(\theta_k)). \quad (51)$$

*Proof:* This result follows from Proposition 1. For further details, see [4, eqs. (8)–(20)]. ■

Next, we provide an alternate formulation using matrix RLS.

*Proposition 5:* Consider the notation and assumptions of Proposition 4. If there exists  $P_0 \in \mathbb{R}^{[\hat{n}(m+p)+m] \times [\hat{n}(m+p)+m]}$  such that  $\bar{P}_0 = P_0 \otimes I_p$ , then, for all  $k \geq 0$ ,  $\theta_{k+1} \in \mathbb{R}^{p \times \hat{n}(m+p)+m}$  is given by

$$P_{k+1} = P_k - \frac{P_k \phi_k \phi_k^T P_k}{1 + \phi_k^T P_k \phi_k}, \quad (52)$$

$$\theta_{k+1} = \theta_k + (y_k - \theta_k \phi_k) \phi_k^T P_{k+1}. \quad (53)$$

*Proof:* This result follows from Proposition 3. ■

*Example 2:* This example is from [10] and uses PCAC for the control of a flexible structure under harmonic and broadband disturbances. Consider the 4-bay truss show in Figure 3 made of flexible truss elements with unknown mass and stiffness. Two actuators are placed at nodes 3 and 4 with control authority in the  $x$ -direction and  $x$ -direction displacement sensors are placed at nodes 5, 6, 7, and 8. The objective is to use PCAC to suppress the effects of broadband disturbances with constraints on actuator force and without prior knowledge of the truss dynamics. See [10] for further details.

This example has inputs  $u_k \in \mathbb{R}^2$  and outputs  $y_k \in \mathbb{R}^4$ . Online identification was done in [10] using vec-permutation, given by (50) and (51), with identity regularization. We replicated the results of [10] using matrix RLS, given by (52) and (53). Table III shows that using matrix RLS resulted in a 97.6% decrease in computation time needed for system identification per step. Moreover, since system identification is a significant part of PCAC, Table III also shows a 21.4% decrease in total computation time per step.

## VII. CONCLUSION

This letter derives batch and recursive least squares algorithms for the identification of matrix parameters. Under

the assumption of independent, identical column weighting, this method minimizes the same cost function as the vec-permutation approach while significantly improving computational complexity. It is also shown how, under persistent excitation, convergence guarantees can be extended from the vector case to the matrix case. This approach can be used fast online identification of MIMO systems which is critical in indirect adaptive model predictive control. A future area of interest is studying how various RLS forgetting algorithms (e.g., [19]) can be applied to matrix update RLS.

## APPENDIX

*Lemma 1:* Let  $x \in \mathbb{R}^{n \times m}$  and let  $A \in \mathbb{R}^{n \times n}$ . Then,  $\text{vec}(x)^T (I_m \otimes A) \text{vec}(x) = \text{tr}(x^T A x)$ .

## REFERENCES

- [1] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publ. Co., Inc., 1994.
- [2] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA, USA: MIT press, 1983.
- [3] S. Islam and D. S. Bernstein, "Recursive least squares for real-time implementation," *IEEE Control Sys. Mag.*, vol. 39, no. 3, pp. 82–85, Jun. 2019.
- [4] T. W. Nguyen et al., "Predictive cost adaptive control: A numerical investigation of persistency, consistency, and exigency," *IEEE Control Syst. Mag.*, vol. 41, no. 6, pp. 64–96, Dec. 2021.
- [5] H. V. Henderson and S. R. Searle, "The vec-permutation matrix, the vec operator and kronecker products: A review," *Linear Multilinear Algebra*, vol. 9, no. 4, pp. 271–288, 1981.
- [6] S. A. U. Islam, T. W. Nguyen, I. V. Kolmanovskiy, and D. S. Bernstein, "Data-driven retrospective cost adaptive control for flight control applications," *J. Guid., Control, Dyn.*, vol. 44, no. 10, pp. 1732–1758, 2021.
- [7] K. Zhu, C. Yu, and Y. Wan, "Recursive least squares identification with variable-direction forgetting via oblique projection decomposition," *J. Automatica Sinica*, vol. 9, no. 3, pp. 547–555, 2021.
- [8] F. Ding, "Coupled-least-squares identification for multivariable systems," *Inst. Eng. Technol. Control Theory Appl.*, vol. 7, no. 1, pp. 68–79, 2013.
- [9] Y. Wang, F. Ding, and M. Wu, "Recursive parameter estimation algorithm for multivariate output-error systems," *J. Franklin. Inst.*, vol. 355, no. 12, pp. 5163–5181, 2018.
- [10] N. Mohseni and D. S. Bernstein, "Predictive cost adaptive control of flexible structures with harmonic and broadband disturbances," in *Proc. Am. Control Conf. (ACC)*, 2022, pp. 3198–3203.
- [11] A. Farahmandi and B. Reitz, "Predictive cost adaptive control of a planar missile with unmodeled aerodynamics," in *Proc. AIAA SCITECH Forum*, 2024, p. 2218.
- [12] H. Ma, J. Pan, L. Lv, G. Xu, F. Ding, A. Alsaedi, and T. Hayat, "Recursive algorithms for multivariable output-error-like ARMA systems," *Mathematics*, vol. 7, no. 6, p. 558, 2019.
- [13] V. Peterka, "A square root filter for real time multivariate regression," *Kybernetika*, vol. 11, no. 1, pp. 53–67, 1975.
- [14] F. Ding, P. X. Liu, and G. Liu, "Multiinnovation least-squares identification for system modeling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 767–778, Jun. 2010.
- [15] B. Bamieh and L. Giarre, "Identification of linear parameter varying models," *Int. J. Robust Nonlin. Control, IFAC-Affiliat. J.*, vol. 12, no. 9, pp. 841–853, 2002.
- [16] A. L. Bruce, A. Goel, and D. S. Bernstein, "Necessary and sufficient regressor conditions for the global asymptotic stability of recursive least squares," *Syst. Control Lett.*, vol. 157, Nov. 2021, Art. no. 105005.
- [17] B. Lai, S. A. U. Islam, and D. S. Bernstein, "Regularization-induced bias and consistency in recursive least squares," in *Proc. Am. Control Conf. (ACC)*, 2021, pp. 3987–3992.
- [18] T. Hastie et al., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2009.
- [19] B. Lai and D. S. Bernstein, "Generalized forgetting recursive least squares: Stability and robustness guarantees," *IEEE Trans. Autom. Control*, early access, Apr. 26, 2024, doi: 10.1109/TAC.2024.3394351.