

Research Article

Processing-Efficient Distributed Adaptive RLS Filtering for Computationally Constrained Platforms

Noor M. Khan and Hasan Raza

Department of Electrical Engineering, Capital University of Science and Technology, Islamabad 44000, Pakistan

Correspondence should be addressed to Hasan Raza; hasan@arwic.com

Received 7 December 2016; Revised 14 February 2017; Accepted 20 February 2017; Published 31 May 2017

Academic Editor: Lei Zhang

Copyright © 2017 Noor M. Khan and Hasan Raza. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a novel processing-efficient architecture of a group of inexpensive and computationally incapable small platforms is proposed for a parallelly distributed adaptive signal processing (PDASP) operation. The proposed architecture runs computationally expensive procedures like complex adaptive recursive least square (RLS) algorithm cooperatively. The proposed PDASP architecture operates properly even if perfect time alignment among the participating platforms is not available. An RLS algorithm with the application of MIMO channel estimation is deployed on the proposed architecture. Complexity and processing time of the PDASP scheme with MIMO RLS algorithm are compared with sequentially operated MIMO RLS algorithm and linear Kalman filter. It is observed that PDASP scheme exhibits much lesser computational complexity parallelly than the sequential MIMO RLS algorithm as well as Kalman filter. Moreover, the proposed architecture provides an improvement of 95.83% and 82.29% decreased processing time parallelly compared to the sequentially operated Kalman filter and MIMO RLS algorithm for low doppler rate, respectively. Likewise, for high doppler rate, the proposed architecture entails an improvement of 94.12% and 77.28% decreased processing time compared to the Kalman and RLS algorithms, respectively.

1. Introduction

Adaptive filtering techniques play a very important role in the emerging fields of science and technology [1–3]; however, the last two decades have witnessed tremendous research in the field of adaptive filtering for the improvement of their convergence and complexity requirements [4–6]. However, achieving fast convergence on an energy-constrained and computationally incapable platform still remains a dream in spite of magnificent advancements in Integrated Circuit (IC) technologies. For instance, in video conferencing, the echo cancellation requires a high definition adaptive filtering algorithm to avail a robust convergence performance while tracking the time varying uncertainties present in the communication link. Nevertheless, such high definition adaptive algorithm cannot be run on an energy-constrained and computationally incapable inexpensive platform. The following lines present a brief review of the literature where significant efforts have been made to propose a low complexity and distributed solution for this problem.

In [7, 8], Banachiewicz inversion formulation is used to perform the matrix inversion for MIMO-OFDM based software defined radio (SDR) signal detection. The inversion of a 4×4 matrix is divided into four 2×2 matrices that reduce the computational operations. Likewise, the authors in [9] derive a low complexity algorithm for Hermitian positive-definite recursive matrix inversion that provides low computational complexity compared to [7, 8] with the utilization of 52 operations for finding matrix inversion only. Therefore, using the concept of matrix inversion of [7–9], it does not exhibit a significant impact on the computational cost of high definition adaptive filtering [10]. In [11], Xiao et al. introduce an LR-MMSE algorithm based on QR decomposition and complex lattice reduction (CLR) which provides 35.5% lesser computational complexity than MMSE based scheme [12]; however, 35.5% lesser computational cost still can not meet the demands of high data rate communications. A low complexity in reduced rank linear interference suppression is proposed in [13, 14] which is based on the use of polynomial expansion (PE). The matrix inverse is represented by an L th

order matrix polynomial [13, 15–17] and its selection is one of the tedious tasks while trading off between the complexity and detection performance. In [18], a comparison is made on renowned subband adaptive filtering (SAF) structures with parallel arrangement of multirate filter banks. The SAF technique exhibits reduced complexity through the use of least mean square (LMS) adaptive filtering algorithm in acoustic noise environment. Therefore, due to phase, aliasing, and amplitude distortions and extra processing delay, these systems may be ruled out for real-time implementation. Another architecture configuration for reducing runtime is MMSE signal estimation using wireless sensor nodes [19]. In this architecture, authors use the distributed adaptive node-specific signal estimation (DANSE) technique to estimate the channel coefficients by following Wiener Hopf equation. However, DANSE technique only follows the MMSE criterion rather than running the adaptive filtering algorithm. This makes DANSE incapable of estimating time varying channel conditions. To the best of our knowledge, there is no parallel structure of recursive adaptive filtering algorithms in the literature where any of the complex adaptive algorithms runs in parallel fashion over computationally incapable platforms with no perfect time alignment. Nevertheless, software parallelism is provided by Matlab[®] [20] and Labview[®] [21] which are available in various architectures of the present fast processing computers with perfect time-alignment processors. These software programs provide parallel processing toolbox to divide the large problems into smaller computations, hence requiring reduced running time. Likewise, graphical processing unit (GPU) enables running high definition graphics on a personal computer (PC) by exploiting hundreds of cores [22]. Furthermore, compute unified device architecture (CUDA) [23] is NVIDIA's GPU architecture which provides multithreaded applications where cores can communicate and exchange information with each other. However, these cores have not been used to run adaptive algorithms in parallel with nonaligned time indexes.

