



FPGA-GPU heterogeneous real-time OFDR strain sensing using streaming linear-interpolation resampling

CAILING FU,^{1,2}  SHIJIE LI,^{1,2} RONGYI SHAN,^{1,2} SHIZHU SUN,^{1,2}
HUAJIAN ZHONG,^{1,2} ZHICONG HE,³ LIJIE WANG,^{1,2} PING CHEN,^{1,2}
SICHUANG LIU,^{1,2} YONGZHENG XU,^{1,2} WEIJIA BAO,^{1,2}
AND YIPING WANG^{1,2,*} 

¹State Key Laboratory of Radio Frequency Heterogeneous Integration, Key Laboratory of Optoelectronic Devices and Systems of Ministry of Education/Guangdong Province, College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China

²Shenzhen Key Laboratory of Photonic Devices and Sensing Systems for Internet of Things, Guangdong and Hong Kong Joint Research Centre for Optical Fibre Sensors, Shenzhen University, Shenzhen 518060, China

³School of Mechanical and Electrical Engineering, Hubei Key Laboratory of Intelligent Transportation Technology and Device, Hubei Polytechnic University, Huangshi 435003, China

*ypwang@szu.edu.cn

Abstract: We propose and demonstrate an FPGA-GPU heterogeneous real-time optical frequency domain reflectometry (OFDR) strain sensing system using streaming linear-interpolation resampling (SLIR). In the FPGA, a streaming FIR-Hilbert-based phase extraction scheme is used to generate uniformly spaced phase-triggered resampling points, and a DDR4-based large-point 2D-FFT architecture is implemented for distance-domain transformation. The strain demodulation is offloaded to the GPU for parallel acceleration. Compared with conventional double-edge zero-crossing resampling, the proposed SLIR method improves the distance-domain spectral contrast from about 10 dB to 15 dB at a sensing position of approximately 9.8 m. Far-end strain measurement is further demonstrated over a 38.7 m fiber. With pipelined FPGA-GPU processing, a frame output interval of approximately 80 ms is achieved, corresponding to a real-time measurement rate of about 12 Hz. The proposed system provides a feasible solution for real-time OFDR strain sensing with improved resampling quality and reduced FPGA resource burden.

© 2026 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

Optical frequency domain reflectometry (OFDR) has been widely applied in distributed strain sensing [1,2], temperature measurement [3,4], vibration measurement [5,6], and three-dimensional shape sensing [7,8], owing to its high spatial resolution, high accuracy, and continuous distributed measurement capability. However, OFDR demodulation requires wavelength-sweep nonlinearity correction, optical-frequency-linearized resampling, distance-domain transformation, and local strain demodulation [9]. For large sampling point numbers, the data volume and computational load significantly limit its application in high-accuracy and real-time sensing.

To improve the processing speed of OFDR demodulation, graphics processing unit (GPU)- and field-programmable gate array (FPGA)-based acceleration methods have been demonstrated. GPUs are well suited for repetitive and data-parallel OFDR demodulation operations, such as windowing, zero padding, and cross-correlation [10,11]. For example, Wang *et al.* reduced the total processing time of OFDR strain demodulation from seconds to milliseconds by GPU-based parallel computation, demonstrating the potential of GPU acceleration for real-time OFDR strain sensing [10]. Shan *et al.* further applied GPU parallel computation to OFDR-based shape

reconstruction and reduced the processing time by a factor of 8.4 under a zero-padding size of 4096, showing the effectiveness of GPU acceleration for computationally intensive OFDR demodulation tasks [11]. In contrast, FPGAs are well suited for high-speed data acquisition, streaming processing, and large-point Fourier transformation owing to their low latency and pipelined architecture [12,13]. Wang *et al.* reported an FPGA-based high-speed OFDR demodulation algorithm using 2D-FFT and frequency-domain cross-correlation, reducing the 2M-point FFT processing time from 5.3 s to 9.75 ms [12]. Liang *et al.* further demonstrated a fully FPGA-based real-time OFDR distributed strain sensing system with improved strain accuracy through zero padding [13].

Although GPU- and FPGA-based acceleration improves the demodulation speed, online sweep-nonlinearity compensation remains a key issue in real-time OFDR. Existing FPGA-based systems usually adopt zero-crossing resampling (ZCR) or external-clock sampling, which are hardware-friendly but can be limited by the number of effective resampling points, resulting in a restricted distance-domain range and degraded spectral quality. To improve the sweep-nonlinearity correction accuracy, several interpolation-based resampling methods have been investigated. Guo *et al.* improved the strain accuracy of OFDR by suppressing nonlinear sweep noise and random sweep-range deviation, achieving a strain accuracy of $\pm 0.51 \mu\epsilon$ at a sensing resolution of 5 mm [9]. They also proposed an equal-frequency resampling method, which reduced the spatial broadening at 10 m from 4.2 mm using ZCR to 0.4 mm [14]. Zhong *et al.* further proposed an instantaneous optical frequency resampling method, which recovered the correct fiber-end position when ZCR failed under a long-distance OFDR configuration [15]. Although these methods can effectively improve the distance-domain spectral quality, they generally rely on full-frame phase extraction, phase unwrapping, and interpolation-based resampling, making them more suitable for offline processing rather than FPGA-based streaming implementation. Therefore, a streaming interpolation-based resampling method that combines the correction accuracy of interpolation resampling with the online processing capability of FPGA is still needed for real-time OFDR strain sensing.

In this work, an FPGA-GPU heterogeneous real-time OFDR strain sensing system using streaming linear-interpolation resampling (SLIR) is proposed and demonstrated. The FPGA performs front-end data reception, SLIR-based optical-frequency-linearized resampling, and large-point 2D-FFT processing, while the GPU performs parallel strain demodulation. A 78th-order FIR Hilbert transformer is designed for streaming auxiliary-interferometer (AI) phase extraction, which is used as the frequency reference for resampling the main-interferometer (MI) signal. The proposed system is evaluated by comparing SLIR with double-edge zero-crossing resampling (DZCR), characterizing strain responses under different sampling point numbers, performing far-end strain measurement over a 38.7 m fiber, and analyzing FPGA resource utilization and processing time.

