

Research Article

The On-Chip D-LMS Filter Design Method of Wireless Sensor Node Based on FPGA

Jian Li , Maojin Li, Ming Meng, and Zepeng Liu

Shanxi Key Laboratory of Information Detection and Processing, North University of China, Taiyuan 030051, China

Correspondence should be addressed to Jian Li; lijian851208@126.com

Received 3 September 2019; Accepted 19 May 2020; Published 30 May 2020

Academic Editor: Adam Glowacz

Copyright © 2020 Jian Li et al. 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 the real-time position technology of underground shallow source, the signal denoising performance of wireless sensor nodes directly determines the location speed and accuracy of underground burst point. Because of the complexity and randomness of the underground medium and the fact that underground explosion is a nonstationary transient process, the problems of low convergence rate and poor steady-state performance of the filter exist when the existing LMS algorithm is used for signal denoising. In light of the above concerns, this paper comes up with a signal denoising algorithm and hardware implementation method based on D-LMS (delay-LMS). Firstly, according to the autocorrelation function characteristic of random signal, using the principle that the autocorrelation function time delay characteristic of narrowband signal such as explosion vibration signal is better than that of wideband random signal such as ground noise, the D-LMS filter algorithm is constructed by introducing the time delay parameter. Secondly, the selection method of key parameters in D-LMS hardware implementation is analyzed. Thirdly, the corresponding hardware circuit is designed by FPGA, and the simulation is carried out. Numerical simulation and experimental verification show that compared with the existing LMS improved algorithm, the D-LMS algorithm proposed in this paper has higher denoising stability and better denoising effect. Compared with the signal postprocessing method based on the host computer, the signal denoising speed of this method is significantly improved. This method will provide a powerful theoretical method to solve the problem of high-precision and fast source positioning and provide technical support for the development of high-speed and real-time source positioning instruments.

1. Introduction

Distributed source location in shallow underground spaces is a positioning technique that integrates sensing, networking, transmission, and positioning [1, 2]. It works by placing a large number of wireless sensor nodes on the ground, which are tasked with collecting vibration signals generated by explosions [3]. Following the explosion, the wireless transmission network transmits the sensor node data to the control center, where the position of the burst point is determined, as shown in Figure 1.

Quick location of a range of seismic sources is an emerging application, such as in real-time monitoring of underground chamber blasting, rapid location of fuze firing, and rapid prediction of rock burst, water burst, and mud burst [4]. In this context, effort on improving the processing

speed of the wireless sensor nodes themselves has become a hot subject in the field of rapid location of underground seismic sources [5].

When a wireless sensor node processes data, a key step in achieving accurate and fast positioning is denoising the vibration signal. Conventional methods for denoising underground seismic signals include wavelet threshold denoising [6], Gabor transform [7], weighted moving average filtering [8], singular value decomposition matrix analysis [9], and HHT [10]. Although these methods improve the signal-to-noise ratio of the vibration signal, they take up a great amount of hardware resource due to their computational complexity, which makes them difficult to implement on wireless sensor nodes and are thus inapplicable. LMS (least mean square) is a typical adaptive filtering algorithm. Despite its low convergence speed, it is simple in

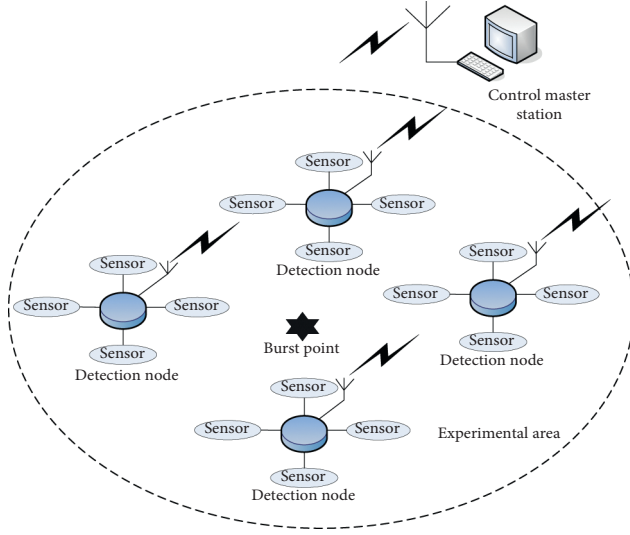


FIGURE 1: Schematic diagram of shallow underground distributed source location.

structure and involves a small amount of calculation and is easy to implement. Besides, it involves no extra calculation complexity if there are changes in the working environment. Therefore, from the perspective of VLSI (very large scale integration) hardware implementation, the LMS algorithm is the best way to complete wireless sensor node on-chip data preprocessing.

Continuing from the LMS algorithm, researchers have changed the principle of selecting the step size to come up with a variety of modified algorithms such as SVS-LMS [11], VS-LMS [12], Witch-LMS [13], Tanh-LMS [14], and Atan-LMS [15]. These algorithms, by constructing a functional relationship between the step factor and the error signal power, have worked with good results in the field of noise detection and noise cancellation. The underground vibration signal is a nonstationary transient time-varying signal, while the noise signal is a broadband random signal, and hence the error signal power spectrum is time-varying, with the result that when a step factor is used to adjust the filter coefficients, the filter convergence rate decreases, the filter's steady-state performance deteriorates, and the denoising effect worsens.

In view of the above problems, this paper introduces a delay parameter and proposes to build a D-LMS filter algorithm (delay least mean square), which leverages the characteristics of the autocorrelation function of the random signal; in terms of the time delay of the autocorrelation function, narrowband signals, such as the explosive vibration signal, are better than broadband random signals, such as the ground noise [16, 17]. By means of simulation, the denoising performance of D-LMS and existing LMS improved filter is compared in this paper, which also discusses about the key filter parameters for engineering implementations. A field experiment was performed to verify the performance of the proposed filter. The result suggests that the proposed filter circuit improves greatly the SNR of the input signal and accelerates the location process. This method adds some engineering value to the field of underground space applications [18–20].

2. Principle of D-LMS Adaptive Filters

According to the underground explosion fluctuation theory and the characteristics of the autocorrelation function of random signals, of which the ground noise is a type, the autocorrelation function of the ground noise attenuates fast, while narrowband random signals, such as explosion vibration signals, have an autocorrelation function that attenuates slow [21]. That means that the explosion vibration signal has a wider autocorrelation function than the ground noise. Therefore, with an appropriate delay length, the noise signal will no longer be correlated with the original signal, but the useful signal still keeps a good correlation with it [22, 23], as shown in Figure 2. Auto-correlation operation is performed on Part A and Part B, the former containing only noise signal while the latter containing both noise and useful signal. The result of the operation is shown in Figure 2.

It can be seen from Figures 2(b) and 2(c) that the autocorrelation of the noise signal in Part B decreases rapidly as the length Δ of the time delay changes, while the autocorrelation of the useful signal in Part A remains at a high level. Therefore, it is possible to extract the effective signal in the explosive vibration signal by introducing the time delay parameter and taking advantage of the autocorrelation characteristics of the signal. The specific structure of the proposed D-LMS (delay least mean square) is shown in Figure 3.

Let the signal collected by a wireless sensor node be $x(n) = s(n) + N(n)$, where $s(n)$ is the explosion vibration signal and $N(n)$ is the environmental noise. $x(n)$ is the expected signal, $w(n)$ is the adaptive filter coefficient, $d(n)$ is the filter input signal, $y(n)$ is the filter output signal, $e(n)$ is the error signal, μ is the step factor, and Δ is the delay amount. The D-LMS filter expression is as follows:

$$x(n) = s(n) + N(n), \quad (1)$$

$$d(n) = x(n - \Delta) = s(n - \Delta) + N(n - \Delta), \quad (2)$$

$$y(n) = w^H(n)d(n), \quad (3)$$

$$e(n) = x(n) - y(n), \quad (4)$$

$$w(n+1) = w(n) - 2\mu d(n)e^*(n). \quad (5)$$

The explosion vibration signal $s(n)$ has an autocorrelation function with a longer attenuation characteristic than the environmental noise $N(n)$ [24]. Therefore, if an appropriate delay time Δ is chosen, the signals $N(n)$ and $N(n - \Delta)$ are no longer correlated, while $s(n)$ and $s(n - \Delta)$ are still correlated to a good degree. So, the adaptive filter's output $y(n)$ is the best estimate of $s(n)$.

3. Simulation Verification

3.1. Performance Comparison of the D-LMS Filter and the Existing Modified LMS Filters. To further compare the performance of the D-LMS filter with the existing modified

