

Article

Validation of Parallel Distributed Adaptive Signal Processing (PDASP) Framework through Processing-Inefficient Low-Cost Platforms

Hasan Raza ¹, Ishtiaq Ahmad ², Noor M. Khan ² , Waseem Abbasi ^{3,*} , Muhammad Shahid Anwar ^{4,*} , Sadique Ahmad ⁵ and Mohammed A. El-Affendi ⁵ ¹ Department of Electrical Engineering, Hamdard University, Islamabad 44000, Pakistan² Department of Electrical Engineering, Capital University of Science and Technology, Islamabad 44000, Pakistan³ Department of Electrical Engineering, MY University, Islamabad 46000, Pakistan⁴ Department of AI Software, Gachon University, Seongnam-si 13120, Republic of Korea⁵ EIAS: Data Science and Blockchain Laboratory, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

* Correspondence: waseem.abbasi@myu.edu.pk (W.A.); shahidanwar786@gachon.ac.kr (M.S.A.)

Abstract: The computational complexity of the multiple-input and multiple-output (MIMO) based least square algorithm is very high and it cannot be run on processing-inefficient low-cost platforms. To overcome complexity-related problems, a parallel distributed adaptive signal processing (PDASP) architecture is proposed, which is a distributed framework used to efficiently run the adaptive filtering algorithms having high computational cost. In this paper, a communication load-balancing procedure is introduced to validate the PDASP architecture using low-cost wireless sensor nodes. The PDASP architecture with the implementation of a multiple-input multiple-output (MIMO) based Recursive Least Square (RLS) algorithm is deployed on the processing-inefficient low-cost wireless sensor nodes to validate the performance of the PDASP architecture in terms of computational cost, processing time, and memory utilization. Furthermore, the processing time and memory utilization provided by the PDASP architecture are compared with sequentially operated RLS-based MIMO channel estimator on 2×2 , 3×3 , and 4×4 MIMO communication systems. The measurement results show that the sequentially operated MIMO RLS algorithm based on 3×3 and 4×4 MIMO communication systems is unable to work on a single unit; however, these MIMO systems can efficiently be run on the PDASP architecture with reduced memory utilization and processing time.

Keywords: distributed MIMO channel estimation; low computational complexity; parallel processing**MSC:** 93-05; 94-05

Citation: Raza, H.; Ahmad, I.; Khan, N.M.; Abbasi, W.; Anwar, M.S.; Ahmad, S.; El-Affendi, M.A. Validation of Parallel Distributed Adaptive Signal Processing (PDASP) Framework through Processing-Inefficient Low-Cost Platforms. *Mathematics* **2022**, *10*, 4600. <https://doi.org/10.3390/math10234600>

Academic Editor: Jan Sykora

Received: 19 October 2022

Accepted: 22 November 2022

Published: 5 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Since the beginning of the last decade, technological improvements in hardware and software provide new dimensions for the deployment of signal processing techniques in many communication applications. However, the deployment of these signal processing techniques on low-cost wireless sensor nodes still faces the restrictions in terms of low processing power and limited energy resources. In wireless sensor networks, fast processing as well as low energy consumption are prominent requirements for their efficient utilization; therefore, modern engineering techniques try to provide ingenious alternatives, so that one can improve the algorithm having reasonable trade off between the low energy consumption and communication burden overhead [1,2]. Distributed network-based architecture provides improved performance for many communication applications, such as channel estimation [3], system identification [4,5], source tracking [6], and environmental monitoring [7]. On a distributed adaptive signal processing estimation platform, the processing is

distributed over the network where the network nodes are allowed to exchange information among themselves in such a way that the filter parameters converge in much lesser time. In [8–12], incremental strategies are introduced in which each node in the distributed network communicates only with the adjacent node. In incremental strategies, the data are transmitted to adjacent nodes cyclically throughout the distributed network. In [13–15], diffusion techniques are used to find the unknown filter coefficients for the distributed channel estimation. In these techniques, the parameters of the adaptive algorithm are distributed over the network to improve the estimation of the adaptive algorithm.

Furthermore, in [16], a Distributed Recursive Least Square (D-RLS) adaptive algorithm is presented. In this technique, the parts of the RLS adaptive algorithm are assigned to different nodes in the network and each respective node waits until information is not collected from the previous node. In [17], PDASP architecture is introduced which executes the RLS adaptive algorithm in parallel distributed fashion even with the time-non-aligned indexes over the low-cost wireless sensor nodes. The PDASP architecture provides parallelly lesser computational cost and processing time in each node involved as compared to the above-mentioned techniques [8–16]. However, it has been never been validated through its implementation on low-cost processing-incapable platforms.