In this paper, our objective is to provide a novel low complexity solution for inexpensive and computationally incapable platforms through proposing their parallelly distributed adaptive signal processing (PDASP) operation making them run computationally expensive procedures cooperatively.

The implementation of the proposed PDASP technique using recursive least square (RLS) makes the inexpensive and computationally incapable platforms work in parallel even with nonaligned time indexes by providing much lesser processing time parallelly than the sequential Kalman [24] and RLS [25, 26] algorithms, whereas RLS is the special case of Kalman filter and is one of the most popular filters in the adaptive filtering domain that offers a superior convergence rate, especially for time varying environments with the price of an increase in nonlinear computational cost.

The idea behind the distributed signal processing is based on parallel processing and “*divide and conquer*” prototype. In this prototype, the part of the algorithm is divided into subalgorithms which are then passed to other processing nodes to provide an efficient low complexity solution.

The rest of the paper is organized in the following manner. Section 2 describes the system model. Section 3 presents

the proposed PDASP scheme for computationally incapable platforms. Complexity analysis is introduced in Section 4. In Section 5, simulation based results are presented and Section 6 draws the conclusions.

2. System Model

In this section, we discuss the working procedure of our proposed parallelly operated recursive least square (RLS) filter in the light of its conventional sequential operation.

In conventional (RLS) adaptive algorithm and its variants, all filter subparts are interdependent on each other and operate sequentially. Before introducing the proposed parallel RLS operation over individual platforms with different clock systems, we define some timing variables with illustration shown in Figure 1, where a single iteration of an RLS algorithm consists of N sequential blocks.

(i) *Computational Time T_c* . This is the time taken by the processor for a single computation. It can be calculated simply by the speed of the processor.

(ii) *Block Processing Time T_b* . It is the processing time of a block of the algorithm. It depends on the number of computations involved in a block. It can thus be a multiple of T_c .

(iii) *Fetch Time T_f* . This is the time in which one block fetches information from another block, usually its predecessor.

(iv) *Algorithm Step Time T_s* . It is the processing time of a complete iteration of the algorithm.

If RLS filtering is operated on a single computationally capable platform, all algorithm blocks would be executed sequentially as shown in Figure 1 with fetch time $T_f \rightarrow 0$. However, if the same RLS filtering is operated on a group of computationally incapable platforms using the proposed PDASP architecture, different algorithm blocks would be executed parallelly on various individual platforms with varying fetch times depending upon the media among the nodes as shown in Figure 2.

The only possible way to operate the RLS algorithm in parallel fashion on individual platforms with different clock systems is by putting the time as nonaligned. While setting the time nonalignment, two things must be taken into account. First, it must be realized that the filter is not showing any uncertain behavior though implementing on any application. Secondly, all the filter subparts are able to work in parallel manner with favorable fetch times with respect to block processing times. In this way, the sequential structure may be able to work parallelly even with nonaligned time indexes. In Figure 2, the cooperative parallelly operated RLS filtering architecture consists of four processing nodes, namely, M_1 , M_2 , M_3 , and M_4 . The processing nodes M_1 and M_4 are interlinked with M_2 and M_3 , respectively, while being connected to themselves also. Likewise, M_2 is interconnected with M_1 and M_4 and M_3 is only linked with M_4 . All the processing nodes would first share information with one another and then would work out the desired process. The processing

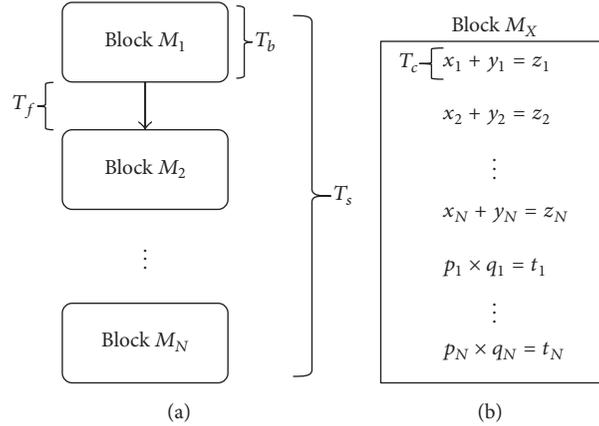


FIGURE 1: Sequential working of conventional RLS algorithm. (a) Sequential working of individual blocks. (b) Processes involved in a single block.