2. System architecture and principle

2.1. OFDR experimental setup

The experimental setup of the FPGA-GPU heterogeneous OFDR strain sensing system is shown in Fig. 1. A tunable laser source was used as the swept source, with a wavelength-sweeping speed of 5000 nm/s. A trigger pulse from the TLS was used to synchronize the FPGA-based data acquisition with the wavelength sweep. The acquisition board was equipped with an FPGA chip (XCKU115), an ADC chip (AD9656), and DDR4 memory, supporting a maximum sampling rate of 1 GSPS with a 16-bit resolution. The acquired data were transferred from the ADC to the FPGA through the JESD204B interface. The computations were performed on a workstation equipped with an Intel Core i7-12700 K CPU and an NVIDIA GeForce RTX 3070 Ti GPU with 8 GB of video memory.

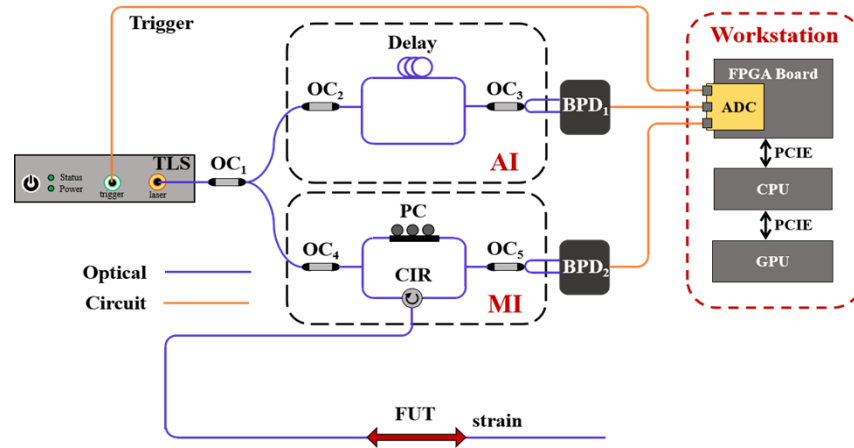


Fig. 1. Experimental setup of the FPGA-GPU heterogeneous OFDR strain sensing system. TLS: tunable laser source; OC: optical coupler; AI: auxiliary interferometer; MI: main interferometer; PC: polarization controller; CIR: circulator; BPD: balanced photodetector; FUT: fiber under test. The blue and orange lines denote the optical and electrical/control paths, respectively.

The output light from the TLS was divided into the AI and the MI by an optical coupler (OC_1). In the AI branch, a 40 m delay fiber was used to generate the auxiliary beat signal, which was detected by a balanced photodetector (BPD_1). This signal was used to extract the nonlinear wavelength-sweep information of the TLS for streaming linear-interpolation resampling (SLIR). In the MI branch, the probe light was launched into the fiber under test (FUT) through a circulator, and the Rayleigh backscattering (RBS) signal from the FUT was routed back and recombined with the polarization-adjusted reference light to generate the MI beat signal. The MI beat signal was detected by BPD_2 and synchronously acquired by the ADC. The acquired AI and MI signals were then processed by the FPGA-GPU heterogeneous architecture shown in Fig. 2.

2.2. FPGA-GPU heterogeneous OFDR strain-demodulation architecture

The FPGA-GPU heterogeneous processing architecture for real-time OFDR strain demodulation is shown in Fig. 2. As shown in Fig. 2(a), the FPGA is used for front-end streaming processing, including data reception, DDR4 buffering, SLIR, and DDR4-based large-point 2D-FFT processing. After digitization, the AI and MI data are received by the FPGA through the JESD204B receiver IP core and converted into parallel data streams. The parallel data are first buffered in the DDR4 module and then read out for SLIR processing. Based on the sweep-nonlinearity information provided by the AI signal, the MI signal is resampled to obtain uniformly spaced optical-frequency sampling points. The resampled MI signal is further cached and reordered by the DDR4 module for large-point 2D-FFT processing. Specifically, the data are first read out column-wise for FFT processing, followed by twiddle-factor multiplication and DDR4 buffering. Subsequently, the data are read out row-wise for the second-stage FFT, thereby completing the large-point FFT operation through the 2D-FFT architecture. The obtained distance-domain reference (Ref.) and measurement (Mea.) signals are then used for subsequent strain demodulation. Here, the Ref. and Mea. signals correspond to the MI signals acquired before and after strain loading, respectively.

Since the windowed strain-demodulation procedure consists of a large number of independent and repetitive operations, it is well suited for GPU-based parallel acceleration. As shown in Fig. 2(b), the distance-domain Ref. and Mea. signals are therefore transferred to the GPU through