As discussed earlier, the PDASP architecture provides much lesser complexity and processing time than sequentially operated adaptive algorithm on a single unit. However, it has not been validated through implementation. In this manuscript, a communication load-balancing procedure is introduced to validate the PDASP architecture using low-cost wireless sensor nodes, namely NANO, UNO, and MEGA [18] having diverse memory utilities. Furthermore, for powerful degree of comparisons, we take MIMO channel estimation with the consideration of Line of Sight (LoS) and diffused components as an example application for our research work which helps to perform an in-depth analysis of the PDASP architecture. Using the communication load-balancing procedure, the obtained measurement results show that the PDASP architecture effectively runs the MIMO RLS algorithm on low-cost sensor nodes. On the other hand, the sequentially operated 3×3 and 4×4 MIMO systems are unable to be run on a single unit. It is also realized that the PDASP architecture using the communication load-balancing procedure provides significant improvement in terms of processing time, computational complexity, and memory utilization as compared to sequentially operated MIMO RLS algorithm. Moreover, the communication burden time is higher than the sequential time of the MIMO RLS algorithm and this time can be reduced by using the transceiver having high data rates as compared to NRF24L01.

The rest of the paper is organized in the following manner. The working procedure of PDASP architecture is described in Section 2. In Section 3, the validation of PDASP architecture for MIMO communication system is presented, the complexity analysis is introduced in Section 4. In Section 5, measurement results are presented and Section 6 draws the conclusions.

2. Parallel Distributed Adaptive Signal Processing (PDASP) Architecture

The input signal vector is divided into N equal-length sub-blocks in a MIMO communication system, each of which is transmitted individually using a different antenna. The block diagram of 2×2 MIMO communication system with one diffused component is shown in Figure 1. At the receiver side, each antenna element receives $(L + 1)N_T$ signals comprising the effect of $h_{tr(l),k}$ channel coefficients from N_T transmitting and L diffused components. The subscript $tr(l)$ shows the information of transmitting antenna, $t = 1, 2, \dots, N_T$ receiving antenna, $r = 1, 2, \dots, N_R$ and $l = 1, \dots, L$ diffused components.

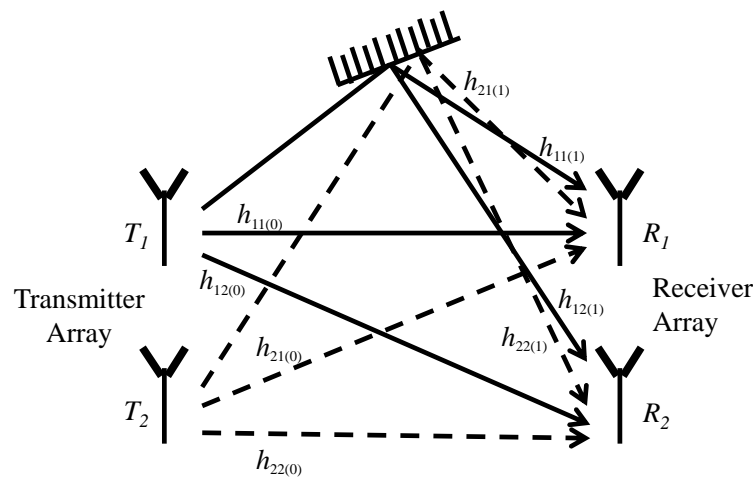


Figure 1. Diffused Component-based 2 × 2 MIMO Communication System.

Therefore, the channel matrix $\tilde{\mathbf{H}}_k$ = for the $N_{\mathcal{T}} \times N_{\mathcal{R}}$ MIMO communication system at time index k with L diffused components [19] can be expressed as

$$\tilde{\mathbf{H}}_k = \begin{bmatrix} h_{11(0),k} & \cdots & h_{11(L),k} & h_{21(0),k} & \cdots & h_{N_{\mathcal{T}}1(L),k} \\ h_{12(0),k} & \cdots & h_{12(L),k} & h_{22(0),k} & \cdots & h_{N_{\mathcal{T}}2(L),k} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ h_{1N_{\mathcal{R}}(0),k} & \cdots & h_{1N_{\mathcal{R}}(L),k} & h_{2N_{\mathcal{R}}(0),k} & \cdots & h_{N_{\mathcal{T}}N_{\mathcal{R}}(L),k} \end{bmatrix} \quad (1)$$

Each entry of channel coefficient, $l = 1, \dots, L$, shows a diffused component which exists between $N_{\mathcal{T}}$ and $N_{\mathcal{R}}$ receiving antenna elements; however, for $l = 0$, the channel coefficient shows parallel interference with the line of sight (LoS) only link. Furthermore, the dispersed components have a significant influence on the dimensions of the channel matrix in time-varying channel environments. Likewise, the dimensions of the filter weight matrix for various MIMO communications systems with LoS and dispersed components are therefore displayed in Table 1.