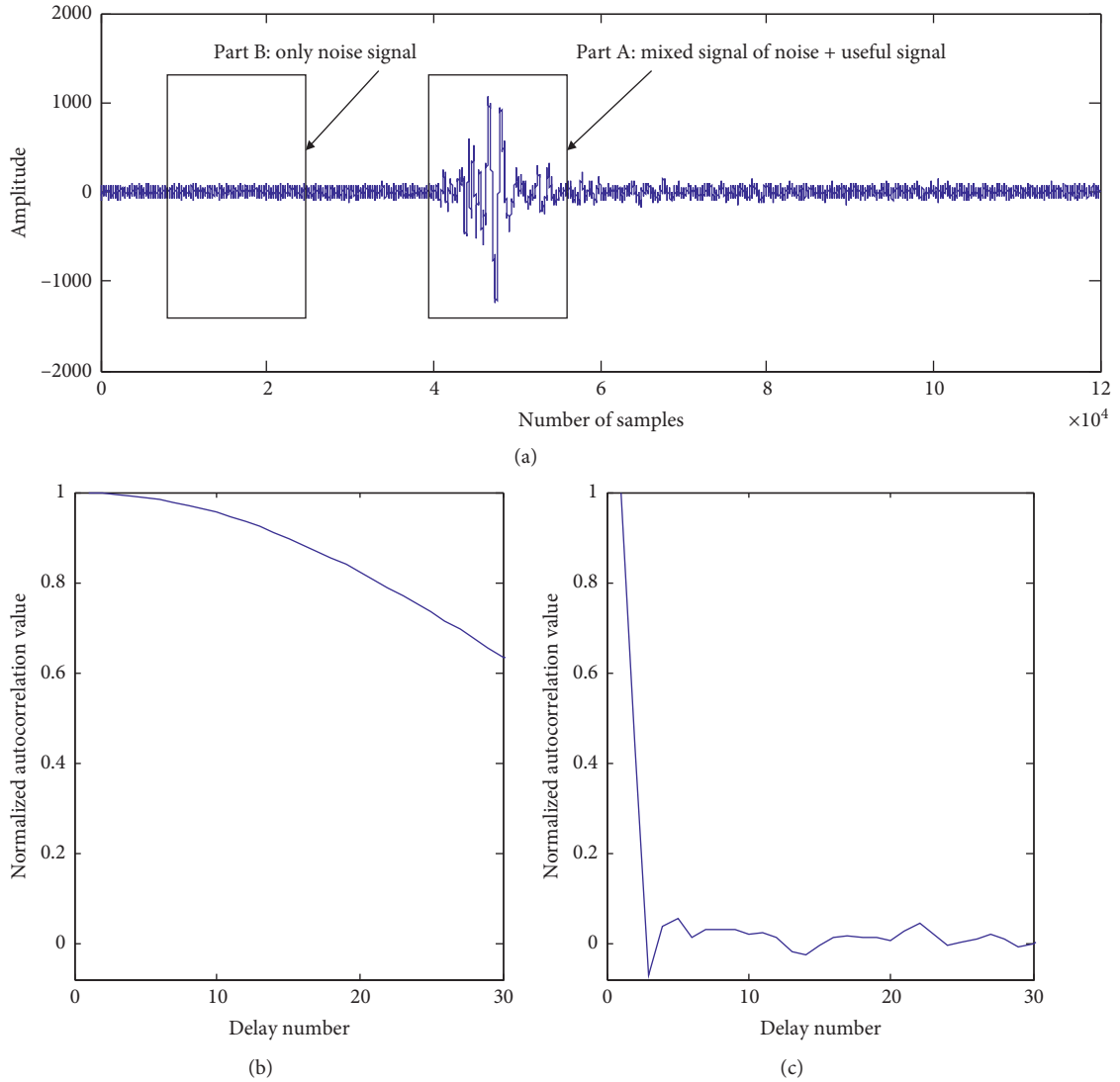


FIGURE 2: Autocorrelation results of signal and noise, respectively. (a) Underground explosion vibration signal. (b) Autocorrelation characteristic curve of Part A. (c) Autocorrelation characteristic curve of Part B.

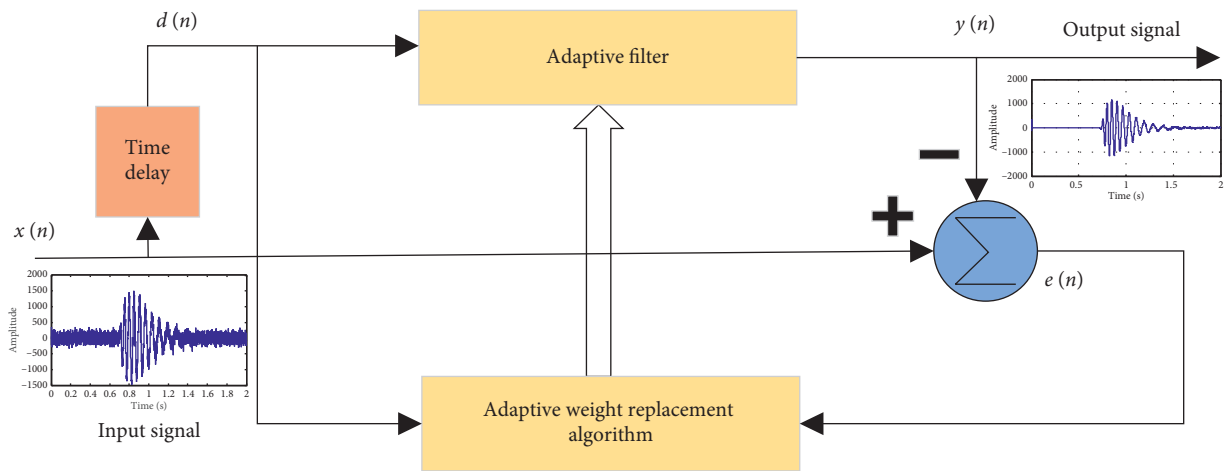


FIGURE 3: D-LMS filter model diagram.

LMS filters, a study was made that used explosion shock waves generated by artificial simulation. The signal sampling rate was set at 10 kHz, and the sampling time was set to 10 s. The useful explosive signal was a damped stretched sine wavelet. The setting frequency was 150 Hz, the arrival time of first break was at 1 s, the attenuation factor was $e^{-i/160}$ ($i = 0, 1, \dots, n$), the noise source was white Gaussian noise, and the initial SNR was -5 dB. The D-LMS filter delay points were set to be 10, and the filter order was set to be 12. Five modified LMS algorithms, namely, SVS-LMS, VS-LMS, Witch-LMS, Tanh-LMS, and Atan-LMS, were used to filter adaptively the explosive vibration signals 1200 times. The resulting error convergence curves are shown in Figure 4.

As can be seen from Figure 4, during 1200 iterations, the convergence speed and convergence accuracy of the D-LMS algorithm are significantly better than other algorithms.

3.2. Choice of Key Parameters in Hardware Implementation of D-LMS. As can be seen from the previous section, among the modified LMS algorithms, the D-LMS algorithm gives the best filtering effect when dealing with explosion vibration signals. In the hardware implementation of the D-LMS filter, the choice of the length of time delay Δ , the filter order N , and the step factor μ is critical to the performance of the filter.

3.2.1. Length of Time Delay Δ . The length of time delay Δ is a key parameter of the D-LMS algorithm to bring about good filtering [25, 26]. Due to the high explosion intensity, the SNRs of the explosion vibration signals collected by the wireless sensor nodes were all greater than 0 dB. Therefore, the emphasis was on the choice of the length of time delay under SNR = 0 dB condition. The signal sampling rate was set at 10 kHz, and the sampling time was set to 10 s. The useful explosive signal was a damped stretched sine wavelet. The setting frequency was 150 Hz, and the arrival time of first break was at 1 s. The noise source was white Gaussian noise, with an SNR of 0 dB. The underground explosion data containing the useful signal and also the data containing only the noise were extracted; autocorrelation was performed on them, respectively. The values thus found were normalized, which are shown in Figure 5(a). The results of the correlation computation are shown in Figure 5(b).

It can be seen from the Figure 5(b) that at a sampling rate of 10 kHz, when the time delay Δ is greater than 6, the correlation coefficient between noises $N(n)$ and $N(n - \Delta)$ is less than 0.2, while that between the useful signals $S(n)$ and $S(n - \Delta)$ is greater than 0.9. Therefore, the delay points were set to 6.

3.2.2. Filter Order N . The filter order N determines the complexity of the design of the circuit. With too small an N , the filtering error will be great and the filter just will not serve its filtering purpose or even result in unconvergence of the LMS algorithm. Too great an N , on the other hand, will increase the computational complexity and thus take up more hardware computing resources [27, 28].

The explosion seismic signal was simulated by a damped stretched sine wavelet, to which a noise with an SNR of 0 dB

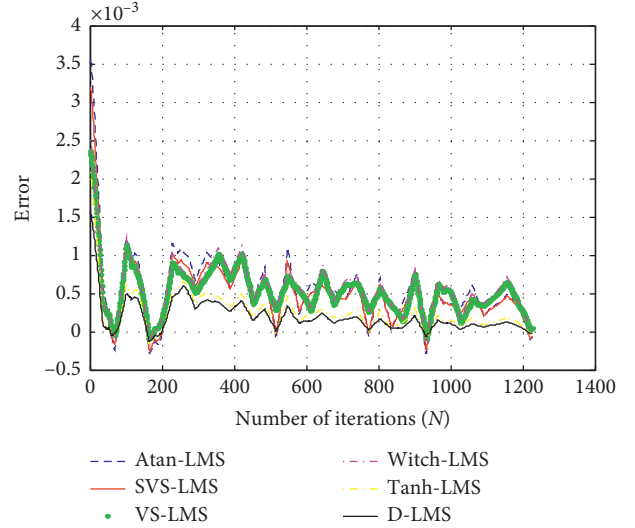


FIGURE 4: Error convergence curves of a few modified LMS algorithms.

was added. With the step factor μ kept constant, the filter order N was changed in the process of simulation; the simulation results are shown in Figures 6(a)–6(c).

With the step factor kept constant, the filtering performance improves gradually as the filter order N increases. When the filter order is greater than 12, the filtering effect stabilizes, and the SNR is greater than 30 dB. For subsequent positioning, when the SNR is greater than 30 dB, subsequent data processing requirements can be satisfied. To strike a balance between the filtering effect and the hardware implementation difficulty, this paper settles on an order $N = 12$.

3.2.3. Step Factor μ . The size of the step factor μ affects the convergence speed and steady-state error of the LMS algorithm [29]. A small step size gives a smaller steady-state error and a slower convergence rate; with a large step size, the convergence rate is faster, but the steady-state error becomes larger too [30].

Below is a filtering test of the measured signal, with the filter order being kept constant at $N = 12$ and the step factor μ being varied.

The simulation results in Figure 7(a) show that with too small a μ value, the convergence rate is slow, and the tracking performance of the filter coefficient is poor at the first break of the vibration, resulting in blurred jump point of the first break signal. In Figure 7(c), where the μ value is too large, the convergence rate becomes fast, but the filtering effect deteriorates. For adaptive filtering of transient vibration signals, it is necessary to prioritize the convergence rate, and after that, the filtering effect should be then improved. To cater to both needs, the choice falls on a $\mu = 1/2$.

