IET Circuits, Devices & Systems

IET The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

A hardware prototype of wideband high-dynamic range analog-to-digital converter

Satish Mulleti[1] | Eliya Reznitskiy[2] | Shlomi Savariego[2] | Moshe Namer[2] |
Nimrod Glazer[2] | Yonina C. Eldar[2]

[1]Department of Electrical Engineering, Indian Institute of Technology (IIT) Bombay, Mumbai, India

[2]Faculty of Math and Computer Science, Weizmann Institute of Science, Rehovot, Israel

Correspondence

Satish Mulleti.
Email: mulleti.satish@gmail.com

Abstract

Key parameters of analog-to-digital converters (ADCs) are their sampling rate and dynamic range. Power consumption and cost of an ADC are directly proportional to the sampling rate; hence, it is desirable to keep it as low as possible. The dynamic range of an ADC also plays an important role, and ideally, it should be greater than the signal's; otherwise, the signal will be clipped. To avoid clipping, modulo folding can be used before sampling, followed by an unfolding algorithm to recover the true signal. Here, the authors present a modulo hardware prototype that can be used before sampling to avoid clipping. The authors' modulo hardware operates prior to the sampling mechanism and can fold higher frequency signals compared to existing hardware. The authors present a detailed design of the hardware and also address key issues that arise during implementation. In terms of applications, the authors show the reconstruction of finite-rate-of-innovation signals, which are beyond the dynamic range of the ADC. The authors' system operates at six times below the Nyquist rate of the signal and can accommodate eight times larger signals than the ADC's dynamic range.

KEYWORDS

analog-to-digital conversion, automatic gain control, sample and hold circuits, sampling methods, signal reconstruction, signal sampling

1 | INTRODUCTION

Analog-to-digital converters (ADCs) bridge real-world analog signals and digital processors on which signals can be processed efficiently. Typically, ADCs measure instantaneous uniform samples of analog signals to represent them digitally. A key parameter in such conversion is the sampling rate. Power consumption and cost of an ADC increase with the increase in the sampling rate. Hence, keeping the sampling rate as low as possible is desirable. Theoretically, the sampling rate has to be greater than the Nyquist rate for perfect reconstruction of bandlimited signals. Apart from the sampling rate, there are several other aspects of an ADC which play a key role in faithful sampling and reconstruction, especially when the sampling frameworks are implemented in hardware.

The dynamic range of an ADC plays a crucial role in sampling an analog signal. Generally, ADC's dynamic range should be larger than the signal's; otherwise, the signal gets clipped. A few approaches exist to recover the true samples from clipped ones for bandlimited signals [1, 2]. These approaches rely on the correlation among the samples when they are measured at a very high rate compared to the Nyquist rate. The requirement of a high sampling rate is a drawback of these approaches.

Several preprocessing approaches to avoid clipping exist, such as automatic gain control (AGC) [3, 4], companding [5, 6], and modulo folding [7–11]. Among these, modulo folding is the most recent approach that need not be differentiable like companding and does not suffer from stability issues of the feedback amplifiers used in AGCs. In the modulo framework, the signal is folded to lie within the

ADC's dynamic range, and then the folded signal is sampled using a conventional ADC. Theoretical guarantees for recovering bandlimited signals from folded samples are presented in Ref. [7]. The results state that a bandlimited signal can be uniquely recovered from its folded samples provided that they are sampled above the Nyquist rate [7].

Several algorithms for *unfolding* or recovering the true samples of a bandlimited signal from modulo or folded samples are presented in Refs [7, 8, 10]. These unfolding algorithms can be compared in terms of sampling rate, amount of unfolding they can handle, and noise robustness. The algorithm proposed in Ref. [7] requires almost 17 times higher sampling rate than the Nyquist rate. The approaches in Refs [8] and [10, 11] operate at relatively lower sampling rates but require knowledge of the ADC's dynamic range. In contrast, the method proposed in Refs [10, 11] requires a lower sampling rate, even in the presence of noise, compared to the algorithms presented in Refs [7, 8]. Modulo sampling is also extended to different problems and signal models such as periodic bandlimited signals [9], finite-rate-of-innovation (FRI) signals [12], sparse vector recovery [13], direction of arrival estimation [14], computed tomography [15], and graph signals [16].