Table 1. Dimensions of MIMO Channel Matrix for Various Multipath Components.

| Multipath Components | 2 × 2 MIMO | 3 × 3 MIMO | 4 × 4 MIMO |
|----------------------|------------|------------|------------|
| $L = 0$ | 2 × 2 | 3 × 3 | 4 × 4 |
| $L = 1$ | 2 × 4 | 3 × 6 | 4 × 8 |
| $L = 2$ | 2 × 6 | 3 × 9 | 4 × 12 |
| $L = 3$ | 2 × 8 | 3 × 12 | 4 × 16 |

A well-known RLS algorithm [20] has been in use for the applications of channel estimation and channel equalization since its advent. In case of the MIMO communication system, the computational complexity of the MIMO RLS algorithm is not only dependent on MIMO antennas but it also depends on multipath components. The step-by-step computational complexity provided by the RLS algorithm is shown in Table 2, where N shows the order of the MIMO system. In Table 2, it can be seen that the L diffused components in each part of the MIMO RLS algorithm provide critical impact on the computational cost; therefore, the MIMO RLS algorithm cannot be run sequentially on processing-inefficient low-cost platforms. In order to run the MIMO RLS algorithm in parallel fashion, a PDASP architecture with non-aligned time indexes is used. The non-aligned time indexes make run the algorithm parts in parallel fashion. The PDASP architecture with its implementation

on the RLS algorithm is shown in Figure 2; whereas, the selection of four nodes in the PDASP architecture are dependent upon the parts (e.g., Kalman gain, error covariance matrix, etc.) of the RLS algorithm. The RLS method works in parallel with non-aligned time indexes when utilizing the PDASP technique, leading to parallelly low processing times at each processing node. Two considerations must be made in order to run the RLS adaptive filtering method concurrently with various clock systems: first, it should be mentioned that the filter does not behave unstably when used with any application. Second, all of the filter sub-components might be able to operate concurrently. Thus, even with non-aligned time indexes, the sequential structure might be able to operate in parallel.

Table 2. Step-by-Step Computational Complexity of MIMO RLS Algorithm.

| Algorithm Part | Multiplication Complexity | Addition Complexity |
|----------------|---------------------------|---------------------|
| Ψ_k | $2(N + NL)^2$ | $(N + NL)^2$ |
| \mathbf{g}_k | $(N + NL)^2 + NL + 1$ | $(N + NL)^2 - N$ |
| \mathbf{e}_k | $N^2(L + 1)$ | $N^2(L + 1)$ |
| \mathbf{W}_k | $N^2(L + 1)$ | $N^2(L + 1)$ |

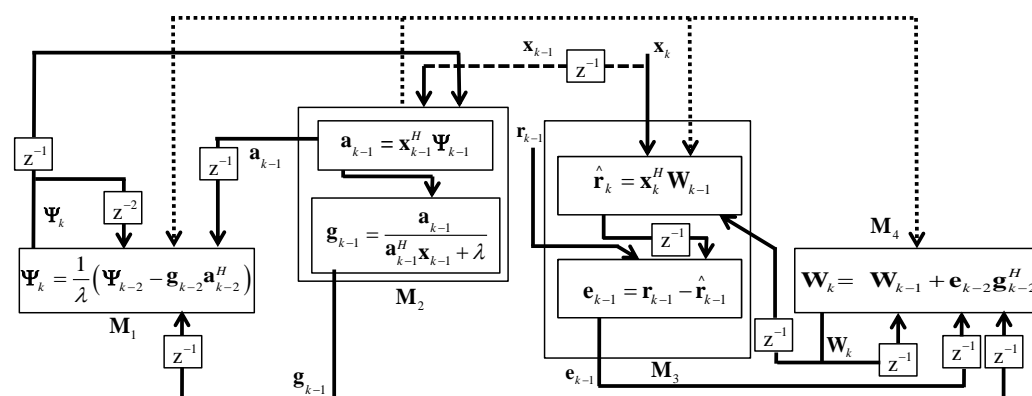


Figure 2. PDASP architecture for low-complexity MIMO channel estimator with non-aligned time indexes [17].

In Figure 2, it can be seen that M_1 , M_2 , M_3 , and M_4 are the four processing nodes that make up the distributed RLS filtering. The adaptive filtering algorithm’s parts (e.g., Kalman gain, error covariance matrix, etc.) influence the selection of the four nodes. The processing nodes M_1 and M_4 are interconnected with M_2 and M_3 , respectively, as well as with each other. Similarly, M_2 is linked to M_1 and M_4 , but M_3 is solely linked to M_4 . All of the processing nodes would share data with one another before figuring out the intended process.