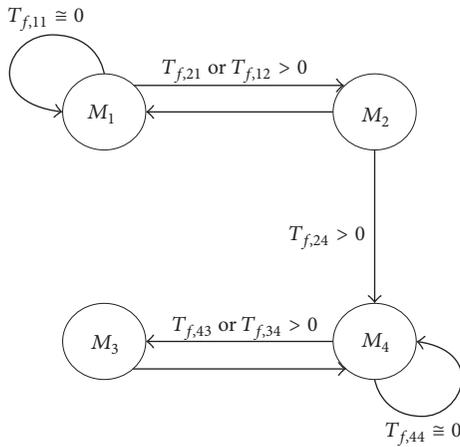


FIGURE 2: Proposed cooperative parallelly operated RLS filtering.

time of each block differs from one another and is known to all nodes; therefore, all processing nodes which complete their processing tasks earlier than others wait the processing time equivalent to the block of maximum processing time till the processing of the block with maximum processing time ends. In this way, the inexpensive and computationally incapable platforms work in parallel for a combined goal.

3. Proposed PDASP Technique for RLS with Nonaligned Time Indexes

In adaptive filtering, all the filter subparts are interdependent on one another. Due to cascaded fashion the algorithm takes mutual processing time while attaining its convergence with respect to uncertain channel conditions. By using the PDASP technique, RLS algorithm runs in parallel manner even with nonaligned time indexes while providing parallelly low processing time at each machine or processing node. The flow diagram of PDASP technique using RLS is shown in Figure 3. The notation " T_X " is used to represent the time used in processing of X whose computation is done

inside the pointed block. Let the processing times taken by error covariance matrix " Φ_k ," Kalman gain " \mathbf{g}_k ," received signal estimation \mathbf{r}_k , estimation error " \mathbf{e}_k ," and update filter coefficient matrix " $\widehat{\mathbf{H}}_k$ " be T_Φ , T_g , T_r , T_e , and $T_{\widehat{\mathbf{H}}}$, respectively.

Therefore, the total time taken by the whole algorithm that runs in cascaded fashion is

$$T_\Phi + T_g + T_r + T_e + T_{\widehat{\mathbf{H}}} = T_{\text{tot}}. \quad (1)$$

The maximum processing time among $T_\Phi \cdots T_{\widehat{\mathbf{H}}}$ is T_Φ because Φ_k takes more multiplications than any of \mathbf{g}_k , \mathbf{r}_k , \mathbf{e}_k , and $\widehat{\mathbf{H}}_k$, in order to operate RLS algorithm parallelly while distributing the operation of various blocks on individual nodes with nonaligned time indexes. The strict and sufficient conditions with respect to fast convergence rate in terms of multiplication and addition computations can thus be written as

$$T_g, T_r, T_e, T_{\widehat{\mathbf{H}}} \leq T_\Phi, \quad (2)$$

$$T_\Phi + T_f \ll T_{\text{tot}}.$$

Due to nonaligned time indexes, the mismatch ζ between the aligned and nonaligned time indexes can be written as

$$\zeta = |e_{\text{seq}} - e_{\text{NA}}|, \quad (3)$$

where e_{seq} is the error of sequential algorithm and e_{NA} is the error of the PDASP algorithm with nonaligned time indexes. The proposed architecture can be run in sequential format for convergence calibration. The sequential implementation of PDASP RLS algorithm with nonaligned time indexes is nearly the same as that of a conventional RLS algorithm run on a single machine. Steps of this sequential format are shown in Algorithm 1.