Beyond theoretical works, there also exist a few related hardware prototypes. High-dynamic-range ADCs, also known as *self-reset* ADCs, are discussed in the context of imaging [17–20]. These hardware architectures measure additional information, such as the amount of folding for each sample or the sign of the folding together with the folded samples. The additional information might enable simpler recovery at the expense of complex circuitry. Importantly, additional bits are required during the quantisation process to store or transmit the side information.

Krishna et al. presented a hardware prototype that encodes the side information by using two bits [20]. The architecture is designed to record the sign of the slope of the signal at each sample that lies outside the ADC's dynamic range. In a conventional ADC, a sample and hold (S/H) circuit is used to hold the sampled value for a prescribed period of time, during which quantisation is performed on the sample. A folding circuit is used after S/H to realise the modulo sampling [20]. In this architecture, the S/H circuit has to hold the sampled value for folding and quantisation, resulting in a larger holding time than a conventional ADC. A large holding time results in slower ADCs, which may not be helpful in applications with high-frequency signals. The resulting hardware circuit is able to fold signals up to 300 Hz, where the signal's amplitude should be less than three times the ADC's dynamic range.

Modulo hardware prototypes where the modulo part is implemented prior to the sampler are presented in Refs [9, 21, 22]. In these works, the authors are focused on different signal models, hardware limitations, and algorithms rather than providing details of the hardware circuitry. It was shown that the modulo hardware is able to fold low-frequency (<300 Hz) signals that are 10-fold larger than the ADC's dynamic range. However, it is not clear how the hardware performs for high-frequency signals, and many details of the circuitry are omitted.

In practical applications, the frequency range of the signals may vary from a few kHz to several MHz. For example, the FRI model is widely used to represent signals in time-of-flight applications such as ultrasound, sonar, and radar [23, 24]. These FRI signals have frequencies much higher than 300 Hz, and hence current hardware prototypes cannot be used, especially when the signal's bandwidth ranges up to a few kHz. Hence, it is desirable to design a modulo sampler that operates at high frequencies while folding signals faithfully.

In this paper, we present a modulo hardware prototype for modulo sampling of signals up to 10 kHz. We show that by using our algorithm [10], it is able to reconstruct bandlimited and FRI signals faithfully. In the following, we present the contributions and the features of the proposed hardware system.

- We design our hardware components to fold signals up to 10 kHz. Existing hardware shows results for signals below 300 Hz.
- The hardware prototype is designed to perform folding prior to the sampler, unlike the hardware in Ref. [20], which operates in the hold part of the sampler. Thus, the suggested system can utilise faster ADCs with shorter hold times.
- In the proposed hardware prototype, modulo folding is realised through a feedback mechanism. At the time instants when the input signal goes beyond the ADC's dynamic range, a trigger signal is generated by using comparators. The trigger then activates a direct voltage generator that adds to the input signal to bring it within the dynamic range. This mechanism imposes a delay between the trigger time and the folding instance. We address this key issue of the hardware, which is not considered in previous works. By using the signal's smoothness and the feedback loop's time delay, we propose a hardware solution to avoid clipping that occurs due to the delay issue.
- The designed hardware prototype operates at a maximum voltage of 11.75 v. The limitation is largely due to the use of a 15 v subtractor or adder in the feedback loop, which enables a fast slew rate in the transitions of $\pm 2\lambda$. At high frequencies, these components cannot be used at voltage above 15 v. In addition, we used an ADC with dynamic range [−1.25, 1.25]. Hence, the hardware can fold signals which are eight times larger than the dynamic range of the signal.
- For demonstration, we consider sampling and reconstruction of bandlimited and FRI signals. For FRI signals, we use a lowpass sampling kernel prior to modulo folding. The filter removes unwanted information in the signal and allows sub-Nyquist sampling. Using our algorithm presented in Refs [10, 11], we show reconstruction of bandlimited signals from their folded samples measured through the hardware. The combination of the proposed hardware and low-rate algorithm is able to reconstruct the signals by using a low-dynamic range ADC. In particular, for FRI signals, we show that the FRI parameters can be estimated with sub-Nyquist samples by utilising the fact that our algorithm operates at the lowest possible rate.

The paper is organised as follows. In the next section, we discuss the signal model considered and the sampling and reconstruction framework in the presence of modulo hardware. In Section 3, we present the hardware system by explaining its working principle and discussing the components of the system. In Section 4, we show the hardware's signal folding and reconstruction abilities.