4. Hardware Design of the FPGA-Based LMS Filter

4.1. General Hardware Design of FPGA-Based Wireless Sensor Nodes. The specific design diagram of the system is shown

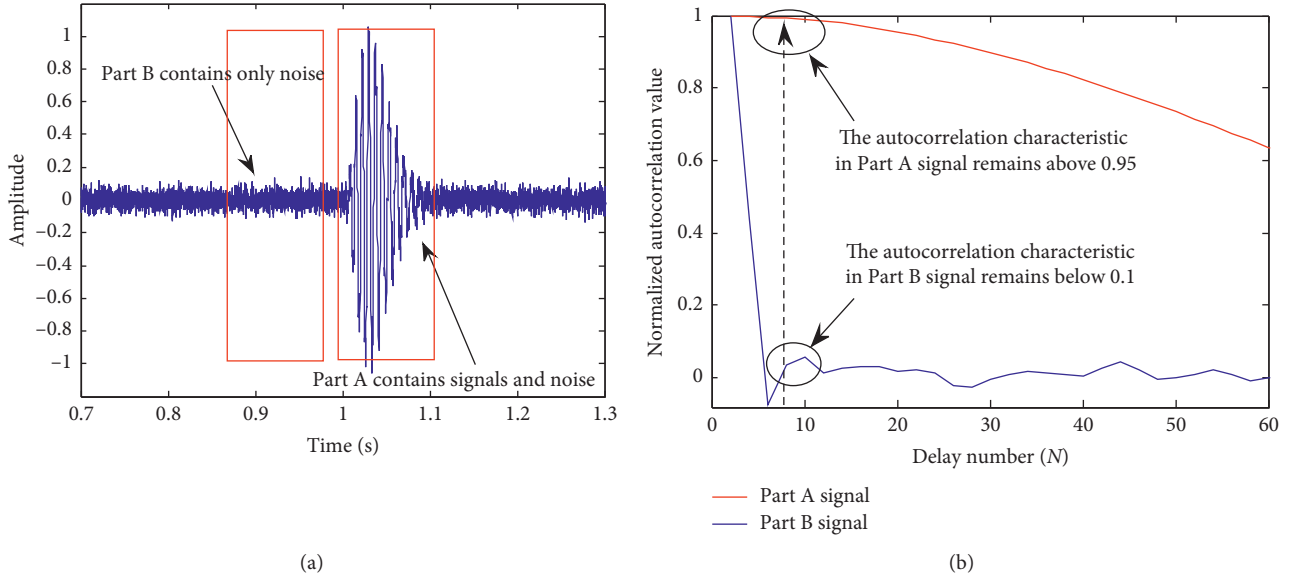


FIGURE 5: Autocorrelation results of signal and noise, respectively. (a) Underground explosion vibration signal. (b) Autocorrelation characteristic curves of 2 kinds of signals.

in Figure 8. This system incorporates ARM + FPGA architecture in its design of the wireless sensor nodes, and its D-LMS filter is implemented in an FPGA. The FPGA is responsible for explosion vibration signal collection, data filtering, and data storage; the ARM works as the control core, transferring the data collected by the FPGA to the WIFI module. The indicators of key devices are shown in Table 1.

In this study, the node control system was based on the FPGA + ARM architecture, whose core circuit board is shown in Figure 9. Major indices were as follows: (1) data storage capacity: 16 GB (expandable); (2) wireless transmission rate: 6.4 Mbps; (3) Beidou clock synchronization accuracy: 20 ns; (4) the system was powered by a 12 V/20000 mAh lithium battery; when the system was in the standby mode, the power consumption was 0.15 W, and when the system was in the working mode, the power consumption was 0.8 W

4.2. Deciding the Length of the LMS Filter System Coefficient $w(n)$. In digital signal processing systems, especially those that include feedback, the extra word length arising from multiplication can be truncated or limited by quantifying the results [31]. The DSP multiplier core integrated in the Xilinx series of FPGA is generally a 18 bit \times 18 bit multiplier. Therefore, for the ease of engineering implementation, the design should be ideally implemented based on a maximum of 18-bit fixed-point number system [32].

The input data are treated as 12-bit integer data, expressed as {1 sign bit + 2 integer bits + 9 decimal digits}, or as $\times(3, 9)$ for short. The weight coefficient is regarded as 18-bit integer data, expressed as {1 sign bit + 2 integer bits + 15 decimal places}, or $w(3, 15)$ for short. A large amount of addition and multiplication is involved in the filtering calculation process [33]. With a fixed finite word length, if two fixed-point numbers of N bits each are added, the result

is of $N+1$ bit; if they are multiplied with each other, the result is of $2N$ digits. Hence, $y(6, 24) = w(3, 15) \times (3, 9)$.

4.3. Actual Hardware Design of the LMS Filter. This system, following the top-down design principle, included a D-LMS filter consisting of 4 circuit modules, including a data latch module, a data filtering module, an error calculation module, and a coefficient update module. The data latch module was used to latch the input data and perform time delay operation on the data; the data filtering module worked to perform 12-order filtering operation on the data; the error calculation module was used to output the filtered result and estimate the error; the coefficient update module had a function of computing the filter coefficient.

The external input signals of this circuit included data clock (clk_din), system clock (clk), input data (din [11:0]), and data enable terminal (dn-en). The output signal of this circuit was (yout [11:0]). The data clock (clk_din) was synchronized with the data input (din), and the system clock (clk) was used to drive each module to work orderly.

The top-level circuit of the system is shown in Figure 10. Each module begins to work the moment its input port "start" receives the pulse signal. The work of a module is completed when it outputs endM pulse signal. The endM of the previous module is connected to the "start" input of the latter module to complete serial timing control.

For the convenience of describing each of the modules, the actual process of the time sequence design in the implementation of the 12-order LMS filter is given below:

Step 1: the filter module inputs data: $x_i = \text{din}_{(i+6)}$ ($i = 0 \sim 11$)

Step 2: filtering of calculation: $dy_i = w_i \times x_i$ ($i = 0 \sim 11$)

Step 3: calculate the filtering result and output it: $\text{yout} = \sum_{i=0}^{11} dy_i$

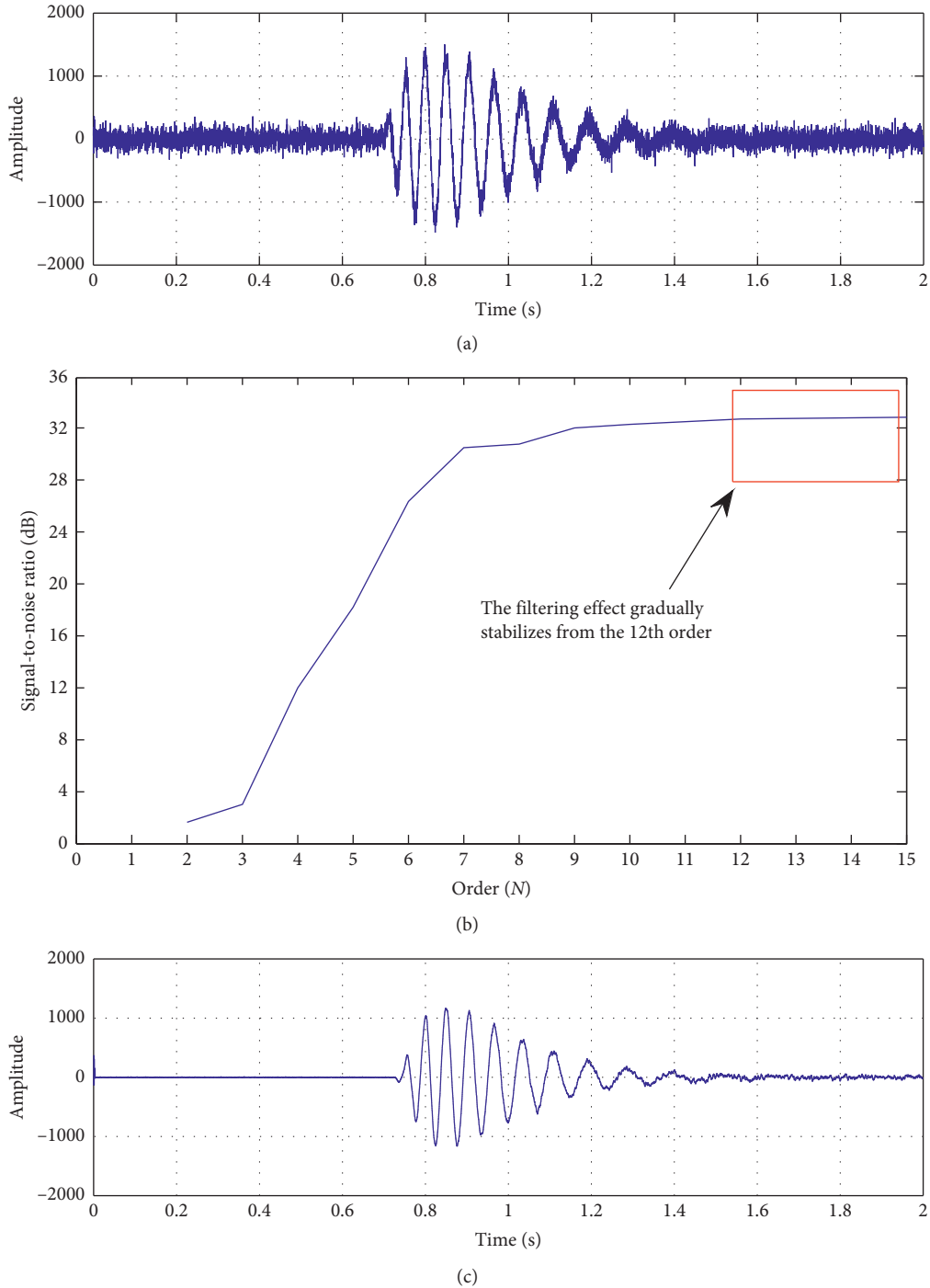


FIGURE 6: Simulation validation diagram of filter orders. (a) Simulation signal diagram of 0 dB noise explosion seismic wavelet. (b) Simulation results of SNR after filtering with different orders. (c) Simulation result graph of a filter with an order $N=12$.