In the PDASP architecture, let the processing time taken by filter weight matrix \mathbf{W}_k , estimation error \mathbf{e}_k , Kalman gain \mathbf{g}_k , and error covariance matrix Ψ_k be T_W , T_e , T_g , and T_Ψ , respectively. Therefore, the time taken by the MIMO-based sequential RLS algorithm, T_{seq} , when it executes in cascade fashion [17], can be written as

$$T_\Psi + T_g + T_e + T_W = T_{seq}. \tag{2}$$

The node M_1 requires more computational cost than the other nodes in the distributed network. The time difference which makes the nodes M_2 , M_3 , and M_4 equivalent to node M_1 can be written as

$$\begin{aligned} \|T_g - T_\Psi\|_1 &= \Delta T_\Psi \\ \|T_{g-T_e}\|_1 &= \Delta T_e \\ \|T_{g-T_W}\|_1 &= \Delta T_W \end{aligned} \tag{3}$$

where $\| \cdot \|_1$ shows the L1 norm operator. The equivalence processing time T_{eq} can be expressed as

$$T_g = T_\Psi + \Delta T_\Psi = T_e + \Delta T_e = T_W + \Delta T_W = T_{eq} \tag{4}$$

In terms of low processing time, the sufficient and strict condition can thus be written as:

$$T_{eq} + T_{f,\Psi} + T_{f,g} + T_{f,e} + T_{f,W} \ll T_{seq} \tag{5}$$

where $T_{f,\Psi}$, $T_{f,g}$, $T_{f,e}$, and $T_{f,W}$ are the fetch times for the transmission of data over the PDASP architecture. The working procedure of PDASP using the MIMO RLS algorithm is shown in Algorithm 1, where all the nodes are capable to share the information among themselves after getting the time equivalent to the maximum processing time of node M_1 .

Algorithm 1: Working procedure of PDASP with Diffused components for $N \times N$ MIMO communication system

Initialize: $\mathbf{W}_{k-1}, \mathbf{a}_{k-2}, \hat{\mathbf{r}}_{k-1}, \Psi_{k-1}, \Psi_{k-2}, \mathbf{e}_{k-2}, \mathbf{g}_{k-2}$
 parallel procedure for M_1, M_2, M_3 and M_4
 for $k=0:N$

Process node M_3
 at time t_1 : $\hat{\mathbf{r}}_k^T = \mathbf{x}_k^T \mathbf{W}_{k-1}$
 at time t_1 : $\mathbf{e}_{k-1} = \mathbf{r}_{k-1} - \hat{\mathbf{r}}_{k-1}$
 at time t_1 : wait ΔT_e

Process node M_2
 at time t_1 : $\mathbf{a}_{k-1} = \mathbf{x}_{k-1}^T \Psi_{k-1}$
 at time t_1 : $\mathbf{g}_{k-1} = \frac{\mathbf{a}_{k-1}}{\mathbf{a}_{k-1}^T \mathbf{x}_{k-1} + \sigma_{v,k}}$
 at time t_1 : wait ΔT_g

Process node M_1
 at time t_1 : $\Psi_k = \Psi_{k-2} - \mathbf{g}_{k-2} \mathbf{a}_{k-2}^T$

Process node M_4
 at time t_1 : $\mathbf{W}_k = \mathbf{W}_{k-1} + \mathbf{e}_{k-2} \mathbf{g}_{k-2}^T$
 at time t_1 : wait ΔT_W

at time t_2 : Transmit \mathbf{g}_{k-1} from M_2 to M_1 and M_4
 at time t_3 : Transmit Ψ_{k-1} from M_1 to M_2 , \mathbf{W}_{k-1} from M_4 to M_3
 at time t_4 : Transmit \mathbf{e}_{k-2} from M_3 to M_4 , \mathbf{a}_{k-2} from M_2 to M_1
 end for

3. Communication Load-Balancing Procedure

In this section, the procedure of information interchange over the PDASP architecture is presented. The PDASP architecture consists of four nodes, M_1, M_2, M_3 , and M_4 . The block diagrams of the sequentially and parallelly working MIMO RLS algorithm are shown in Figure 3a,b, respectively.

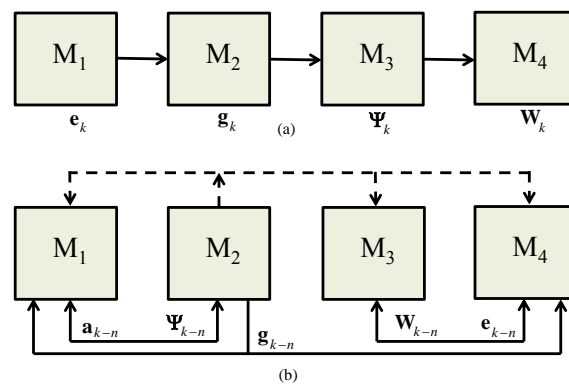


Figure 3. Working procedure of low complexity MIMO Algorithm (a). Sequential procedure (b). Parallel working procedure

Before running the expensive procedure parallelly, M_2 sends a beacon message to all the slave nodes which makes them ready for working on a desired goal. After that, all the four nodes are capable to work on the desired process. The working procedure of information interchange over PDASP architecture is clearly depicted in Figure 4. In this architecture, first of all, g_k is transmitted from M_2 towards M_4 and M_1 then M_1 and M_4 become capable to forward information of W_k and Ψ_k towards M_3 and M_2 , respectively. Likewise, after getting information of Ψ_k and W_k , M_2 and M_3 become capable to share information regarding e_k and a_k towards M_4 and M_1 , respectively. In this way, the information interchange is totally balanced over the distributed network.