2 | SIGNAL MODEL AND SYSTEM DESCRIPTION

In this section, we consider modulo sampling for signals whose amplitudes lie beyond the bandwidth of the ADC's dynamic range. The class of signals that can be folded by modulo hardware can be very large; however, the recovery is limited by existing unfolding algorithms. For example, most unfolding algorithms are designed for bandlimited signals. Given this, we consider bandlimited signals as input to the modulo hardware and corresponding unfolding algorithms. We use a lowpass sampling kernel for FRI signals to make them bandlimited and, at the same time, reduce the sampling rate following the sub-Nyquist framework [24, 25].

Consider a ω_c-bandlimited signal $y(t)$ such that its Fourier transform $Y(\omega)$ vanishes outside the frequency interval $[-\omega_c, \omega_c]$. The signal can be perfectly reconstructed from its uniform samples measured at the Nyquist rate $\omega_{Nyq} = 2\omega_c$ rad/ses provided that the ADC's dynamic range is above the signal's dynamic range. Specifically, if the dynamic range of the ADC is $[-\lambda, \lambda]$ for some $\lambda > 0$ then it is assumed that $|y(t)| \leq \lambda$ for perfect reconstruction. If $|y(t)| > \lambda$, then the signal and its samples will be clipped, and perfect reconstruction is not guaranteed. In the latter scenario where $|y(t)| > \lambda$, one can either increase the dynamic range of the ADC or use pre-possessing to avoid clipping. We consider the later solution where the modulo operation $\mathcal{M}_\lambda(\cdot)$ is applied to the signal $y(t)$ to restrict its dynamic range to $[-\lambda, \lambda]$. The output of the modulo operator in response to input $y(t)$ is given as follows:

$$y_\lambda(t) = \mathcal{M}_\lambda(y(t)) = (y(t) + \lambda) \bmod 2\lambda - \lambda. \quad (1)$$

The folded signal $y_\lambda(t)$ is then sampled to get discrete measurements $y_\lambda(nT_s)$. Due to modulo folding, $y_\lambda(t)$ is no longer bandlimited. To recover $y(t)$ while sampling slightly above the Nyquist rate of the input, one first applies an unfolding algorithm to recover $y(nT_s)$ from $y_\lambda(nT_s)$ [10, 11, 26]. Then $y(t)$ is reconstructed from $y(nT_s)$ by assuming that the sampling is performed above the Nyquist rate.

A schematic of our modulo sampling and reconstruction framework is shown in Figure 1. It consists of a modulo-ADC followed by unfolding and reconstruction blocks. The modulo-ADC is comprised of a modulo-folding block followed by a conventional uniform sampler. The unfolding operation is implemented in the digital domain, and it should operate at the lowest possible sampling rate. To this end, we use the B^2R^2 algorithm for unfolding [10, 11], which samples efficiently compared to other algorithms for bandlimited signals. Low-

rate sampling and low-dynamic range requirements significantly reduce the power consumption and cost of the ADC.

Our objective is to demonstrate a robust hardware prototype of modulo ADC as discussed next.

3 | MODULO HARDWARE PROTOTYPE

In this section, we discuss the prototype of our modulo hardware. The modulo block's working principle and design will be discussed first, followed by its hardware implementation.

3.1 | Working principle of modulo block

The principle of computing $y_\lambda(t)$ from $y(t)$ is shown by the block diagram in Figure 2. The system comprises an adder S, a direct-voltage generator (DVG), and two comparators, Comp-1 and Comp-2. To understand the working flow, let us first assume that for some time instant t_1, we have that $|y(t)| < \lambda$ for all $t < t_1$. Hence $y_\lambda(t) = y(t)$ and $z(t) = 0$ for all $t < t_1$. At $t = t_1$, let $|y(t)|$ cross λ. If $y(t_1) > \lambda$, then Comp-1 triggers a positive value. Else if, $y(t_1) < \lambda$, Comp-2 triggers a negative value. The DVG is designed such that for each positive input value, its output signal level increases by -2λ, whereas, for a negative input value, it decreases its output voltage by 2λ. Hence, in the current example, DVG generates a signal $z(t) = \mathrm{sgn}(y_\lambda(t_1))2\lambda u(t - t_1)$ where $u(t)$ is the unit-step function. In this way, by adding or subtracting (using S) constant DC signals from $y(t)$ whenever it crosses the dynamic range $[-\lambda, \lambda]$, the amplitude levels of $y_\lambda(t)$ are kept within the ADC's dynamic range.