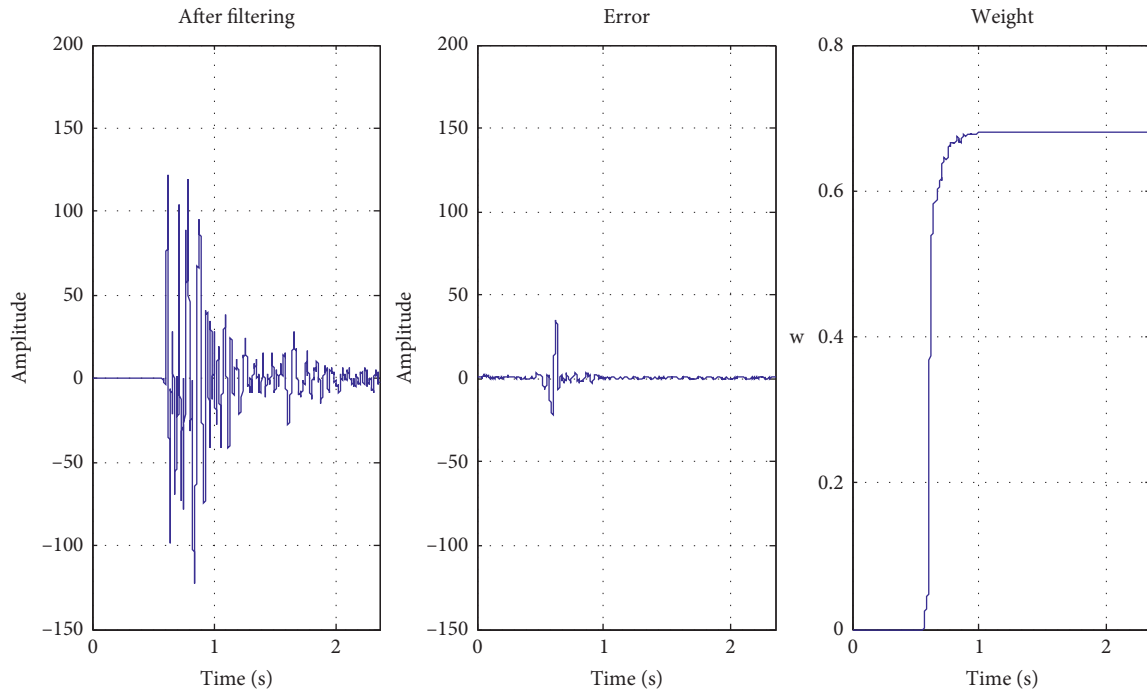
Step 4: error calculation: $e = \text{din} - \text{yout}$

Step 5: weight update step calculation: $dw_i = 2\mu x_i \times e$

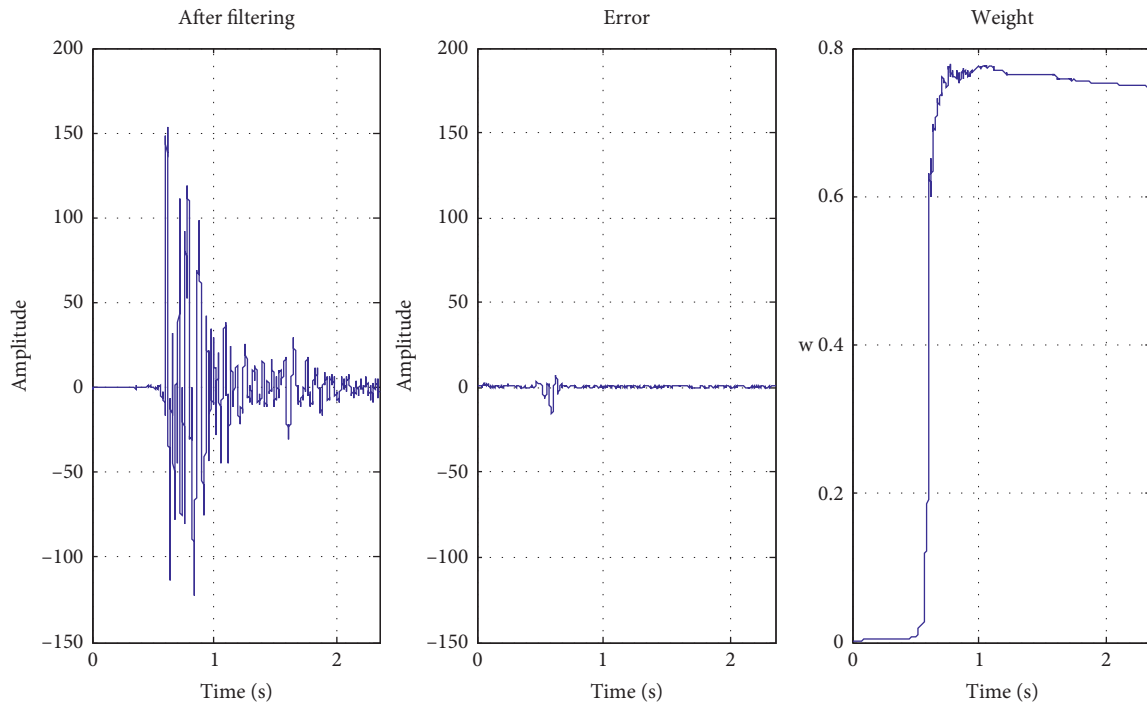
Step 6: update weight: $w_i = w_{i-1} + dw_i$

(a) The M_XnDnWn module is intended to latch the data input and the filter coefficient, as well as complete the operations of Steps 1 and 6. The module relies on the system clock (clk) to detect the rising

jump edge of the data clock (clk_din). When the rising jump edge of the data clock (clk_din) is detected, the new data din are considered to be coming, and the input data latching and the shift operation begin. The data input "din," after a delay of 6 data cycles ($d_1 \sim d_6$), enters the shift latch $x_0 \sim x_{11}$, and $x_0 \sim x_{11}$ is taken as the input vector of the filter. At the end of the shift operation, the output



(a)



(b)

FIGURE 7: Continued.

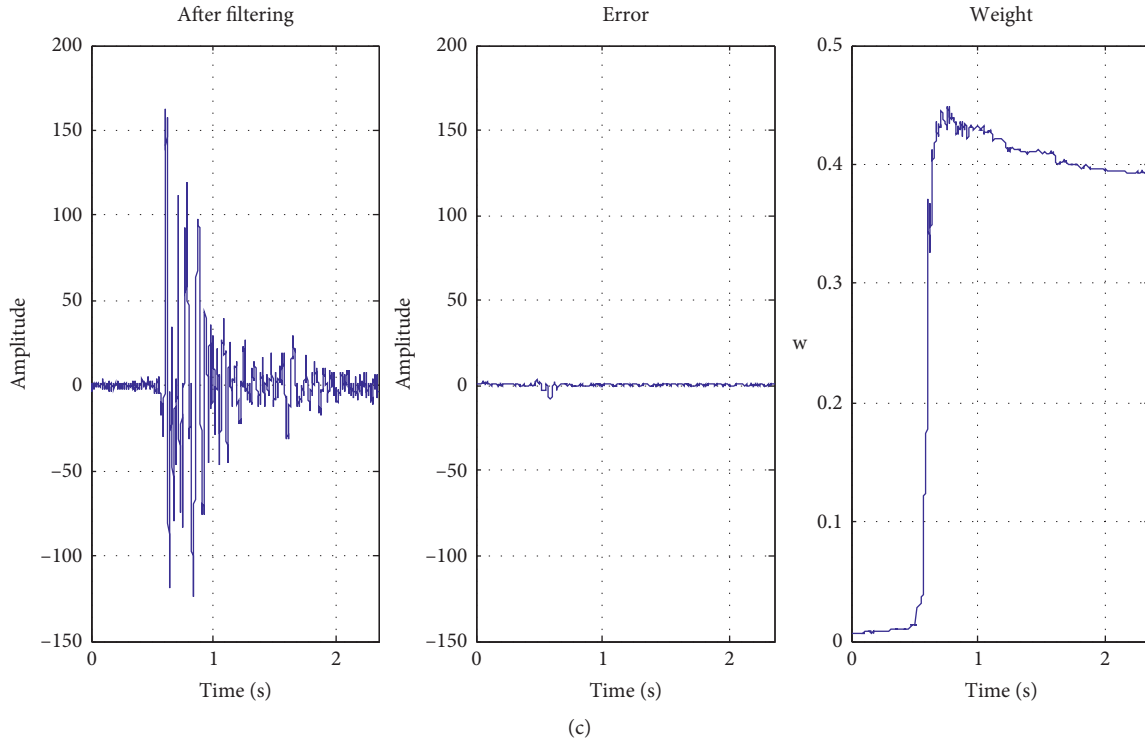


FIGURE 7: Simulation results of measured signal with different μ values. (a) $\mu = 1/8$ filtering result graph. (b) $\mu = 1/2$ filtering result graph. (c) $\mu = 2$ filtering result graph.