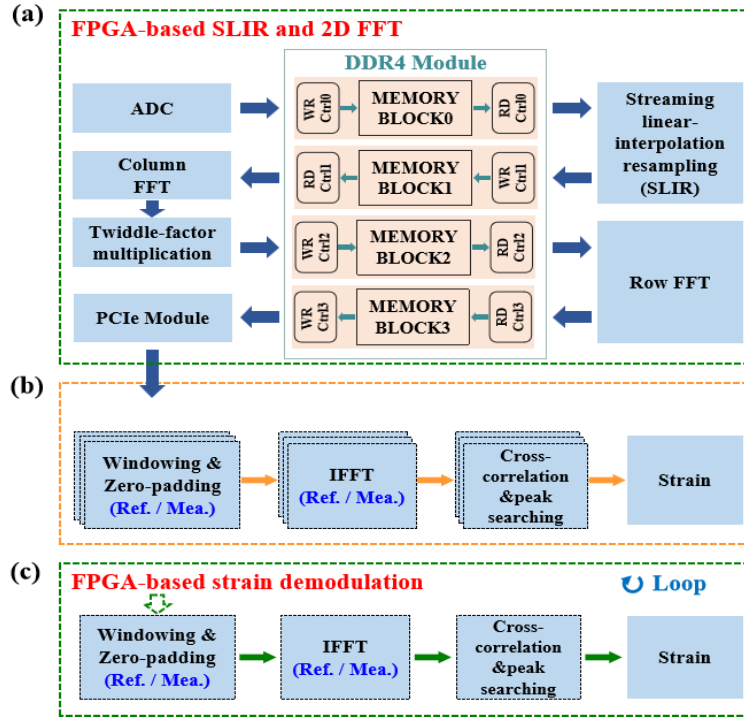


Fig. 2. Schematic of the FPGA-GPU heterogeneous real-time OFDR strain demodulation architecture using streaming linear-interpolation resampling (SLIR): (a) FPGA-based SLIR and 2D-FFT processing for real-time resampling and distance-domain transformation; (b) GPU-based parallel strain demodulation; (c) FPGA-based strain demodulation for resource-utilization comparison.

PCIe for parallel strain demodulation. The Ref. and Mea. signals are divided into multiple sensing windows with a window size of N sampling points. The corresponding physical window length, which determines the nominal sensing spatial resolution, is given by $Z = N\Delta z$, where $\Delta z = c/2n\Delta F$ [16], c is the speed of light in vacuum, n is the effective refractive index of the fiber, and ΔF is the optical frequency sweep range. Each N -point window is zero-padded to M points and then transformed by IFFT to reconstruct the local optical spectrum. The spectral shift between the Ref. and Mea. spectra is obtained by cross-correlation and peak searching, from which the strain distribution is demodulated.

For resource-utilization comparison, the same strain-demodulation procedure was also implemented on the FPGA, as shown in Fig. 2(c). In this path, windowing, zero padding, IFFT-based local spectrum reconstruction, cross-correlation, peak searching, and strain calculation are all mapped onto the FPGA and executed in a streaming loop. Note that FPGA-based implementation is used only to quantify the additional hardware resource overhead of the complete strain-demodulation procedure.

2.3. Principle of streaming linear-interpolation resampling

In the FPGA-based front-end processing described above, the principle of streaming linear-interpolation resampling (SLIR) is the key step for compensating the nonlinear wavelength sweep of the TLS before 2D-FFT processing. The principle of the proposed SLIR algorithm is shown in Fig. 3. As shown in Fig. 3(a), the AI signal is first processed by a Hilbert FIR filter for quadrature generation, and the obtained in-phase and quadrature components are then sent to a CORDIC

atan2 module to extract the wrapped phase, i.e., Φ_w [17]. Since the AI phase is related to the instantaneous optical frequency of the TLS, uniformly spaced AI-phase positions can be used as the target points for optical-frequency-linearized resampling. In the FPGA-based streaming implementation, these target positions are identified in real time by detecting the crossings of predefined phase thresholds. The trigger-generation module generates the corresponding resampling triggers, while the local phase-unwrapping module provides the locally unwrapped phase, i.e., Φ_{luw} , near the wrapping points for interpolation.

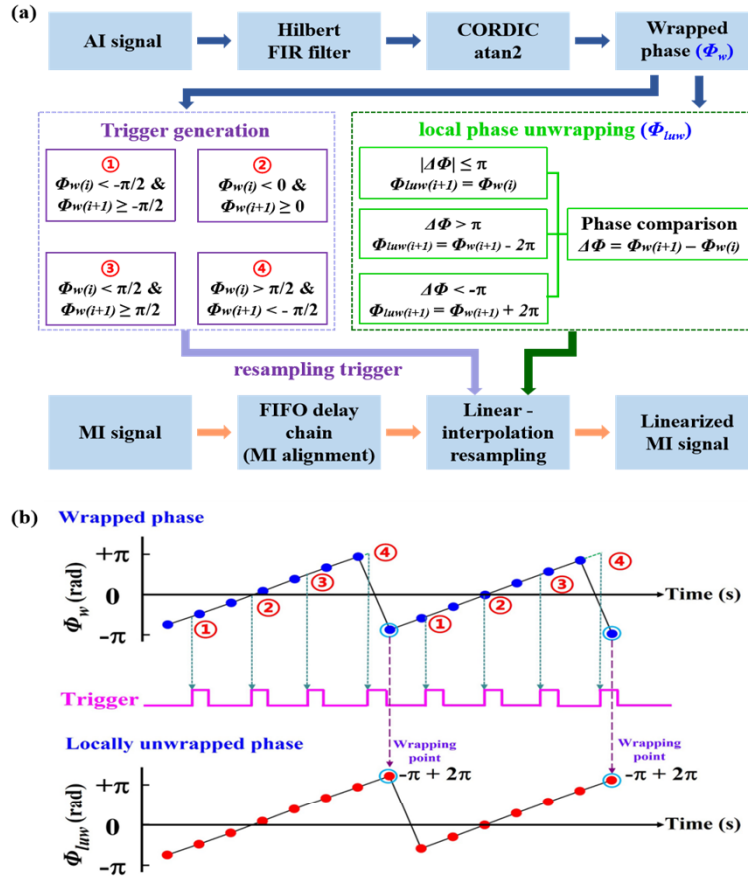


Fig. 3. Principle of the proposed streaming linear-interpolation resampling (SLIR) algorithm. (a) FPGA implementation flow of SLIR based on AI-signal phase extraction, trigger generation, local phase unwrapping, MI-signal delay compensation, and linear-interpolation resampling; (b) relationship among the wrapped phase, resampling trigger, and locally unwrapped phase.