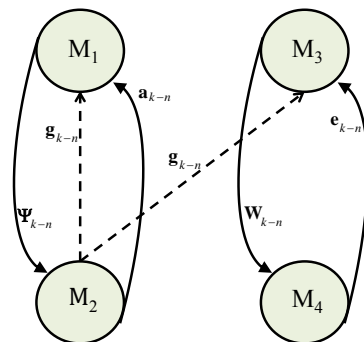


Figure 4. Working procedure for the transmission of data over PDASP architecture.

4. Complexity Analysis

In this section, the complexity of the sequentially operated and distributed adaptive RLS filtering for various MIMO communication systems is discussed. The computational complexity provided by the sequentially operated MIMO RLS algorithm entails $3(N + NL)^2 + 2N^2(L + 1) + NL + 1$ multiplications and $2N^2(L + 1) + 2(N + NL)^2 - N$ additions per iteration.

On the other hand, in the PDASP architecture, the node M_1 provides increased computational complexity as compared to the entire nodes in the PDASP architecture; therefore, the node M_1 entails $2(N + NL)^2$ multiplications and $(N + NL)^2$ additions at maximum. Likewise, the multiplication and addition complexity with LoS and diffused components for various MIMO systems are shown in Figure 5 and Figure 6, respectively. It can be observed that the maximum computational cost provided by the PDASP architecture is much lesser than that of the sequentially operated MIMO RLS algorithm. Furthermore, the percentage improvement in computational complexity provided by the PDASP architecture is shown in Table 3. It can be visualized that the percentage improvement of more than 50% in sense of parallelly decreased computational cost shows a superlative enhancement in computational cost of the algorithm as compared to the sequentially operated MIMO RLS algorithm.

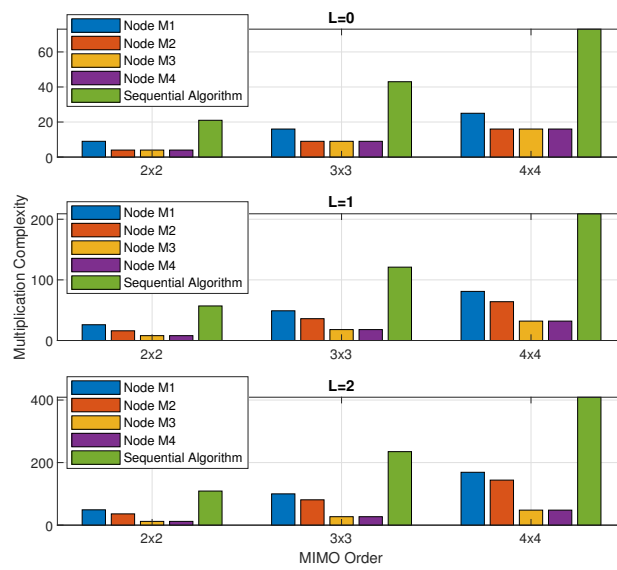


Figure 5. Sequential and Distributed Multiplication Complexity for Various MIMO Systems with LoS and Diffused Components.

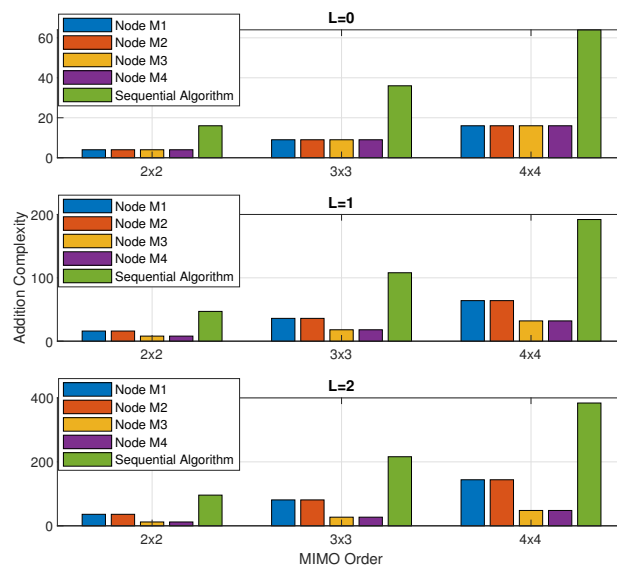


Figure 6. Sequential and Distributed Addition Complexity for Various MIMO Systems with LoS and Diffused Components.