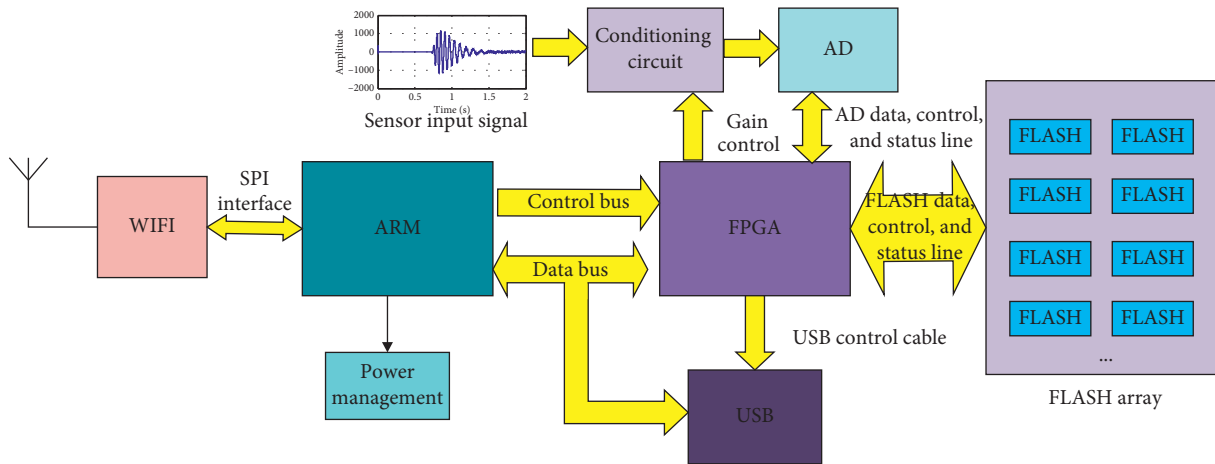


FIGURE 8: Design block diagram of the multichannel synchronous data acquisition module.

- $x_0 \sim x_{11}, w_0 \sim w_{11}$ ($w_0 \sim w_{11}$ is the filter weight vector) is maintained, and the output pulse signal endM notifies the M_filter module to start working. The integrated circuit of this module is shown in Figure 11.
- (b) The data filtering module completes Step 2. When the module's input "start" receives the pulse, it multiplies the corresponding input data x_i by w_i , with the product sent to the corresponding port dy_i . In this step, 12 12 bit * 18 bit multipliers are used, and the filter calculation is completed in

- parallel by the multiplier group. The integrated circuit of this part is shown in Figure 12.
- (c) The error calculation module completes the accumulation calculation of Step 3 and the error calculation of Step 4. The integrated circuit of this part is shown in Figure 13.
- (d) The coefficient update module completes the calculation in step 5 with 12 12 bit * 12 bit multipliers doing the calculation in parallel. At the same time, the saturation truncation method is used to convert the 24 bit multiplication result to 18 bit values. The

TABLE 1: Indicators of key devices.

Function unit	Main components	Technical index
FPGA	Spartan6-XC6SLX16-FT256	Basic logic units: 14,579; configurable logic blocks: 18,224; embedded memory block capacity: 576 KB; DSP computing core: 32; IO interfaces: 186.
ARM	STM32F407VET6	Maximum frequency: 168 MHz; maximum flash memory: 1 MB; IO interfaces: 140; 12 16-bit timers; 2 32-bit timers; each timer has up to 4 IC/OC/PWMs.
ADC	MAX1308	Power supply: ± 5 V; conversion accuracy: 12 bit; number of channels: 8; maximum sampling rate: 500 Ksps/channel; input range: 9 V; operating temperature: -40°C to $+85^{\circ}\text{C}$.
WIFI	W5500	Supports LWIP, TCP, UDP, ICMP, IPv4, ARP, and IGMP; a built-in 32 KB large-capacity transceiver buffer unit; supports speed adaptivity and full duplex and half duplex communication.
FLASH	MT29F4G08AFABA	Storage capacity: 4G; block size: 64 pages (128K + 4 KB); page size: 2,112 bytes (2,048 + 64 bytes); operating temperature: -40°C to $+85^{\circ}\text{C}$.

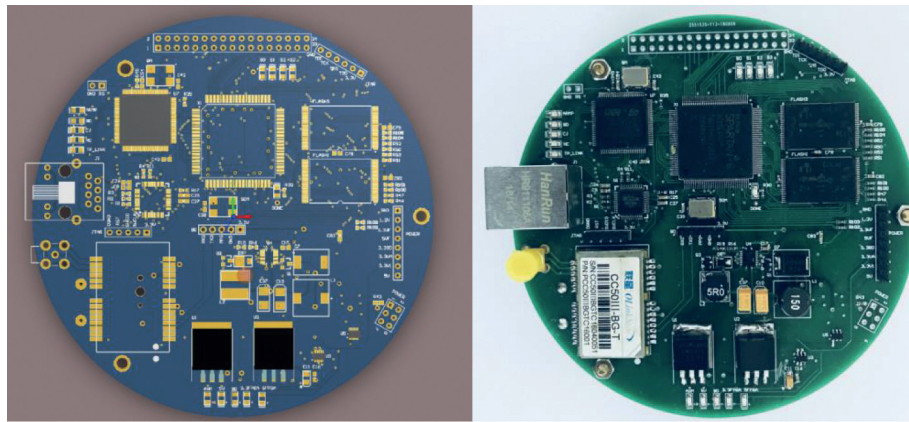


FIGURE 9: Photos of the core circuit board of the wireless vibration sensor node.

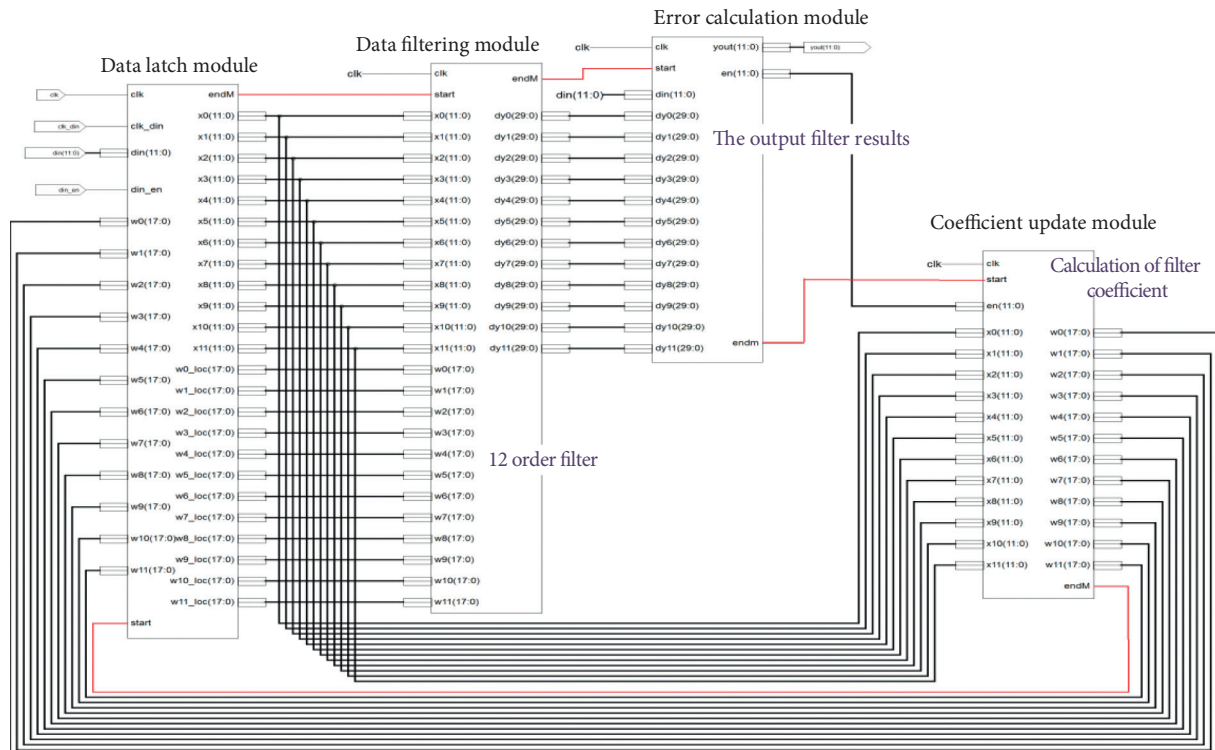


FIGURE 10: Schematic diagram of the LMS filter top module.

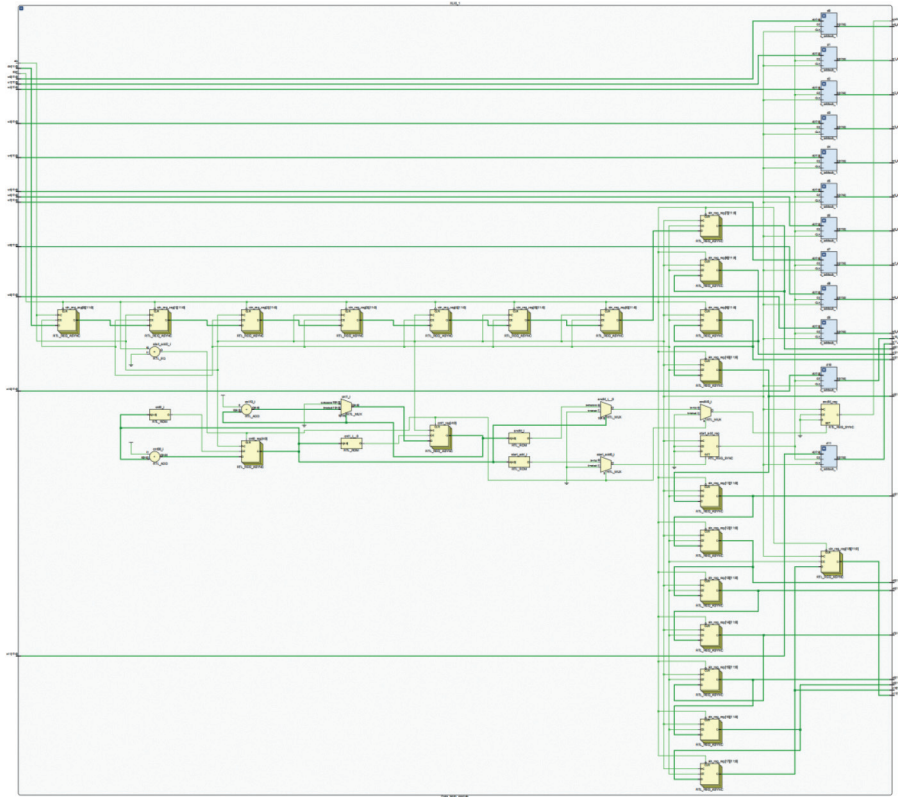


FIGURE 11: Integrated circuit diagram of the data latch module.

shift operation is performed according to the step factor, and the above calculation result is converted into a filter coefficient. The integrated circuit of this part is shown in Figure 14.