Meanwhile, the MI signal is delayed by a FIFO chain to compensate for the processing latency of the AI branch. The delayed MI samples are aligned with the resampling triggers and then sent to the linear-interpolation module. Based on the locally unwrapped AI phase, the interpolation position between two adjacent MI samples is determined, and the corresponding resampled MI value is calculated. As a result, the MI signal is converted into an optical-frequency-linearized signal with uniformly spaced sampling points for subsequent 2D-FFT processing and strain demodulation.

The trigger-generation and local phase-unwrapping processes are further illustrated in Fig. 3(b). Four uniformly spaced phase thresholds, i.e., $-\pi/2$, 0 , $\pi/2$, and π ($-\pi$), are used for trigger

generation, as marked by ①-④ in Fig. 3(a). When two adjacent wrapped-phase samples, i.e., $\Phi_w(i)$ and $\Phi_w(i+1)$, cross one of these thresholds, a resampling trigger is generated. The corresponding trigger conditions are given by

$$\begin{cases} \Phi_w(i) < -\pi/2, \Phi_w(i+1) \geq -\pi/2, \\ \Phi_w(i) < 0, \Phi_w(i+1) \geq 0, \\ \Phi_w(i) < \pi/2, \Phi_w(i+1) \geq \pi/2, \\ \Phi_w(i) > \pi/2, \Phi_w(i+1) < -\pi/2. \end{cases} \quad (1)$$

The first three conditions correspond to the regular crossings of $-\pi/2$, 0 , $\pi/2$, respectively, while the fourth condition corresponds to the phase wrapping point from $+\pi$ to $-\pi$. To remove the discontinuity near the wrapping point, the phase difference between two adjacent wrapped-phase samples is calculated as

$$\Delta\Phi = \Phi_w(i+1) - \Phi_w(i). \quad (2)$$

If $|\Delta\Phi| \leq \pi$, no phase wrapping occurs, and the current phase sample is directly used. If $\Delta\Phi > \pi$, the current phase sample is corrected as

$$\Phi_{luw}(i+1) = \Phi_w(i+1) - 2\pi. \quad (3)$$

If $\Delta\Phi < -\pi$, the current phase sample is corrected as

$$\Phi_{luw}(i+1) = \Phi_w(i+1) + 2\pi. \quad (4)$$

The case shown in Fig. 3(b) corresponds to the phase jump from $+\pi$ to $-\pi$, where $\Delta\Phi < -\pi$, and the current wrapped phase sample is therefore corrected by adding 2π . In this way, the phase discontinuity near the wrapping point is locally removed without full-sequence phase unwrapping, making the proposed SLIR algorithm suitable for FPGA-based streaming implementation.

2.4. Streaming FIR Hilbert transformer for phase extraction

After introducing the SLIR principle, the performance of the Hilbert FIR filter used for AI-signal phase extraction was further evaluated. In conventional offline processing, the FFT-based Hilbert transform is commonly used to generate the analytic signal. However, it requires full-frame data storage and block-wise processing, which is not suitable for FPGA-based streaming implementation. Therefore, a 78th-order equiripple FIR Hilbert transformer was designed for streaming quadrature-signal generation, with a sampling rate of 250 MHz and a normalized passband of 0.05-0.95.

To evaluate the performance of the designed FIR Hilbert transformer, a 2M-point AI signal was processed using both the FFT-based Hilbert transform and the FIR Hilbert transformer. The obtained analytic signals were then used for arctangent phase extraction and phase unwrapping. As shown in Fig. 4(a), the unwrapped phase traces obtained by the two methods exhibit nearly identical linearity, with R^2 values of 0.99998. The phase difference between the two unwrapped phase traces is shown in Fig. 4(b), where the standard deviation is approximately 0.006 rad, and the maximum and minimum differences are about 0.04 rad and -0.04 rad, respectively. The zoomed-in phase-difference trace around the maximum-deviation region is shown in Fig. 4(c), indicating that the local phase deviation remains within a small range. In the same local region, the wrapped phase traces obtained by the two methods almost overlap, as shown in Fig. 4(d). These results indicate that the designed FIR Hilbert transformer provides sufficient quadrature-generation accuracy for the proposed SLIR algorithm. More importantly, it supports sample-by-sample streaming output with a fixed group delay of 39 clock cycles, avoiding the full-frame buffering required by the FFT-based Hilbert transform.

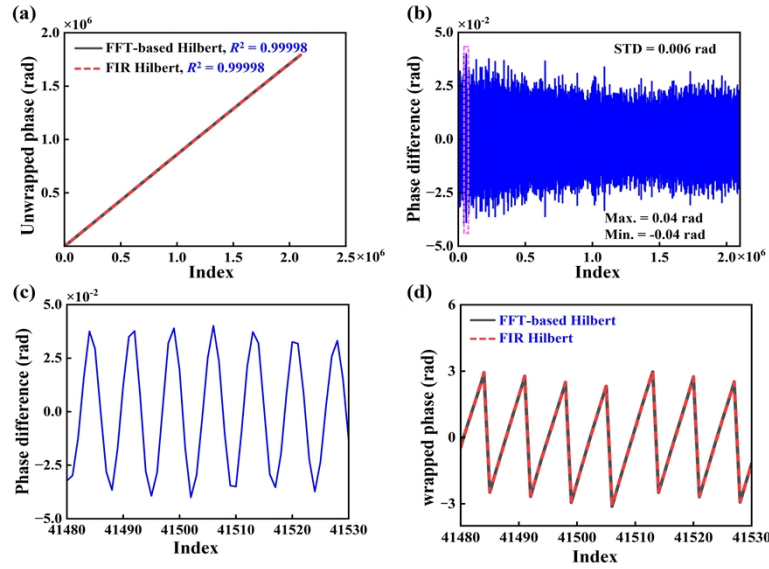


Fig. 4. Performance comparison between the FFT-based Hilbert transform and the designed FIR Hilbert transformer. (a) Unwrapped phase traces obtained from a 2M-point AI signal; (b) phase difference between the two unwrapped phase traces; (c) zoomed-in phase-difference trace around the maximum-deviation position; (d) corresponding wrapped phase traces in the same local region.