4. Complexity Analysis

The complexity of the linear Kalman filter requires $2N^3 + 6N^2 + 3N + 1$ multiplications and $3N^3 + 4N^2 + 2$ additions per iteration, where N represents the dimension of the filter

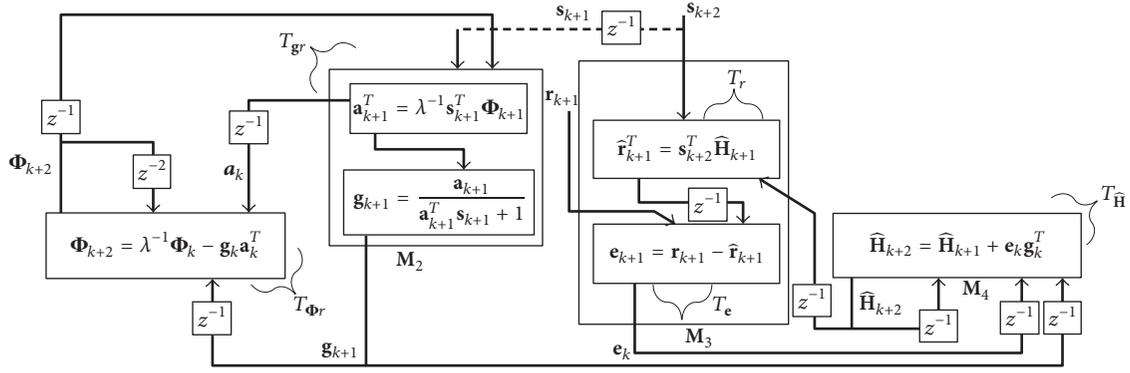
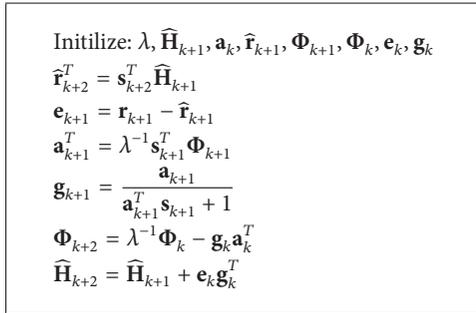


FIGURE 3: Proposed PDASP architecture for MIMO RLS algorithm with nonaligned time indexes.



ALGORITHM 1: PDASP RLS algorithm with nonaligned time indexes if it runs sequentially.

order. Likewise, RLS algorithm that is the special case of Kalman filter entails $5N^2 + 2N + 1$ multiplications and $4N^2 + 2$ additions per iteration. The implementation of the proposed PDASP technique on RLS algorithm exhibits much lesser computational cost for each parallelly distributed entity block. The proposed PDASP technique with nonaligned time indexes entails parallelly $2N^2$ multiplications and N^2 additions per iteration at maximum. The proposed parallel technique thus provides much lesser processing time than that of sequential Kalman and RLS algorithms.

5. Simulation Results

In this section, Monte Carlo simulations with binary phase shift keying (BPSK) are performed on 4×4 MIMO communication system to substantiate the validation of our proposed PDASP architecture. The forgetting factor λ is set to be 0.98 for both the proposed PDASP and sequential RLS algorithms.

The proposed parallel technique that is implemented on MIMO RLS is then compared with the sequential MIMO RLS adaptive algorithm and Kalman filter in terms of computational complexity, mean square error (MSE), and processing time with nonaligned time indexes. The implementation of the proposed PDASP scheme is done using MIMO RLS algorithm and its performance in terms of computational complexity is then compared with sequentially operated nondistributed Kalman and RLS adaptive filtering algorithms. The parallel technique provides much

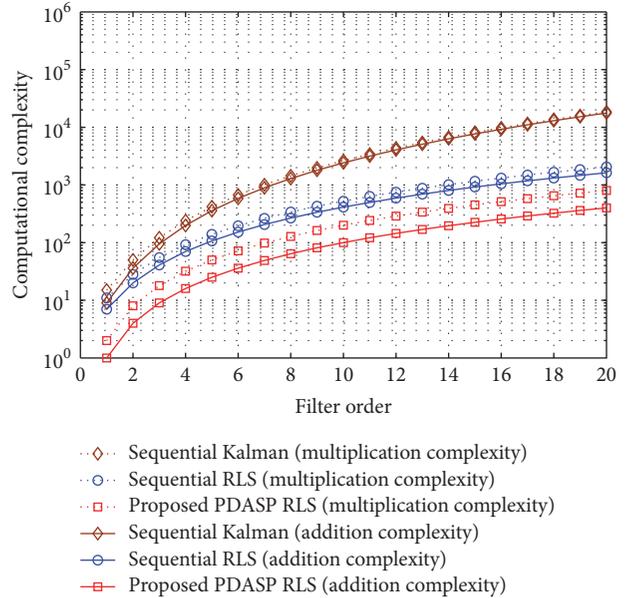


FIGURE 4: Computational complexity comparison among 4×4 MIMO sequential algorithms and proposed PDASP technique.