While the parts such as comparators Comp-1 and Comp-2 and adder S can be realised by using off-the-shelf components, DVG is a more involved system due to its feedback nature and requires careful design. Specifically, the feedback loop should follow changes in the input signal in the desired frequency and amplitude ranges. A detailed architecture of
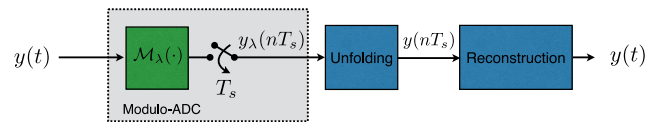

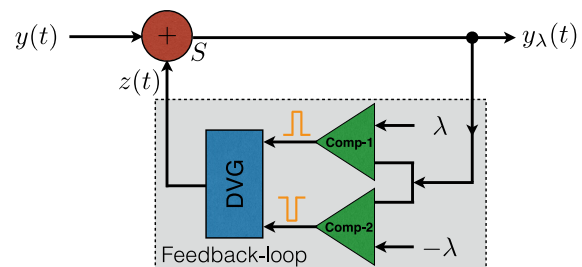**FIGURE 1** A schematic of modulo-sampling and reconstruction of bandlimited signals.

FIGURE 2 Folding principle.

DVG is shown in Figure 3. Its task is to generate a constant voltage signal whose amplitude is a constant multiple of 2λ. Importantly, its output voltage $z(t)$ should increase (or decrease) by 2λ v for every negative (or positive trigger) at its input. To realise this task in the hardware, we use an up/down counter, a digital-to-analog converter (DAC), a multiplexer (MUX), and a multiplier M.

In the hardware design, we set $\lambda = 1.25$ v. We start with the DAC, which can generate piecewise constant voltage output $s(t)$ in response to its digital input. Let the resolution, or step size of the DAC, be α v. Then, when the input bits of DAC go from one state to the next, the DAC output increases by α v. On the other hand, when the bits change from the present state to the previous state, output voltage $s(t)$ reduces by α v. Hence, the ADC works in a fashion expected by DVG with the following exceptions: (1) Input to the DAC is bits and one needs to map positive/negative trigger from comparators to these bits; (2) Output of the DAC takes only positive values and are multiple of α. A scaling is required to make them multiple of $\pm 2\lambda$. To address the first issue, we employ a UP/

DOWN counter whose inputs are the trigger voltages from the capacitors Comp-1 and Comp-2, and the output is bits. For every positive trigger at the input counter, output bits change to the next state, whereas for a negative trigger, they go back to the previous state. By connecting these bits to the input of the DAC, the output of the DAC is controlled by triggers.

To address the scaling issue, we use a MUX and a voltage multiplier M. The MUX and the multiplier are designed, together with a set of amplifiers, such that $s(t)$ is scaled to $z(t)$. A sign bit at the output of the counter, which is a function of the trigger's sign, is used as input to the MUX, which in turn controls the sign of the multiplier's output or $z(t)$'s sign.

To explain the sequences of events in DVG, let us consider our previous scenario where $|y(t)| < \lambda$ for some $t < t_1$ and at t_1, $|y(t)|$ crosses λ. For $t < t_1$, we have $y(t) = y_\lambda(t)$, $s(t) = 0$, $z(t) = 0$, and all the output bits of the counter are set to be zero. If $y(t_1) > \lambda$, Comp-1 triggers a positive voltage, and the counter's output bits state changes. Specifically, the least significant bit changes to one, and in response, the DAC's output voltage changes to α v. Meanwhile, after the positive trigger, the MUX outputs a voltage $-2\lambda/\alpha$ which is multiplied to $s(t)$ and outputs $z(t)$ as $-2\lambda u(t - t_1)$ as desired.

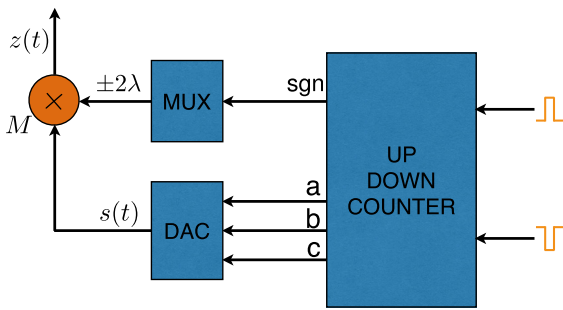Next, we discuss the hardware board that realises the folding operation discussed in this section.