Table 3. Percentage Improvement in Multiplication and Addition Complexity for Various MIMO Systems.

| Complexity | Diffused Components | 2 × 2 MIMO | 3 × 3 MIMO | 4 × 4 MIMO |
|---------------------------|---------------------|------------|------------|------------|
| Multiplication Complexity | $L = 0$ | 57.14% | 62.79% | 65.75% |
| Multiplication Complexity | $L = 1$ | 56.14% | 59.50% | 61.24% |
| Multiplication Complexity | $L = 2$ | 55.04% | 57.44% | 58.70% |
| Addition Complexity | $L = 0$ | 75.00% | 75.00% | 75.00% |
| Addition Complexity | $L = 1$ | 66.66% | 66.66% | 66.66% |
| Addition Complexity | $L = 2$ | 62.50% | 62.50% | 62.50% |

5. Measurement Results

In test bed setup, four low-cost wireless sensor nodes from the Arduino platform are used to implement the PDASP architecture. The measurement results are presented by considering various MIMO communication systems with LoS and diffused components. The low-cost wireless sensor nodes, namely NANO, UNO, and MEGA [18], are used to validate the performance of PDASP architecture. The distance among the four sensor nodes is set approximately 24 cm from each apart. All the sensor categories (NANO, UNO, and MEGA) have the same processing speed of 16 MH with diverse static random memory specifications of 1, 2, and 8 KBytes, respectively. The communication is being possible among the network nodes by using NORDIC radio NRF24L01 Module [21].

The maximum transmission speed provided by this Module is 2 Mbps at power of 20 dB. The MIMO RLS algorithm is deployed on low-cost wireless sensor nodes to substantiate the validation of the PDASP technique in terms of processing time, computational complexity, and communication burden. The processing time with respect to sequential and PDASP technique relative to each processing node is shown in Figure 7.

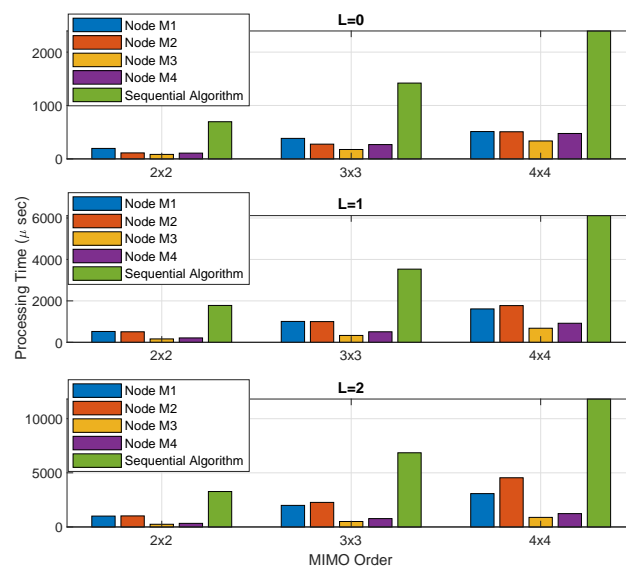


Figure 7. Sequential and Distributed Processing Time in µsec for Various MIMO Systems with LoS and Diffused Components.

It is realized that the processing time obtained by using the PDASP architecture utilizes parallelly less execution time than the sequential RLS filtering algorithm. Furthermore, Table 4 displays the reduction in processing time utilizing the PDASP architecture. It can be realized that the PDASP provides significant improvement in decreased processing time parallelly than that of sequential RLS algorithm which makes a critical impact on the efficiency of the processing device as well as on the low power consumption.

Table 4. Percentage Improvement in Processing Time for Various MIMO communication Systems Using PDASP architecture.

| Diffused Components | 2 × 2 MIMO | 3 × 3 MIMO | 4 × 4 MIMO |
|---------------------|------------|------------|------------|
| $L = 0$ | 71.84% | 75.49% | 78.63% |
| $L = 1$ | 70.56% | 71.46% | 71.02% |
| $L = 2$ | 68.82% | 66.90% | 61.54% |

Furthermore, the comparison among the working of low-cost devices and their memory limitations for sequential and distributed MIMO systems with multipath components is shown in Tables 5 and 6, respectively. It is observed that the distributive strategy effectively runs the sequential algorithm for various MIMO systems with diffused components without showing any complication of memory error. However, among of all three sensor nodes, NANO is still unable to work for the 3×3 and 4×4 MIMO system with two diffused components because of lesser memory specification. Moreover, the percentage improvement in memory utilization by using the PDASP architecture is shown in Table 7. It can be observed that the PDASP architecture while running the sequential algorithm parallelly on low-cost sensor nodes provides lesser memory utilization in the sense of fetching information from the memory which entails proficient impact on the efficiency of the nodes. Furthermore, the time taken by the PDASP architecture for one iteration with a communication burden is presented in Table 8. It can be realized that the communication time is higher than the MIMO RLS-based sequential processing time, and this communication time can be reduced by using the transceiver with high data rates as compared to NORDIC radio NRF24L01.