lesser computational complexity parallelly than the sequential Kalman and MIMO RLS algorithms. Figure 4 represents the multiplication and addition complexity comparison of proposed PDASP technique and those of sequentially operated Kalman and MIMO RLS algorithms. It is observed that the proposed PDASP technique using nonaligned time indexes provides parallelly much lesser multiplication and addition complexity than sequential Kalman and MIMO RLS algorithms. Figures 5 and 6 show the MSE performance at low doppler rate $f_D T = 10^{-6}$ and high doppler rate $f_D T = 10^{-3}$, respectively, and Figure 7 shows their MSE difference among the proposed PDASP MIMO RLS and sequentially operated algorithms. It is realized that the difference in convergence performance of proposed PDASP scheme run with nonaligned time indexes and that of the sequential Kalman and MIMO RLS algorithms is only of 20 and 10 iterations, respectively, at low doppler spread and about 30 and 20 iterations with relatively high doppler spread,

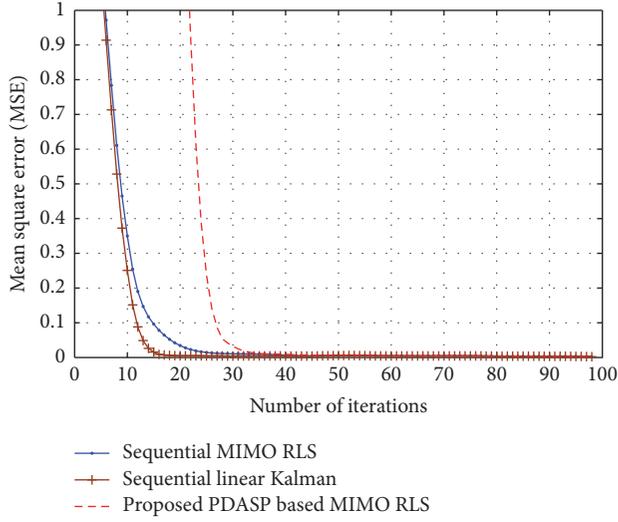


FIGURE 5: Mean Square Error (MSE) tracking performance versus training length for 4×4 MIMO when $f_D T = 10^{-6}$.

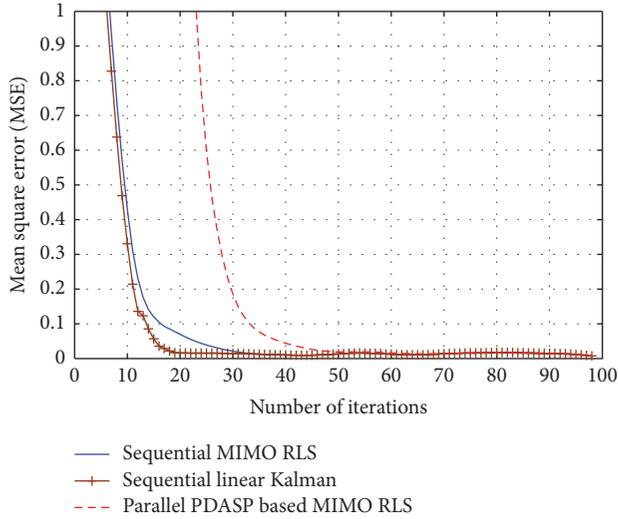


FIGURE 6: Mean Square Error (MSE) tracking performance versus training length for 4×4 MIMO when $f_D T = 10^{-3}$.

respectively. Considering the difference in Figure 7, it can be seen that, due to initialization of the algorithm parameters of PDASP technique, the error difference that is small at the start gradually increases and then reverses to decrease and eventually becomes zero on a specific number of iterations.

The fast Ethernet speed of 125 Mbits/s is taken as the reference peak bit rate in the wired communication. In 4×4 MIMO PDASP scheme, the maximum size of 4×4 matrix is to be transmitted from one machine node to another through wired communication. However, each entry in 4×4 MIMO matrix consists of 4 bytes, in which two significant bytes are before the decimal point and two significant bytes are taken after the decimal point. The total number of bits for 4×4 matrix are $4 \times 16 \times 8 = 512$ bits. The fetch time T_f according to this number of bits is $41 \mu\text{s}$. Therefore, the processing time