3.2 | Modulo hardware board

Our modulo hardware board is presented in Figure 4, along with the roles of the major building components. Table 1 contains a detailed listing of the hardware's components. The board is designed for $\lambda = 1.25$ v. While selecting components for the MUX, multiplier, and amplifiers involved, we observe that these components operate in their linear regions if the
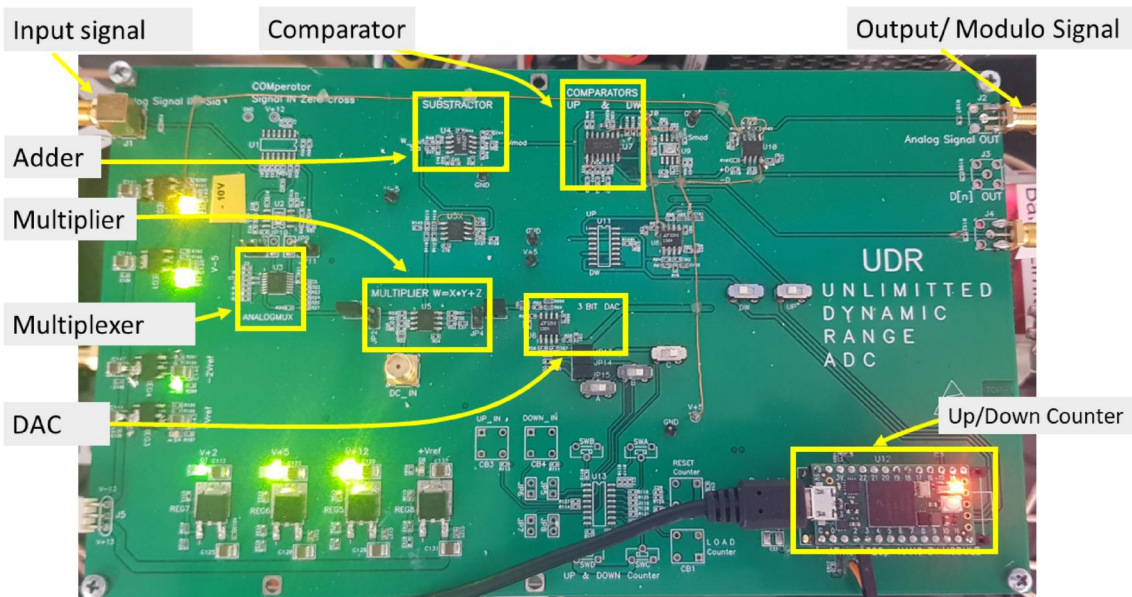

FIGURE 3 Discrete voltage generator.

FIGURE 4 Modulo hardware board.

operating voltages are <12 v. This implies that $|z(t)| \leq 12$ v which limits the maximum value of input signal to $|y(t)| < 9\lambda = 11.75$ v. This is because if $y(t)$ crosses 9λ then $z(t)$ should be $-10\lambda = 12.5$ v to ensure that $y_\lambda = y(t) + z(t) \in [-\lambda, \lambda]$. Hence, the current design of the hardware can fold and sample signals eight times larger than ADC's dynamic range. This implies that the DVG output should take values from the set $\{0, \pm 2\lambda, \pm 4\lambda, \pm 6\lambda, \pm 8\lambda\}$. This requires the DAC to have five uniform voltage levels at its output (it produces only positive voltages), and a 3-bit DAC and hence a 3-bit counter are used as shown in Figure 3. Instead of using an off-the-shelf DAC, we build a customised DAC for the hardware. By noting that the DAC's output is a linear combination of its three input bits, we used adders LT1364 to realise the DAC. In Table 2, we list the values of bits and the counter (denoted as counter values). The three bits (a, b, and c) of the counter are used as input to the DAC, which converts the bits to an analog DC voltage. Here the resolution of the DAC is $\alpha = 1$ v.

We further analyse the working of the modulo hardware by considering a sinusoidal signal $y(t) = A \sin(2\pi f_0 t)$ where A is amplitude, and f_0 is the frequency (in Hz). In this experiment, we set $\lambda = 1.25$. First, we analyse the folding ability of the hardware for different amplitude levels. Figure 5a,b depicts screenshots of an oscilloscope capturing input $y(t)$ (in yellow), folded output $y_\lambda(t)$ (in green), and the DVG output $z(t)$ (in blue), for two signals with $f_0 = 1$ kHz, $A = 4\lambda$ and $f_0 = 2$ kHz, $A = 8\lambda$, respectively. We observed that the signals are folded back to lie within the dynamic range of the ADC as expected without clipping.