Table 5. Sequential Memory Limitation Comparison on Various MIMO Systems with Multipath Components.

| Aurdino Platform | Multipath Components | 2×2 MIMO | 3×3 MIMO | 4×4 MIMO |
|------------------|----------------------|-------------------|-------------------|-------------------|
| NANO | $L = 0$ | Working | Working | Working |
| UNO | $L = 0$ | Working | Working | Working |
| MEGA | $L = 0$ | Working | Working | Working |
| NANO | $L = 1$ | Working | Memory Error | Memory Error |
| UNO | $L = 1$ | Working | Working | Memory Error |
| MEGA | $L = 1$ | Working | Working | Working |
| NANO | $L = 2$ | Memory Error | Memory Error | Memory Error |
| UNO | $L = 2$ | Working | Memory Error | Memory Error |
| MEGA | $L = 2$ | Working | Working | Working |

Table 6. Distributed Memory Limitation Comparison for Various MIMO Systems with Multipath Components.

| Aurdino Platform | Multipath Components | 2×2 MIMO | 3×3 MIMO | 4×4 MIMO |
|------------------|----------------------|-------------------|-------------------|-------------------|
| NANO | $L = 0$ | Working | Working | Working |
| UNO | $L = 0$ | Working | Working | Working |
| MEGA | $L = 0$ | working | Working | Working |
| NANO | $L = 1$ | Working | Working | Working |
| UNO | $L = 1$ | Working | Working | Working |
| MEGA | $L = 1$ | Working | Working | Working |
| NANO | $L = 2$ | Working | Memory Error | Memory Error |
| UNO | $L = 2$ | Working | Working | Working |
| MEGA | $L = 2$ | Working | Working | Working |

Table 7. Percentage Improvement in Memory Utilization Using PDASP Architecture.

| Processing Time | Diffused Components | 2 × 2 MIMO | 3 × 3 MIMO | 4 × 4 MIMO |
|------------------------------------|---------------------|------------|------------|------------|
| Sequential Algorithm | $L = 0$ | 696 | 1420 | 2396 |
| Combined Time ($M1, S1, S2, S3$) | $L = 0$ | 500 | 1068 | 1832 |
| Percentage Improvement | $L = 0$ | 28.16% | 24.78% | 23.53% |
| Sequential Algorithm | $L = 1$ | 1780 | 3532 | 6116 |
| Combined Time ($M1, S1, S2, S3$) | $L = 1$ | 1408 | 2848 | 4984 |
| Percentage Improvement | $L = 1$ | 20.89% | 19.36% | 18.50% |
| Sequential Algorithm | $L = 2$ | 3272 | 6852 | 11816 |
| Combined Time ($M1, S1, S2, S3$) | $L = 2$ | 2604 | 5536 | 9740 |
| Percentage Improvement | $L = 2$ | 20.41% | 19.20% | 17.56% |

Table 8. Maximum Time Taken for One Complete Iteration Using PDASP Architecture for Various MIMO Systems.

| Diffused Components | 2 × 2 MIMO | 3 × 3 MIMO | 4 × 4 MIMO |
|---------------------|----------------|----------------|----------------|
| $L = 0$ | 9520 μ s | 11,449 μ s | 1533 μ s |
| $L = 1$ | 16,934 μ s | 26,320 μ s | 38,986 μ s |
| $L = 2$ | 27,143 μ s | 43,380 μ s | 66,898 μ s |

6. Conclusions

In this paper, a communication load-balancing procedure is introduced to validate the PDASP architecture for various MIMO streams. The PDASP architecture has been validated in terms of processing time, computational complexity, memory utilization, and communication delay. Processing time and computational complexity using the PDASP architecture both have been compared with the sequentially operated MIMO-based RLS algorithm. It has been realized that the PDASP architecture utilizes less processing time and computational cost than the sequential MIMO-based RLS algorithm. Moreover, the memory utilization in sense of data storing and fetching in memory has also been improved when compared with the sequentially operated MIMO-based RLS algorithm. Furthermore, it has been realized that the communication burden for one complete iteration using the PDASP architecture is higher than the sequential time of the MIMO RLS algorithm and this communication time can be reduced by using a transceiver with higher data rates as compared to the existing NORDIC radio NRF24L01 Module.