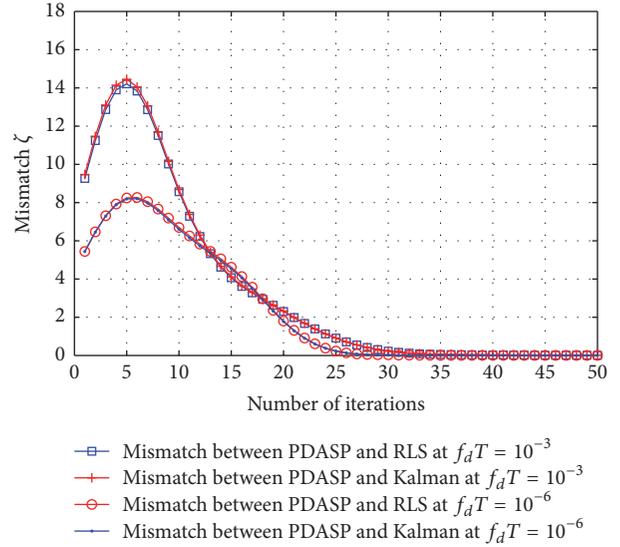


FIGURE 7: Mean Square Error difference among sequential algorithms and proposed PDASP scheme.

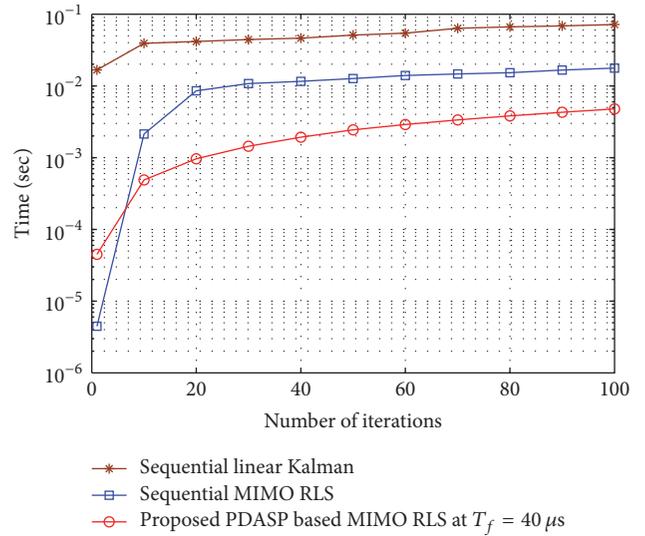


FIGURE 8: Processing time comparison among sequential algorithms and proposed PDASP technique.

comparison and the processing time difference at $T_f = 41 \mu\text{s}$ among the sequential algorithms and the proposed PDASP technique are presented in Figures 8 and 9, respectively. It is clear that the proposed PDASP technique provides much lesser processing time than the sequentially operated Kalman filter and MIMO RLS algorithm. The percentage improvement in decreased processing time is shown in Tables 1 and 2. At low doppler rate, it is realized that the proposed PDASP MIMO RLS algorithm converges about 35 iterations with the addition of $T_f = 41 \mu\text{s}$ at each iteration but still utilizes 95.83% and 82.29% lesser processing time than the sequential Kalman and MIMO RLS algorithms, respectively. Likewise, for high doppler rate, the proposed PDASP MIMO RLS takes 50 iterations for its convergence with the increase of 30 and