- (e) Resource utilization: the above circuit modules were then synthesized, and the total resource utilization on the FPGA is shown in Table 2.

4.4. Simulation Verification. MATLAB and VIVADO (2017.4 version) were used in a joint simulation to verify the function. The stimulus test file was written using the MATLAB software. The explosion seismic signal was a damped stretched sine wavelet, to which a noise with an SNR of 0 dB was added. The sampling rate was set to 10 kHz. The above signal was quantized as the input data of the system. The filter order was set to 12, and the step factor was set to 1/2. The above data were loaded into VIVADO software; the output result is shown in Figure 15.

The simulation result shows that the signal output from the filter, compared with the input signal, obviously has a higher SNR. It can be seen that the hardware filter designed in this study is capable of effective signal denoising.

5. Field Experiment

To test the signal denoising performance of this system, a wireless sensor network was constructed, consisting of 8 self-developed sets of wireless sensor nodes, APs, and bridges.

The test site, which was 20 km away from the control master station, was loaded with 5 kg TNT to create an artificial vibration source. Following the quake, two solutions—terminal data postprocessing and on-chip real-time processing—were used, respectively, to assess the performance and speed of data preprocessing. The test site layout is shown in Figure 16.

The dots in the figure represent sensors, and * represents the burst point. The devices were positioned, and each seismic measuring point and the preset explosion point were calibrated for their location with the aid of the high-precision Beidou. The data are given in Tables 3 and 4.

The test site is shown in Figure 17.

The number of acquisition channels of a wireless sensor node was set to be 8, sampling rate was set to 10 kHz, and sampling time was set to 2.5 s. The wireless transmission rate was 6.4 Mbps. The original vibration signal collected by wireless sensor node 1 in the experiment is shown in Figure 18(a). The signal after on-chip preprocessing is shown in Figure 18(b). On the upper computer, the original signal was processed with the MATLAB software and the D-LMS filter algorithm, with the result shown in Figure 18(c).

The signal processing results under 2 modes were found by calculation, as shown in Table 5. Number of wireless sensor nodes: 16; number of internal sampling channels of a wireless sensor node: 8; sampling rate: 10 kHz; sampling time: 2.5 s; and quantization bit number: 12 bit, stored in 16 bit. Therefore, the total data storage capacity of the wireless sensor network was 6.4 MB. The wireless transmission rate

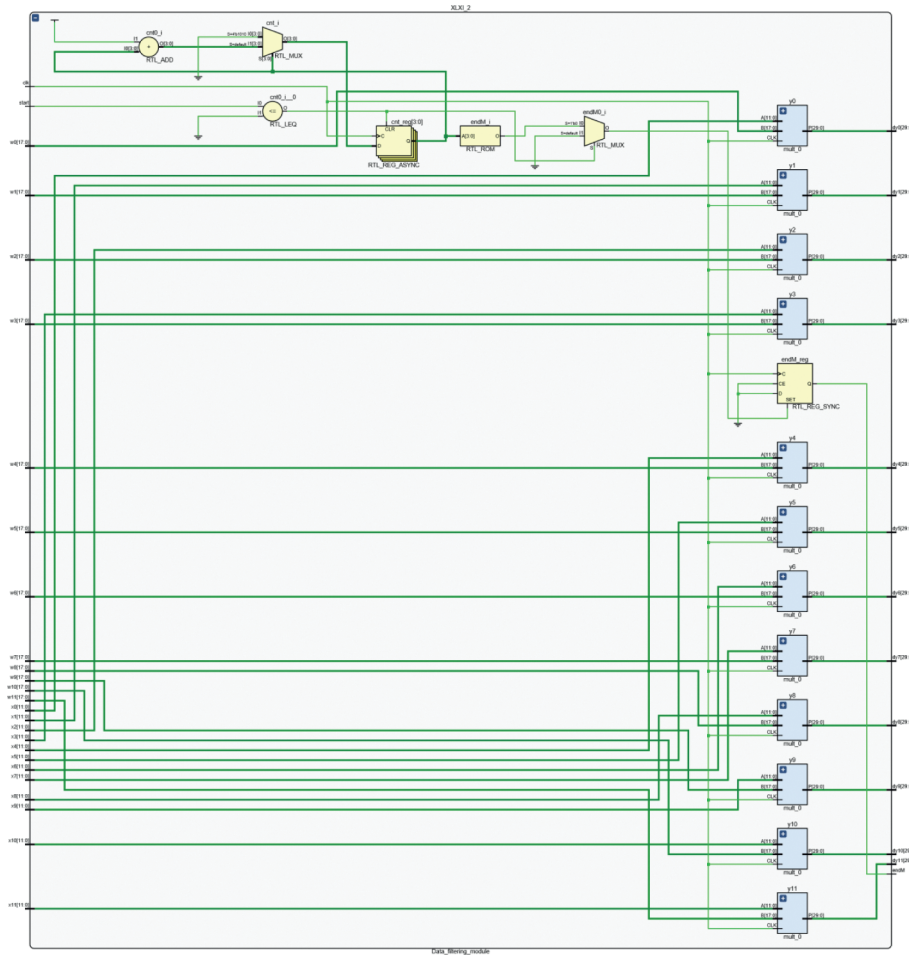


FIGURE 12: Integrated circuit diagram of the data filtering module.

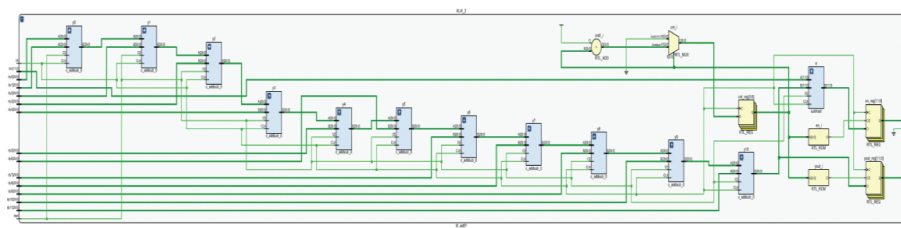


FIGURE 13: Integrated circuit diagram of the error calculation module.

was set at 6.4 Mbps, and it took 10 s to transmit all of the data. The upper computer did not begin to process the data until receiving all of the data. Therefore, upper computer processing lagged at least 10 s behind on-chip processing.

As can be seen from the table, due to the time spent on manual software operation, the PC software preprocesses the data much slower than the on-chip real-time solution. Though the filtering effect of the on-chip real-time solution fell slightly behind the PC software, it gave an SNR that was

still above 30 dB. Given the extraction principle of the first break arrival time—an important positioning parameter, when the SNR is greater than 28 dB, the first break arrival time can be extracted practically with no inferior accuracy [34]. Therefore, the proposed on-chip preprocessing method is able to achieve the same effect as can be achieved on upper computer software. More importantly, this method cuts the signal processing time of an individual sensor node by nearly 10 s, which is a great merit in real-time source location.

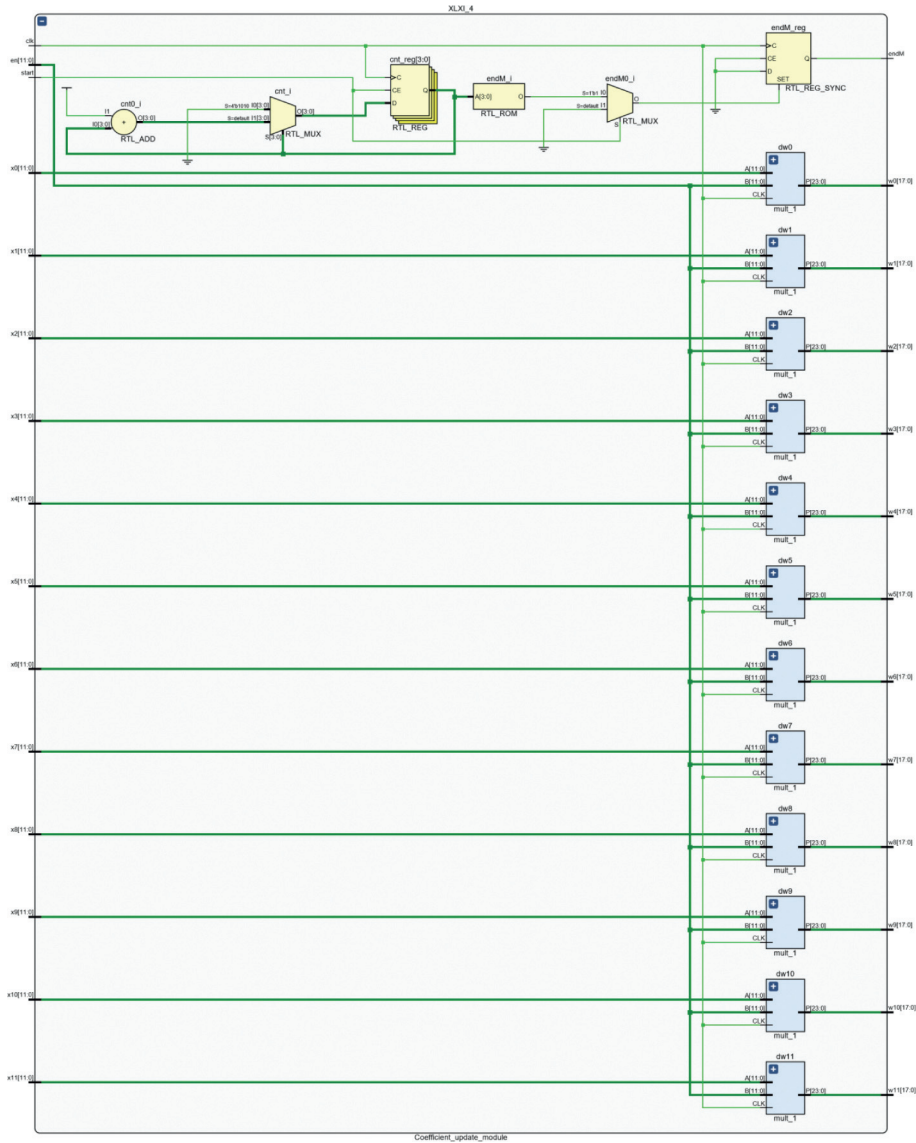


FIGURE 14: Integrated circuit diagram of the coefficient update module.