3. Experimental results and discussion

3.1. Performance comparison between DZCR and SLIR

After verifying the accuracy of the FIR Hilbert transformer for streaming phase-extraction, the overall performance of the proposed SLIR method was further evaluated by comparing it with the conventional double-edge zero-crossing resampling (DZCR) method. The wavelength sweep range was fixed at 20 nm, and the sensing-window length was set to $N = 256$ for both methods to ensure the same nominal gauge length for local strain demodulation. Figures 5(a) and 5(b) show the distance-domain spectra obtained using DZCR and SLIR at the sensing position of approximately 9.8 m, respectively. At a distance-domain sampling interval of 38.15 μm , the fiber-end reflection peaks obtained using DZCR and SLIR exhibited nearly identical full width at half maximums (FWHMs) of approximately 29.8 μm , indicating comparable spatial resolutions, as shown in the insets of Figs. 5(a) and 5(b). The reason was that the spatial resolution was determined by the same optical-frequency sweep bandwidth rather than by the number of resampling points. Moreover, the spectral contrast is increased from about 10 dB for DZCR to approximately 15 dB for SLIR, indicating improved distance-domain signal quality after optical-frequency-linearized resampling. Unlike DZCR, which uses discrete ADC samples adjacent to the zero crossings as resampling points [14], SLIR estimates the target positions between adjacent samples through local phase unwrapping and linear interpolation of the extracted AI phase. This enables sub-sample optical-frequency resampling, reducing residual sidelobes and the background noise of the MI signal and thereby improving the spectral contrast. For a direct comparison within the same displayed distance range, only the first 20 m of the SLIR result is shown in Fig. 5(b).

The corresponding zero-strain demodulation results are shown in Figs. 5(d) and 5(e). For DZCR, the strain standard deviations are 1.59 $\mu\epsilon$ and 1.51 $\mu\epsilon$ with zero-padding lengths of $M = 8192$ and $M = 16384$, respectively. In comparison, SLIR reduces the strain fluctuations

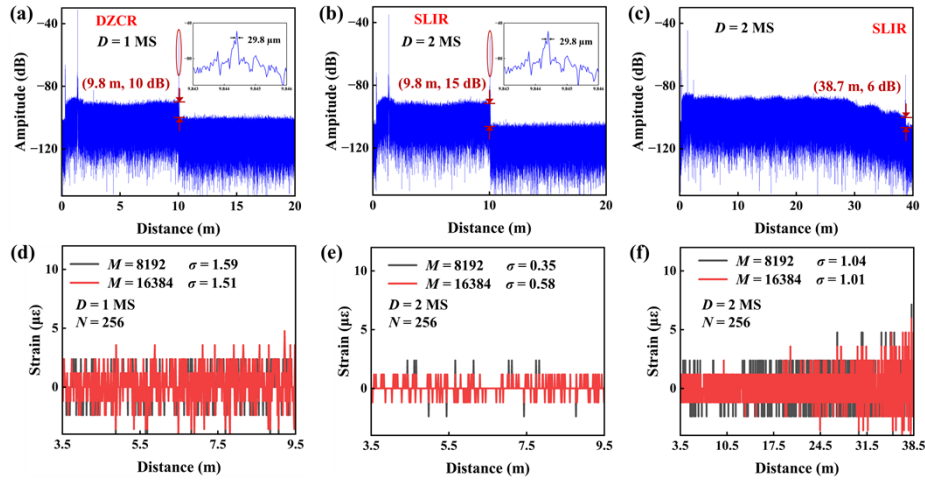


Fig. 5. Performance comparison of double-edge zero-crossing resampling (DZCR) and SLIR. (a-c) Distance-domain spectra and (d-f) corresponding zero-strain demodulation results. (a, d) DZCR results at a sensing distance of approximately 9.8 m with ($D = 1$ MS; (b, e) SLIR results at approximately 9.8 m with ($D = 2$ MS; and (c, f) SLIR results at 38.7 m with $D = 2$ MS. Only the first 20 m is displayed in (b) for direct comparison with (a). The insets in (a) and (b) show the enlarged fiber-end reflection peaks, both with an FWHM of approximately 29.8 μm .

to 0.35 $\mu\text{ε}$ and 0.58 $\mu\text{ε}$, respectively, demonstrating better demodulation stability under the same sensing-length window. Moreover, the long-range measurement capability of SLIR was evaluated using an approximately 38.7 m-long sensing fiber, as shown in Fig. 5(c). Although the distance-domain spectral contrast near the fiber end decreases to about 6 dB, the zero-strain demodulation results in Fig. 5(f) still show standard deviations of 1.04 $\mu\text{ε}$ and 1.01 $\mu\text{ε}$ for $M = 8192$ and $M = 16384$, respectively. Here, $D = 1$ MS and $D = 2$ MS were used for DZCR and SLIR, respectively, because SLIR provides more optical-frequency-linearized sampling points within the same sweep range, thereby extending the available distance-domain range.

3.2. Strain measurement under different sampling point numbers

To further verify the strain demodulation capability of the proposed SLIR method, the strain response was characterized under different sampling point numbers, as shown in Fig. 6. Strain measurements were performed on an approximately 50 cm-long fiber section with sampling point numbers of $D = 2$ MS and $D = 4$ MS. For the two cases, the sensing-window lengths were set to $N = 256$ and $N = 512$, respectively, while the zero-padding number was fixed at $M = 16384$. In this way, the same physical window size of 9.77 mm was maintained for the two measurements. Without applying any additional filtering, stable strain demodulation was achieved from 1000 to 8000 $\mu\text{ε}$ for $D = 2$ MS and from 1000 to 9000 $\mu\text{ε}$ for $D = 4$ MS. Local distortion appeared at 8500 $\mu\text{ε}$ and 9500 $\mu\text{ε}$, respectively, indicating that increasing the sampling point number improves the stability of high-strain demodulation under the same physical sensing-window size.