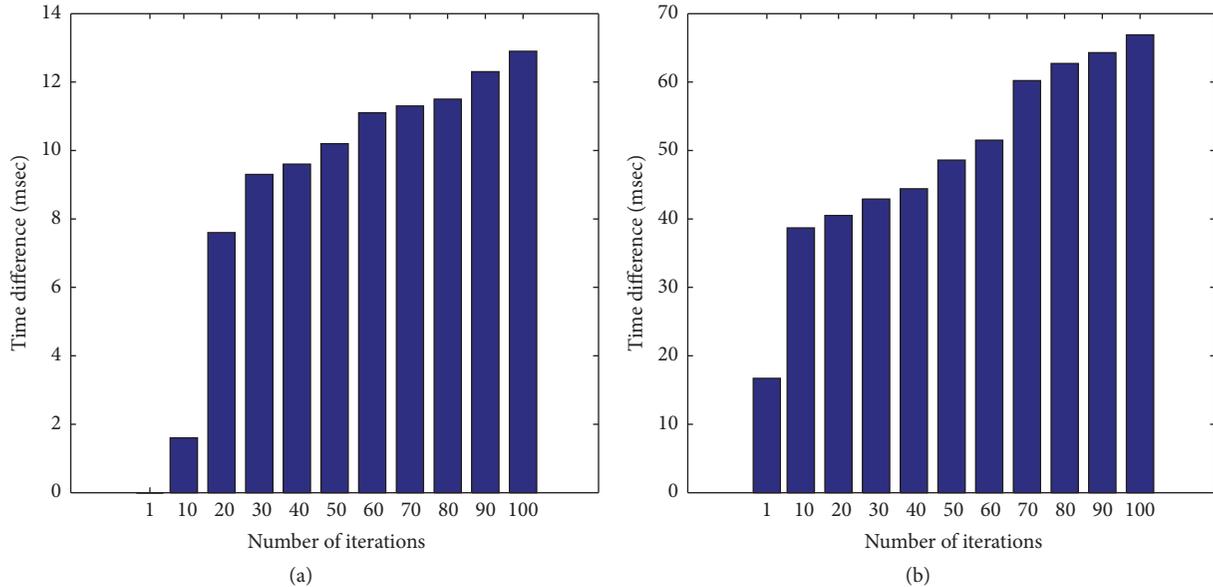


FIGURE 9: Processing time difference for 4×4 MIMO system. (a) Sequential RLS versus proposed PDASP technique. (b) Sequential linear Kalman versus proposed PDASP technique.

TABLE 1: Percentage improvement in decreased processing of PDASP MIMO RLS with respect to sequential MIMO RLS.

Doppler rate	MIMO RLS MSE convergence	MIMO RLS processing time, P_{RLS}	Proposed PDASP MIMO RLS convergence	Proposed PDASP RLS processing time, P_{PDASP} at $T_f = 41 \mu\text{sec}$	% improvement in decreased processing time, $(P_{\text{RLS}} - P_{\text{PDASP}})/P_{\text{RLS}} \times 100$
$f_D T = 10^{-6}$	25 iterations	0.0096373 sec	35 iterations	0.0016825 sec	82.29%
$f_D T = 10^{-3}$	30 iterations	0.0107447 sec	50 iterations	0.0024410 sec	77.28%

TABLE 2: Percentage improvement in decreased processing of PDASP with respect to sequential linear Kalman.

Doppler rate	Kalman MSE convergence	Kalman processing time, P_{Kalman}	Proposed PDASP MIMO RLS convergence	Proposed PDASP RLS processing time, P_{PDASP} at $T_f = 41 \mu\text{sec}$	% improvement in decreased processing time, $(P_{\text{Kalman}} - P_{\text{PDASP}})/P_{\text{Kalman}} \times 100$
$f_D T = 10^{-6}$	15 iterations	0.0403 sec	35 iterations	0.0016825 sec	95.83%
$f_D T = 10^{-3}$	20 iterations	0.0415 sec	50 iterations	0.0024410 sec	94.12%

20 iterations compared to the sequential Kalman and MIMO RLS algorithms, respectively. It can be seen that the proposed technique still entails 94.12% and 72.28% lesser processing time than the sequentially operated Kalman and MIMO RLS algorithms, respectively.

6. Conclusions

In this paper, a novel low complexity architecture for the parallelly distributed adaptive signal processing (PDASP) operation of inexpensive and computationally incapable small platforms has been proposed. The proposed architecture makes the inexpensive and computationally incapable devices run computationally expensive procedures like complex adaptive Kalman and RLS algorithms cooperatively. The operation of

the proposed PDASP architecture has been evaluated on the basis of presence of time nonalignment in the execution of its parallel block entities. Complexity and processing time of proposed PDASP scheme with RLS algorithm have been compared with those of sequentially operated Kalman and RLS algorithms. It has been observed that PDASP scheme exhibits much lesser computational complexity and processing time parallelly than the sequentially operated Kalman and RLS algorithms. The proposed PDASP technique with nonaligned time indexes provides parallelly $2N^2$ multiplications and N^2 additions per iteration at maximum. Likewise, the proposed technique utilizes 95.83% and 82.29% lesser processing time than the sequential Kalman and MIMO RLS algorithms, respectively, for low doppler rate. Likewise, for high doppler rate, the proposed technique entails 94.12%