TABLE 2: Resource utilization rate of the filter module.

Resource	Utilization	Available	Utilization (%)
LUT	562	14579	3.85
FF	586	18224	3.22
DSP	24	32	75.00
IO	25	256	9.77

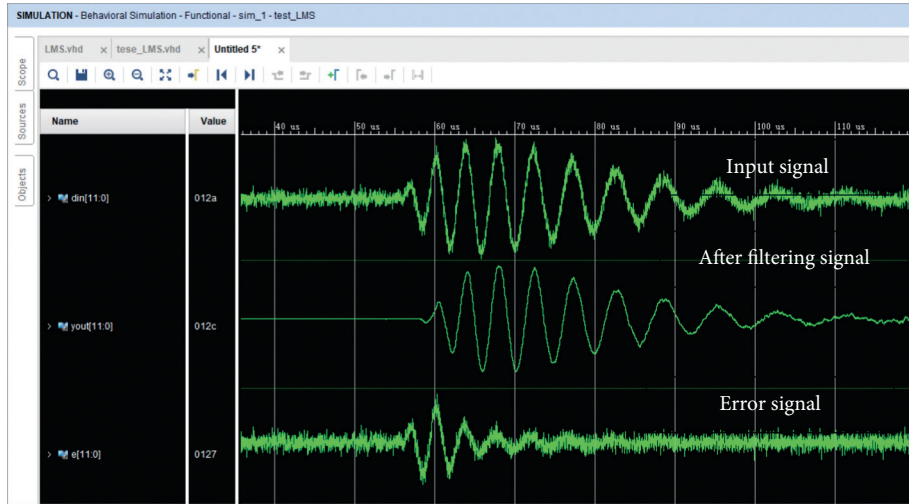


FIGURE 15: Denoising simulation results of the filter hardware.

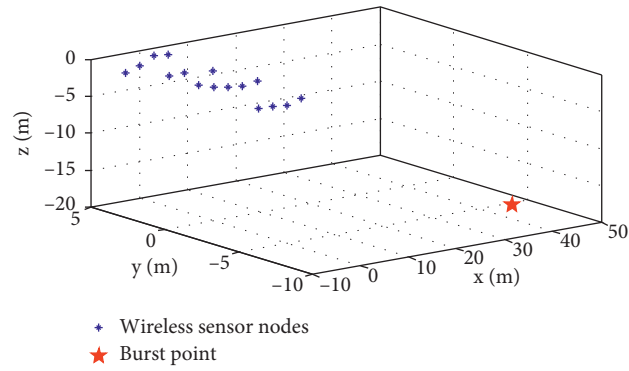


FIGURE 16: Sensor layout.

TABLE 3: Coordinates of the sensors.

Sensor serial number	Actual coordinates (unit: m)			Sensor serial number	Actual coordinates (unit: m)		
	X	Y	Z		X	Y	Z
1	-4.50	-4.50	-1.54	9	1.50	-4.50	-1.78
2	-4.50	-1.50	-0.47	10	1.50	-1.50	-0.96
3	-4.50	1.50	-0.73	11	1.50	1.50	-2.76
4	-4.50	4.50	-2.12	12	1.50	4.50	-0.54
5	-1.50	-4.50	-1.52	13	4.50	-4.50	-1.21
6	-1.50	-1.50	-0.76	14	4.50	-1.50	-0.64
7	-1.50	1.50	-0.65	15	4.50	1.50	-1.13
8	-1.50	4.50	-1.64	16	4.50	4.50	-0.83

TABLE 4: Location of the detonation point.

Serial number	Dosage (kg)	Actual coordinates (unit: m)		
		X	Y	Z
1	5	43.21	-6.17	-19.12

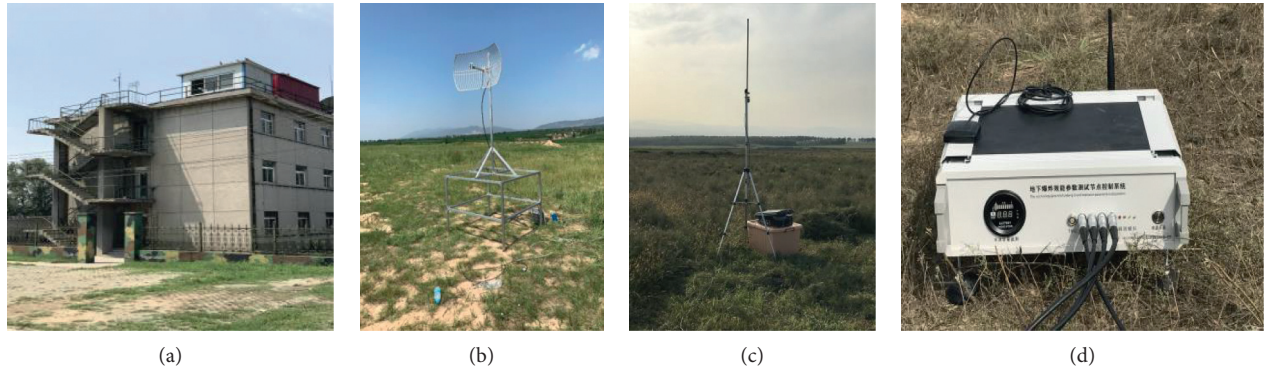


FIGURE 17: Field layout of the control distance of the wireless sensor nodes. (a) Main control station. (b) Field transmitting antenna. (c) Wireless APs. (d) Wireless sensor node.

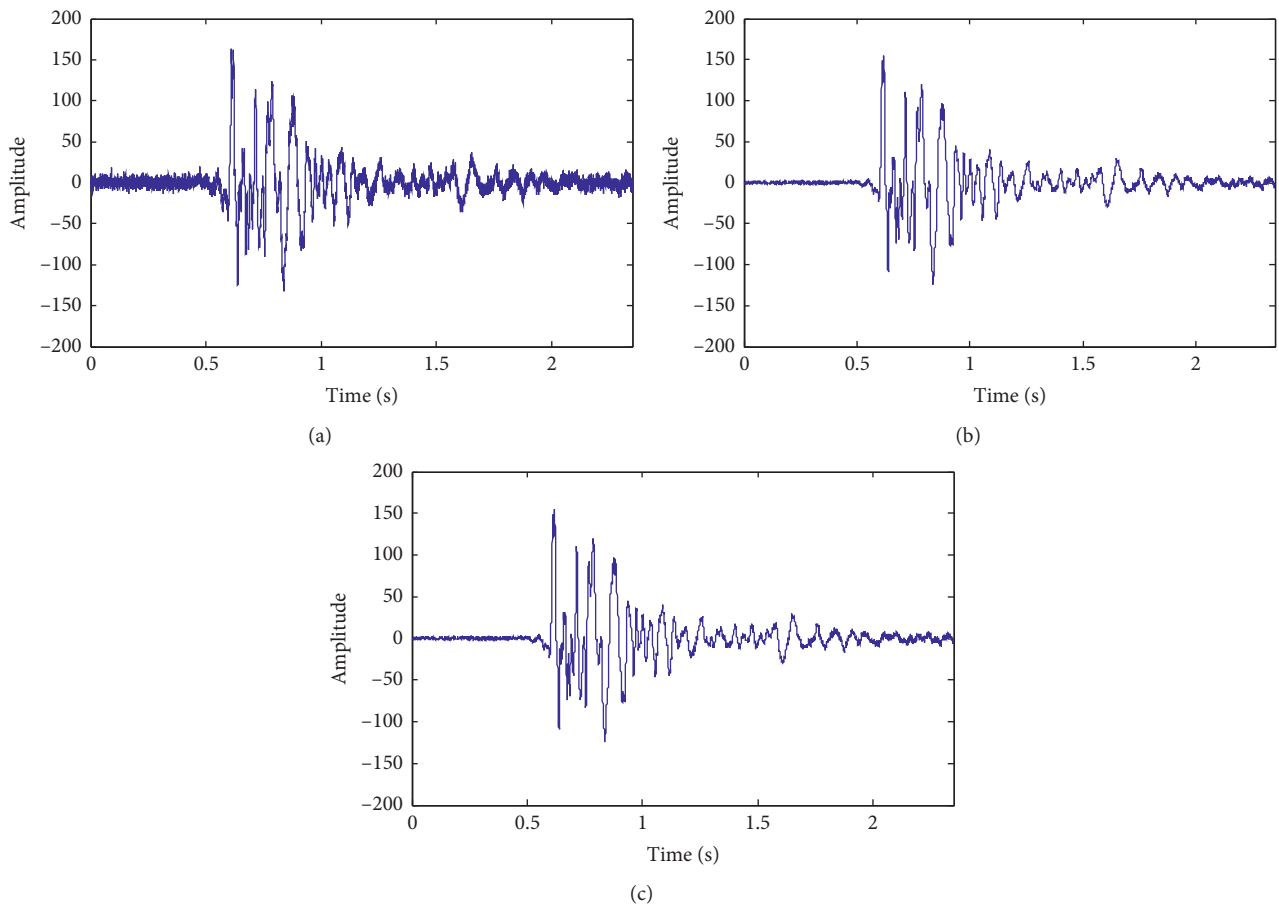


FIGURE 18: Diagram of different modes of signal processing. (a) Original input signal diagram. (b) Signals after on-chip preprocessing. (c) Signals after processing by the upper computer.