To further verify the far-end strain measurement capability of the proposed SLIR, strain was applied to an approximately 50 cm-long section near the end of a 38.7 m-long fiber. As shown in Fig. 7, the number of sampling points was varied from $D = 0.5$ MS to $D = 4$ MS, while the sensing-window length and zero-padding number were fixed at $N = 256$ and $M = 16384$, respectively. As shown in Figs. 7(a)–7(d), the strained section located around 37.4–38.0 m can be clearly identified in all cases, and stable strain responses are obtained from 500 to 2500 $\mu\text{ε}$. As

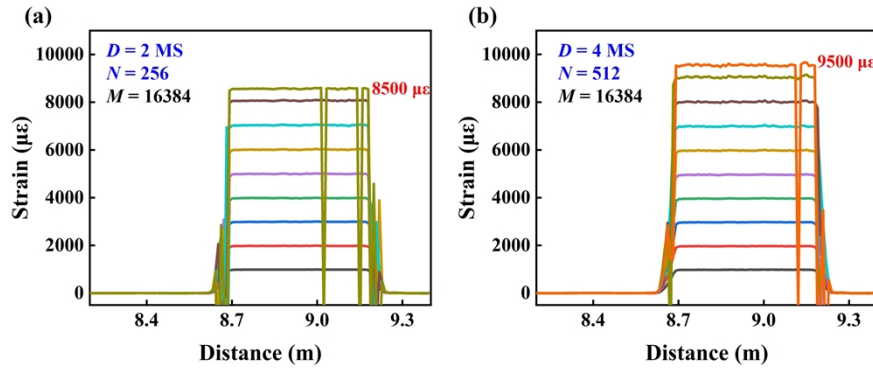


Fig. 6. Strain response of the proposed SLIR-based OFDR system under different sampling point numbers. (a) Demodulated strain profiles obtained with $D = 2$ MS, $N = 256$, and $M = 16384$; (b) demodulated strain profiles obtained with $D = 4$ MS, $N = 512$, and $M = 16384$. The sensing-window lengths were selected to maintain the same physical window size of 9.77 mm in the two cases.

the sampling point number increases, the transition regions at the two ends of the strained section become more distinguishable, indicating that a larger sampling point number helps improve the visibility of local strain details. The wavelength sweep ranges were 5, 10, 20, and 40 nm for $D = 0.5, 1, 2$, and 4 MS, respectively.

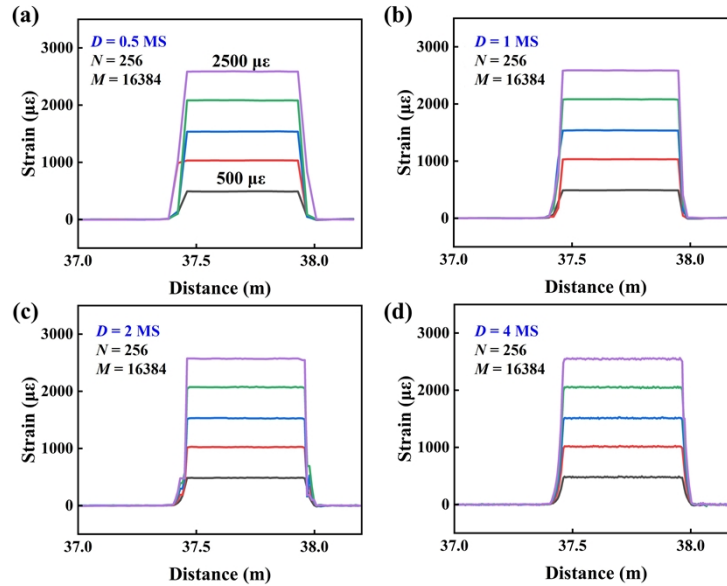


Fig. 7. Far-end strain measurement results obtained from a 38.7 m-long fiber under different sampling point numbers. (a) $D = 0.5$ MS, (b) $D = 1$ MS, (c) $D = 2$ MS, and (d) $D = 4$ MS. The strained section was approximately 50 cm long and located near the fiber end. The sensing-window length and zero-padding number were fixed at $N = 256$ and $M = 16384$, respectively.

3.3. FPGA resource utilization and processing time

The FPGA resource utilization of the FPGA-based SLIR and 2D-FFT processing was further analyzed under different sampling point numbers, as shown in Fig. 8. As D increases from 0.5 MS to 4 MS, the utilization of look-up table (LUT), LUT-based random-access memory (LUTRAM), flip-flop (FF), and digital signal processing (DSP) remains relatively stable. Specifically, the LUT utilization varies only slightly from 18.82% to 18.70%, the FF utilization remains around 14%, and the DSP utilization stays below 2.21%. This indicates that the main control logic and arithmetic resources of the SLIR and 2D-FFT modules do not increase significantly with the sampling point number. In contrast, the block random-access memory (BRAM) utilization increases evidently with D . As shown in Figs. 8(a)–8(d), the BRAM utilization increases from 21.60% to 24.65%, 31.02%, and 40.37% for $D = 0.5$ MS, 1 MS, 2 MS, and 4 MS, respectively, attributing to the larger buffering requirement for data caching, reordering, and 2D-FFT processing at larger data scales.