and 77.28% decreased processing than sequentially operated Kalman and MIMO RLS algorithms, respectively. In a nutshell, processing time and parallel complexity of the proposed PDASP based MIMO RLS scheme have been observed to be much lesser than those of Kalman and RLS adaptive filtering algorithms, if operated sequentially on a single unit.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] K. Pelekanakis and M. Chitre, "Robust equalization of mobile underwater acoustic channels," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 4, pp. 775–784, 2015.
- [2] B. Zhang, L. Zhang, F. Karray, and L. Zhang, "Retinal vessel extraction by matched filter with first-order derivative of gaussian," *Computers in Biology and Medicine*, vol. 40, no. 4, pp. 438–445, 2010.
- [3] S. Bidyadhar and R. Pradhan, "An adaptive predictive error filter based maximum power pot trackg algorithm for a photovoltaic system," *The Journal of Engineering*, vol. 1, no. 1, 2016.
- [4] C. Schüldt, F. Lindstrom, and I. Claesson, "Low-complexity adaptive filtering implementation for acoustic echo cancellation," in *Proceedings of the IEEE Region 10 Conference (TENCON '06)*, Hong Kong, November 2006.
- [5] R. A. Soni, K. A. Gallivan, and W. K. Jenkins, "Low-complexity data reusing methods in adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 394–405, 2004.
- [6] C. Schüldt, F. Lindstrom, and I. Claesson, "A low-complexity delayless selective subband adaptive filtering algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5840–5850, 2008.
- [7] F. Z. Zhang, *The Schur Complement and Its Applications*, vol. 4 of *Numerical Methods and Algorithms*, Springer, New York, NY, USA, 2005.
- [8] R. R. Müller, L. Cottatellucci, and M. Vehkaperä, "Blind pilot decontamination," *IEEE Journal on Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 773–786, 2014.
- [9] S. Eberli, D. Cescato, and W. Fichtner, "Divide-and-conquer matrix inversion for linear MMSE detection in SDR MIMO receivers," in *Proceedings of the 26th Norchip Conference*, pp. 162–167, Tallinn, Estonia, November 2008.
- [10] M. Z. A. Bhotto and A. Antoniou, "Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments," *IEEE Signal Processing Letters*, vol. 18, no. 3, pp. 185–188, 2011.
- [11] L. Xiao, S. Liu, and D. Yang, "A low-complexity block diagonalization algorithm for mu-mimo two-way relay systems with complex lattice reduction," *International Journal of Distributed Sensor Networks*, Article ID 615202, 2015.
- [12] D. Wubben, R. Bohnke, V. Kuhn, and K. Kammeyer, "Near-maximum likelihood detection of MIMO systems using MMSE-based lattice reduction," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, pp. 798–802, June 2004.
- [13] M. L. Honig and W. Xiao, "Performance of reduced-rank linear interference suppression," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1928–1946, 2001.
- [14] B. Kovačević, Z. Banjac, and I. K. Kovačević, "Robust adaptive filtering using recursive weighted least squares with combined scale and variable forgetting factors," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, article 37, 2016.
- [15] S. Moshavi, E. G. Kanterakis, and D. L. Schilling, "Multistage linear receivers for DS-CDMA systems," *International Journal of Wireless Information Networks*, vol. 3, no. 1, pp. 1–17, 1996.
- [16] G. M. A. Sessler and F. K. Jondral, "Low complexity polynomial expansion multiuser detector for CDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1379–1391, 2005.
- [17] J. Hoydis, M. Debbah, and M. Kobayashi, "Asymptotic moments for interference mitigation in correlated fading channels," in *Proceedings of the IEEE International Symposium on Information Theory Proceedings (ISIT '11)*, pp. 2796–2800, rus, August 2011.
- [18] A. O. A. Noor, S. A. Samad, and A. Hussain, "A review of advances in subband adaptive filtering," *World Applied Sciences Journal*, vol. 21, no. 1, pp. 113–124, 2013.
- [19] A. Bertrand and M. Moonen, "Distributed adaptive node-specific MMSE signal estimation in sensor networks with a tree topology," in *Proceedings of the 17th European Conf. Signal Process*, pp. 794–798, Glasgow, UK, 2009.
- [20] S. Kajan, I. Sekaj, and M. Oravec, "The use of MATLAB parallel computing toolbox for genetic algorithm-based MIMO controller design," in *Proceedings of the 17th International Conference on Process Contro*, vol. 9, pp. 9–12, Štrbské Pleso, Slovakia, 2009.
- [21] E. Starkloff, "Designing a parallel, distributed test system," in *Proceedings of IEEE AUTOTESTCON*, pp. 564–567, Anaheim, Calif.
- [22] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, "GPUs and the future of parallel computing," *IEEE Micro*, vol. 31, no. 5, pp. 7–17, 2011.
- [23] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, "A performance study of general-purpose applications on graphics processors using CUDA," *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1370–1380, 2008.
- [24] A. H. Sayed and T. Kailath, "A state-space approach to adaptive rls filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–60, 1994.
- [25] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, USA, 4th edition, 2002.
- [26] J. Benesty and Y. Huang, *Adaptive Signal Processing-Applications to Real-World Problems*, Springer, Berlin, Germany, 2003.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