TABLE 5: Results of signal preprocessing.

Category	SNR (dB)	Runtime (s)
Original signal	10	—
Signals after on-chip hardware preprocessing	30	1.085
Signals after preprocessing by upper computer software	32	10.344

6. Conclusions

This paper proposes a signal denoising algorithm based on D-LMS as well as an FPGA hardware design method to cater to the needs of rapid and high-precision location of underground sources. First, the proposed D-LMS and the existing modified LMS were compared through simulation for their denoising effect. Then, analysis was provided on how to determine the length of time delay, the filter order, and the step factor—all critical parameters in implementing the D-LMS hardware. The circuit modules were designed using an FPGA for the data latch, filter, error calculation, and weight updating modules. A field experiment was also carried out. The results demonstrate that the designed filter circuit is capable of greatly improving the data processing speed of explosion vibration signals. Compared with post-processing methods based on an upper computer, the proposed method shortens the data processing time by 30 s. This method is of some engineering application value in the field of underground source positioning.

Data Availability

The field experiment data used to support the findings of this study have not been made available because the internal data are updated in real time and cannot be directly saved.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jian Li was responsible for overall design and filter structure design; Maojin Li was responsible for FPGA-based hardware code design; Zepeng Liu was responsible for PCB circuit board design; and Ming Meng was responsible for D-LMS filter key parameter simulation and field test.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 61901419), Shanxi Provincial Youth Fund (no. 201801D221205), Shanxi Provincial University Innovation Project Funding (no. 201802083), "13th Five-Year" Equipment Pre-Research Weapons Industry Joint Fund (nos. 6141B012895 and 6141B021303).

References

- [1] S.-C. Lin, A. A. Alshehri, P. Wang, and I. F. Akyildiz, "Magnetic induction-based localization in randomly deployed wireless underground sensor networks," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1454–1465, 2017.
- [2] I. F. Akyildiz and E. P. Stuntebeck, "Wireless underground sensor networks: research challenges," *Ad Hoc Networks*, vol. 4, no. 6, pp. 669–686, 2006.
- [3] J. Li, Y. Liu, Y. Han, and X. Chen, "A new method to access information for underground omni-dimensional vibration vector," *Sensor Review*, vol. 35, no. 1, pp. 125–132, 2015.
- [4] J. Wang, Y. Guo, Y. Jia, Y. Zhang, and M. Li, "Modeling and application of the underground emergency hedging system based on internet of things technology," *IEEE Access*, vol. 7, pp. 63321–63335, 2019.
- [5] J. Liu, X. Zeng, J. Xia, and M. Mao, "The separation of p- and s- wave components from three-component crosswell seismic data," *Journal of Applied Geophysics*, vol. 82, pp. 163–170, 2012.
- [6] J. C. Liu, W. X. Gao, and J. Jin, "Vibration signal analysis of water seal blasting based on wavelet threshold denoising and HHT transformation," *Advances in Civil Engineering*, vol. 2020, Article ID 4381480, 14 pages, 2020.
- [7] T. Yang, R. Z. Li, and N. Liang, "The application of key feature extraction algorithm based on gabor wavelet transformation in the diagnosis of lumbar intervertebral disc degenerative changes," *PLoS One*, vol. 15, no. 2, pp. 54–59, 2020.
- [8] O. F. Yalcin and M. Dicleli, "Effect of the high frequency components of near-fault ground motions on the response of linear and nonlinear SDOF systems: a moving average filtering approach," *Soil Dynamics and Earthquake Engineering*, vol. 129, pp. 89–97, 2020.
- [9] G. G. Jesus and Z. R. Marta, "Singular value decomposition for bidiagonal filter matrices," *Applied Mathematics & Computation*, vol. 289, pp. 214–219, 2016.
- [10] J. Cai, "A combinatorial filtering method for magnetotelluric time-series based on hilbert-huang transform," *Exploration Geophysics*, vol. 45, no. 2, pp. 63–73, 2014.
- [11] K. Seema and S. Bhim, "Improved reweighted zero-attracting quaternion-valued LMS algorithm for islanded distributed generation system at IM load," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3705–3716, 2020.
- [12] Q. Shu, J. M. Zhang, and F. W. Xu, "Set-membership LMS adaptive algorithms based on an error-estimation time-varying bound method," *Circuits, Systems and Signal Processing*, vol. 39, no. 7, pp. 3711–3736, 2020.
- [13] R. K. Sarma, M. T. Khan, R. A. Shaik, and J. Hazarika, "A novel time-shared and lut-less pipelined architecture for lms adaptive filter," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 1, pp. 188–197, 2020.
- [14] D. Shi, W.-S. Gan, J. He, and B. Lam, "Practical implementation of multichannel filtered-x least mean square algorithm based on the multiple-parallel-branch with folding architecture for large-scale active noise control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 940–953, 2020.
- [15] S. M. Jung, J.-H. Seo, and P. Park, "A variable step-size diffusion normalized least-mean-square algorithm with a combination method based on mean-square deviation," *Circuits, Systems, and Signal Processing*, vol. 34, no. 10, pp. 3291–3304, 2015.
- [16] L. Wu and C. C. Jong, "A high-throughput VLSI architecture for real-time full-HD gradient guided image filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1868–1877, 2019.
- [17] B. K. Mohanty, P. K. Meher, S. K. Singhal, and M. N. S. Swamy, "A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic," *Integration*, vol. 54, pp. 37–46, 2016.
- [18] J. K. Lee, "A parallel attitude-heading kalman filter without state-augmentation of model-based disturbance components," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 7, pp. 2668–2670, 2019.

- [19] D. Bismor and M. Pawelczyk, "Stability conditions for the leaky lms algorithm based on control theory analysis," *Archives of Acoustics*, vol. 41, no. 4, pp. 731–739, 2016.
- [20] B. Lagos-Álvarez, L. Padilla, J. Mateu, and G. Ferreira, "A kalman filter method for estimation and prediction of space-time data with an autoregressive structure," *Journal of Statistical Planning and Inference*, vol. 203, pp. 117–130, 2019.
- [21] R. Abdolee, V. Vakilian, and B. Champagne, "Tracking performance and optimal adaptation step-sizes of diffusion-lms networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 67–78, 2018.
- [22] T. Bensouici, A. Charef, and A. Imen, "A simple design of fractional delay fir filter based on binomial series expansion theory," *Circuits, Systems, and Signal Processing*, vol. 38, no. 7, pp. 3356–3369, 2019.
- [23] S. Liu, G. Shen, W. Zhang, and H. Tian, "A 60 GHz RoF system with blind VSS-DD-LMS equalizer for optical-wireless transmission," *IEEE Photonics Technology Letters*, vol. 28, no. 21, pp. 2383–2386, 2016.
- [24] M. Xiang, S. Kanna, and D. P. Mandic, "Performance analysis of quaternion-valued adaptive filters in nonstationary environments," *IEEE Transactions on Signal Processing*, vol. 66, no. 6, pp. 1566–1579, 2018.
- [25] P. Lara, L. D. T. J. Tarrataca, and D. B. Haddad, "Exact expectation analysis of the deficient-length lms algorithm," *Signal Processing*, vol. 162, pp. 54–64, 2019.
- [26] S. Zubair, N. I. Chaudhary, Z. A. Khan, and W. Wang, "Momentum fractional lms for power signal parameter estimation," *Signal Processing*, vol. 142, pp. 441–449, 2018.
- [27] J. Ma, L. Shi, Y. Li, S. Xiao, and X. Wang, "Angle estimation of extended targets in main-lobe interference with polarization filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 169–189, 2017.
- [28] A. K. Maurya, P. Agrawal, and S. Dixit, "Modified model and algorithm of lms adaptive filter for noise cancellation," *Circuits, Systems, and Signal Processing*, vol. 38, no. 5, pp. 2351–2368, 2019.
- [29] G. Narimani, P. A. Martin, and D. P. Taylor, "Spectral analysis of fractionally-spaced MMSE equalizers and stability of the LMS algorithm," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1675–1688, 2018.
- [30] M. Wang, B. Zhao, and M. Liu, "A high-speed realization of the delayed dual sign lms algorithm," *IEICE Electronics Express*, vol. 15, no. 7, pp. 546–549, 2018.
- [31] C. Aleksandr, P. Janusz, and M. M. Dorota, "Some structures of parallel vlsi-oriented processing units for implementation of small size discrete fractional fourier transforms," *Electronics*, vol. 8, no. 5, pp. 56–59, 2019.
- [32] K. S. Reddy and H. N. Suresh, "A low-power vlsi implementation of rfir filter design using radix-2 algorithm with lcls," *IETE Journal of Research*, vol. 10, pp. 96–99, 2019.
- [33] B. Pandey, N. Pandey, A. Kaur, D. M. Akbar Hussain, B. Das, and G. S. Tomar, "Scaling of output load in energy efficient FIR filter for green communication on ultra-scale FPGA," *Wireless Personal Communications*, vol. 106, no. 4, pp. 1813–1826, 2018.
- [34] M. J. Piggott and V. Solo, "Diffusion lms with correlated regressors i: realization-wise stability," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5473–5484, 2016.