Author Contributions: Conceptualization, H.R. and I.A.; methodology, H.R.; validation, H.R., I.A. and N.M.K.; formal analysis, W.A., M.S.A. and M.A.E.-A.; investigation, H.R.; resources, I.A.; data curation, M.S.A.; writing—original draft preparation, H.R., S.A. and N.M.K.; writing—review and editing, W.A., S.A. and M.S.A.; supervision, N.M.K.; funding acquisition, M.S.A. and M.A.E.-A.; Simulations, H.R., S.A. and M.A.E.-A. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of Prince Sultan University for Article Processing Charges (APC) of this publication.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Acknowledgments: The authors would like to acknowledge Prince Sultan University and the EIAS: Data Science and Blockchain Laboratory for their valuable support.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Li, X.; Zhao, L.; Yu, K.; Aloqaily, M.; Jararweh, Y. A cooperative resource allocation model for IoT applications in mobile edge computing. *Comput. Commun.* **2021**, *173*, 183–191. [[CrossRef](#)]
2. Shandil, N.; Dutt, S.; Agrawal, S. Handling the Energy-Delay Trade-off in Wireless Sensor Networks: State-of-the-Art. In *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*; Springer: Singapore, 2019; pp. 979–986.
3. Sayed, A.H.; Tu, S.Y.; Chen, J.; Zhao, X.; Towfic, Z.J. Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior. *IEEE Signal Process. Mag.* **2013**, *30*, 155–171. [[CrossRef](#)]
4. Shin, J.; Park, B.Y.; Lee, W.I.; Yoo, J. Variable Matrix-Type Step-Size Affine Projection Sign Algorithm for System Identification in the Presence of Impulsive Noise. *Symmetry* **2022**, *14*, 1985. [[CrossRef](#)]
5. Xia, H.; Chen, F. Filtering-based parameter identification methods for multivariable stochastic systems. *Mathematics* **2020**, *8*, 2254. [[CrossRef](#)]
6. Li, D.; Wong, K.D.; Hu, Y.H.; Sayeed, A.M. Detection, classification, and tracking of targets. *IEEE Signal Process. Mag.* **2002**, *19*, 17–29.
7. Estrin, D.; Girod, L.; Pottie, G.; Srivastava, M. Instrumenting the world with wireless sensor networks. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, UT, USA, 7–11 May 2001.
8. Sayed, A.H.; Lopes, C.G. Distributed recursive least-squares strategies over adaptive networks. In Proceedings of the IEEE Fortieth Asilomar conference on Signals, Systems and Computers (ACSSC), Pacific Grove, CA, USA, 29 October–1 November 2006; pp. 233–237.
9. Jian, J.; Lijuan, J.; Ran, T.; Yue, W. Distributed incremental bias-compensated rls estimation over multi-agent networks. *Sci. China Inf. Sci.* **2017**, *60*, 1–15.
10. Li, L.; Chambers, J.A.; Lopes, C.G.; Sayed, A.H. Distributed estimation over an adaptive incremental network based on the affine projection algorithm. *IEEE Trans. Signal Process.* **2010**, *58*, 151–164. [[CrossRef](#)]
11. Jalali, S.; Nuzman, C.; Sun, Y. Incremental maximum likelihood estimation for efficient adaptive filtering. *arXiv* **2022**, arXiv:2209.01594.
12. Mostafapour, E.; Ghobadi, C.; Nourinia, J.; Amirani, M.C. Tracking performance of incremental LMS algorithm over adaptive distributed sensor networks. *arXiv* **2021**, arXiv:2103.11903
13. Huang, F.; Zhang, J.; Zhang, S.; Chen, H.; Poor, H.V. Diffusion Bayesian Subband Adaptive Filters for Distributed Estimation Over Sensor Networks. *IEEE Trans. Commun.* **2021**, *69*, 6909–6925. [[CrossRef](#)]
14. Xu, W.; Zhao, H. Diffusion Constrained Least Mean M-estimate Algorithm for Adaptive Networks. In Proceedings of the 6th International Conference on Digital Signal Processing, Chengdu, China, 25–27 February 2022; pp. 183–186.
15. Zheng, C.; Jia, L.; Yang, Z.J.; Wang, Y. Diffusion Bias-Compensation RLS Estimation Over Noisy Node-Specific Networks. *Circuits Syst. Signal Process.* **2021**, *40*, 2564–2583. [[CrossRef](#)]
16. Mateos, G.; Schizas, I.D.; Giannakis, G.B. Distributed recursive least squares for consensus-based in-network adaptive estimation. *IEEE Trans. Signal Process.* **2009**, *57*, 4583–4588. [[CrossRef](#)]
17. Khan, N.M.; Hasan, R. Processing-Efficient Distributed Adaptive RLS Filtering for Computationally Constrained Platforms. *Wirel. Commun. Mob. Comput.* **2017**, *2017*, 1248796. [[CrossRef](#)]
18. Louis, L. Working principle of Arduino and using IT as a tool for study and research. *Int. J. Control. Autom. Commun. Syst. (IJCACS)* **2016**, *1*, 21–29. [[CrossRef](#)]
19. Raza, H.; Khan, N.M. Low Complexity Linear Channel Estimation for MIMO Communication Systems. *Wirel. Pers. Commun.* **2017**, *97*, 5031–5044. [[CrossRef](#)]
20. Haykin, S. *Adaptive Filter Theory*, 4th ed.; Prentice-Hall: Englewood Cliffs, NJ, USA, 2002.
21. Sonavane, S.S.; Patil, B.P.; Kumar, V. Experimentation for packet loss on MSP430 and nRF24L01 based wireless sensor network. *Int. J. Adv. Netw. Appl.* **2017**, *8*, 25–29.