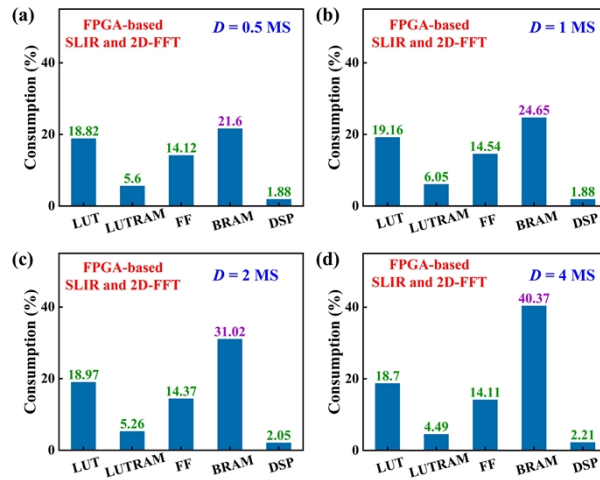


Fig. 8. FPGA resource utilization of the FPGA-based SLIR and 2D-FFT processing under different sampling point numbers: (a) $D = 0.5$ MS, (b) $D = 1$ MS, (c) $D = 2$ MS, and (d) $D = 4$ MS.

To further evaluate the hardware cost of implementing the complete strain-demodulation procedure on the FPGA, the resource utilization was analyzed under $D = 2$ MS, $N = 256$, and $M = 8192$, as shown in Fig. 9. After adding the FPGA-based strain-demodulation implementation, the total utilization of LUT, LUTRAM, FF, BRAM, and DSP increase to 30.28%, 11.30%, 23.85%, 75.28%, and 9.58%, respectively. Among these resources, BRAM shows the most significant increase. The FPGA-based strain-demodulation implementation introduces an additional BRAM of 44.26%, increasing the total BRAM utilization from 31.02% to 75.28%. This is mainly caused by the buffering requirement of windowing, zero-padding, IFFT, cross-correlation, and peak searching. These results indicate that BRAM becomes the main bottleneck when the complete strain-demodulation procedure is implemented on the FPGA. Therefore, the adopted FPGA-GPU architecture offloads strain demodulation to the GPU, while retaining SLIR and 2D-FFT processing on the FPGA.

To further evaluate the real-time processing capability of the proposed FPGA-GPU heterogeneous system, the processing time was analyzed, as shown in Fig. 10. As D increases from 0.5 MS to 1 MS, 2 MS, and 4 MS, the processing time of FPGA-based SLIR and 2D-FFT processing increases from 20.5 ms to 42.7 ms, 77.2 ms, and 164.4 ms, respectively. These results show that the FPGA front-end processing time increases with the number of sampling points.

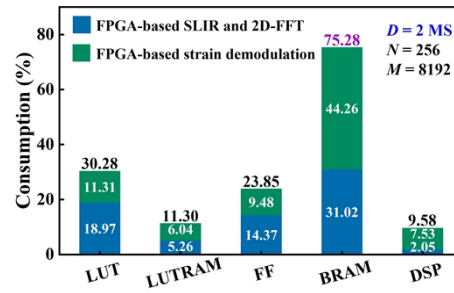


Fig. 9. FPGA resource utilization before and after implementing the strain-demodulation procedure on the FPGA under $D = 2$ MS, $N = 256$, and $M = 8192$. The blue bars represent the resource utilization of the FPGA-based SLIR and 2D-FFT processing, and the green bars represent the additional utilization introduced by the FPGA-based strain-demodulation implementation.

The processing time of the FPGA-only and FPGA-GPU heterogeneous implementations at different sensing distances was compared, under $D = 2$ MS, $N = 256$, and $M = 8192$. Because the FPGA-based SLIR and 2D-FFT stages always process the complete 2 MS data frame, their processing time remains constant at 77.2 ms. As the processed sensing distance increases from 10 to 38.7 m, the single-frame end-to-end processing time increases from 91.3 to 129.3 ms for the FPGA-GPU implementation, compared with an increase from 140.9 to 331.3 ms for the FPGA-only implementation. With the sensing-window length and zero-padding length fixed, the number of sensing windows increases approximately in proportion to the processed sensing distance, resulting in an approximately linear increase in strain-demodulation time. Note that Fig. 10(b) shows the single-frame end-to-end processing latency. In the FPGA-GPU heterogeneous implementation, the FPGA and GPU can process different frames in a pipelined manner. At 38.7 m, the GPU-based strain-demodulation time of 52.1 ms remains shorter than the FPGA processing time of 77.2 ms. Therefore, the frame output interval is approximately 80 ms, corresponding to a real-time measurement rate of about 12 Hz.

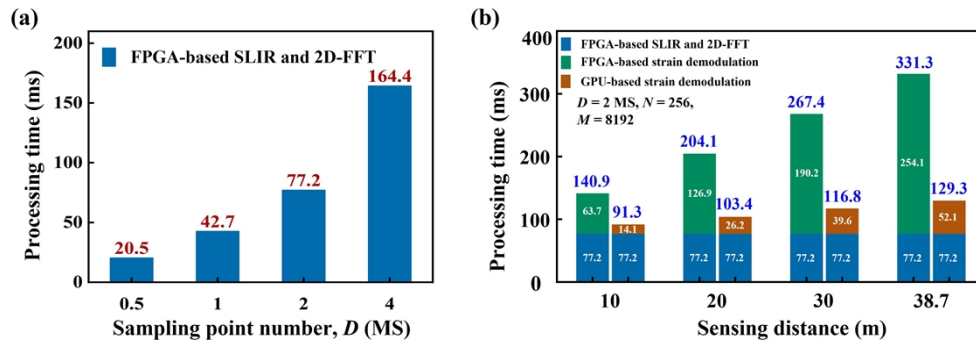


Fig. 10. Processing-time evaluation of the proposed FPGA-GPU heterogeneous OFDR demodulation system: (a) processing time of the FPGA-based SLIR and 2D-FFT stages at different numbers of sampling points; (b) comparison of the single-frame processing times for the FPGA-only and FPGA-GPU heterogeneous implementations at different processed sensing distances, with $D = 2$ MS, $N = 256$, and $M = 8192$.

4. Conclusion

In conclusion, an FPGA-GPU heterogeneous real-time OFDR strain sensing system based on SLIR was proposed and experimentally demonstrated. The FPGA was used for streaming resampling and DDR4-based large-point 2D-FFT processing, while the GPU was used for parallel strain demodulation. A 78th-order FIR Hilbert transformer was designed for streaming AI-signal phase extraction, showing a phase-difference standard deviation of approximately 0.006 rad compared with the FFT-based Hilbert transform. Compared with DZCR, SLIR improved the distance-domain spectral contrast from about 10 dB to 15 dB at approximately 9.8 m and reduced the zero-strain standard deviation from 1.59 $\mu\epsilon$ to 0.35 $\mu\epsilon$ for $M = 8192$ under the same sensing-window length. Far-end strain measurement was further demonstrated over a 38.7 m fiber. When the complete strain-demodulation procedure was implemented on the FPGA, the BRAM utilization increased from 31.02% to 75.28%, indicating a memory-resource bottleneck. By offloading strain demodulation to the GPU, the demodulation time was reduced from 254.1 ms to 52.1 ms, and a pipelined frame output interval of approximately 80 ms was achieved, corresponding to a real-time measurement rate of about 12 Hz. The proposed FPGA-GPU heterogeneous SLIR-based OFDR system therefore provides an effective solution for real-time distributed strain sensing with improved resampling quality and reduced FPGA resource burden.

Funding. National Key Research and Development Program of China (2023YFF0715700); National Natural Science Foundation of China (U22A2088, U25B2051, 62375178, 62405196); Shenzhen Science and Technology Program (JCYJ20241202124226032, SYSPG20241211173908021); Lingchuang Research Project of China National Nuclear Corporation (CNNC-LCKY-202265); Key Project of Hubei Province's Education Science Planning in 2025 (2025GA035); Fellowship Program of China Postdoctoral Science Foundation (GZC20260455).

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. J. Knebusch, J. Gundlach, D. Meier, *et al.*, "Static testing using OFDR-based quasi continuous fiber optic strain sensing on the WiValdi wind turbine rotor blade," *J. Phys.: Conf. Ser.* **2767**(4), 042005 (2024).
2. J. Shi, Z. Wang, Y. Zhang, *et al.*, "Full-length strain and damage monitoring for carbon fiber reinforced polymer cable based on optical frequency domain reflectometry," *Mechanical Systems and Signal Processing* **219**, 111627 (2024).
3. R. Zahoor, R. Vallifuoco, L. Zeni, *et al.*, "Distributed temperature sensing through network analysis frequency-domain reflectometry," *Sensors* **24**(7), 2378 (2024).
4. S.-O. Yang, S. Lee, S. H. Song, *et al.*, "Development of a distributed optical thermometry technique for battery cells," *Int. J. Heat Mass Transfer* **194**, 123020 (2022).
5. Y. Meng, W. Xie, Y. Feng, *et al.*, "Quantitative quasi-distributed vibration sensing by φ -OFDR for multiple events over spatially consecutive sensing spatial resolutions," *Opt. Express* **31**(1), 211–226 (2023).
6. Q. Yang, W. Xie, C. Wang, *et al.*, "Dynamic strain sensing using Doppler-shift-immune phase-sensitive OFDR with ultra-weak reflection array and frequency-tracking," *Opt. Express* **32**(25), 44816–44828 (2024).
7. H. Zhai, Q. Wang, B. Zhou, *et al.*, "Advances in fiber-optic-based 3D shape sensing technology," *Opt. Laser Technol.* **190**, 113190 (2025).
8. C. Fu, W. Liang, S. Li, *et al.*, "FPGA-based optical frequency domain reflectometry for real-time fiber shape sensing," *J. Lightwave Technol.* **44**(9), 3755–3760 (2026).
9. Z. Guo, J. Yan, G. Han, *et al.*, "High sensing accuracy realisation with millimetre/sub-millimetre resolution in optical frequency domain reflectometer," *J. Lightwave Technol.* **40**(12), 4050–4056 (2022).
10. C. Wang, K. Liu, Z. Ding, *et al.*, "GPU-based real-time distributed dynamic strain sensing in optical frequency domain reflectometry," *IEEE Sensors J.* **21**(21), 24166–24176 (2021).
11. R. Shan, C. Fu, Y. Meng, *et al.*, "High-spatial-resolution 2D shape sensing based on OFDR using adaptive spectrum method with narrow sweep range," *Opt. Express* **33**(14), 29626–29634 (2025).
12. H. Wang, T. Zhai, Y. Wang, *et al.*, "Real-time sensing approach for optical frequency domain reflectometry using an FPGA-based high-speed demodulation algorithm," *Opt. Express* **32**(19), 33247–33261 (2024).
13. W. Liang, C. Fu, Y. Meng, *et al.*, "Real-time distributed strain sensing using FPGA-based optical frequency domain reflectometry," *IEEE Trans. Instrum. Meas.* **74**, 1–8 (2025).
14. Z. Guo, G. Han, J. Yan, *et al.*, "Ultimate spatial resolution realisation in optical frequency domain reflectometry with equal frequency resampling," *Sensors* **21**(14), 4632 (2021).

15. H. Zhong, C. Fu, P. Li, *et al.*, “Distributed high-temperature sensing based on optical frequency domain reflectometry with a standard single-mode fiber,” *Opt. Lett.* **47**(4), 882–885 (2022).
16. B. J. Soller, D. K. Gifford, M. S. Wolfe, *et al.*, “High resolution optical frequency domain reflectometry for characterization of components and assemblies,” *Opt. Express* **13**(2), 666–674 (2005).
17. S. Du, T. Chen, C. Guo, *et al.*, “Hardware-efficient phase demodulation for digital ϕ -OTDR receivers with baseband and analytic signal processing,” *Sensors* **25**(10), 3218 (2025).