Next, we discuss the frequency response of the modulo ADC. As in any analog system, the modulo folder's response also depends on the input signal's frequency or bandwidth. In particular, beyond a particular frequency range, components of the hardware and the overall feedback loop may not respond quickly to fast changes in the input signal, as demonstrated in

Figure 6. We observed that for 1 and 10 kHz, the hardware folds the signal accurately. However, for $f_0 = 20$ kHz, folding instants are not symmetric for positive and negative folds.

In the next section, we discuss several challenges of the modulo hardware and our proposed solutions.

3.3 | Artefacts during folding

Errors or other artifacts that arise during folding operations in hardware result from various reasons. One of the major issues that arise in a modulo ADC is the time delay in the feedback loop (see Figure 2). To elaborate, consider a scenario where $y(t) < \lambda$ for $t < t_1$ and it crosses λ at time t_1. To fold the output voltage to the dynamic range of the ADC, $z(t) = 2\lambda u(t - t_1)$ needs to be subtracted from $y(t)$. However, there is a finite delay between the trigger time t_1 to generating $z(t)$. If the time delay is T_d, then $z(t) = 2\lambda u(t - t_1 - T_d)$ is subtracted, which causes distortion. To illustrate this effect, in Figure 7, we considered a sinusoidal signal (in blue) and its folded versions with and without delay. We observe that in the absence of any time delay ($T_d = 0$) the signal folds perfectly (shown in red) to stay within the dynamic range. However, for a non-zero value of T_d, foldings do not take place at the folding instants, and the

TABLE 1 List of hardware components.

Component	Model number	Make
Comparator	LM339	Texas instruments
UP/DOWN counter	TEENSY4.1	PJRC
Analog MUX	ADG1608	Analog devices
Analog multiplier	AD835	Analog devices
Adder	LT1364	Analog devices

TABLE 2 Up/down counter operation.

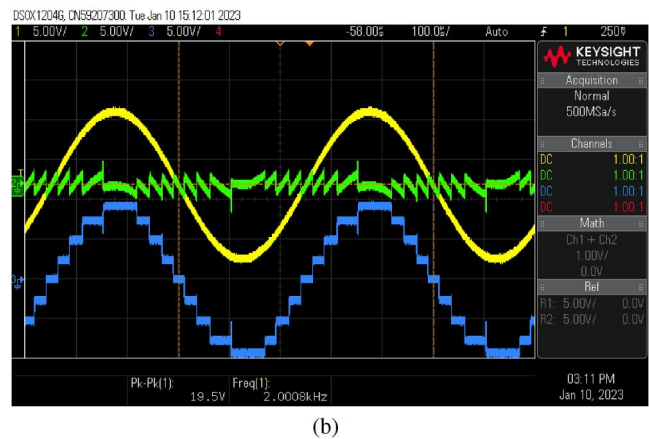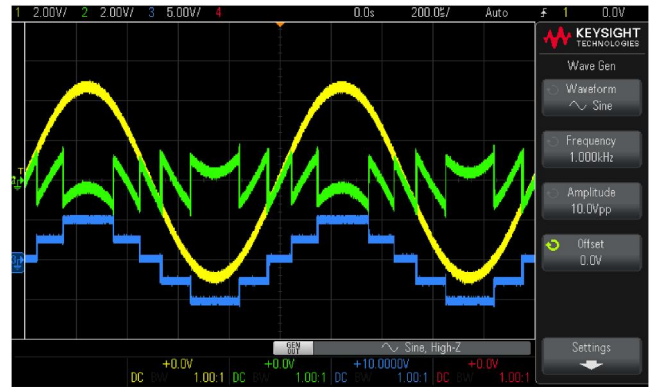| c	b	a	sgn	Counter values	$s(t)$	$z(t)$
0	0	0	0 (1)	0 (0)	0	0
0	0	1	0 (1)	1 (−1)	1	2λ (−2λ)
0	1	0	0 (1)	2 (−2)	2	4λ (−4λ)
0	1	1	0 (1)	3 (−3)	3	6λ (−6λ)
1	0	0	0 (1)	4 (−3)	4	8λ (−8λ)

(a)

(b)

FIGURE 5 Screenshots of an oscilloscope capturing input signals (yellow), its folded outputs (green), and the direct-voltage generator signals (blue): (a) 1 kHz sinusoid with maximum amplitude 4λ and (b) 2 kHz sinusoid with maximum amplitude 8λ.
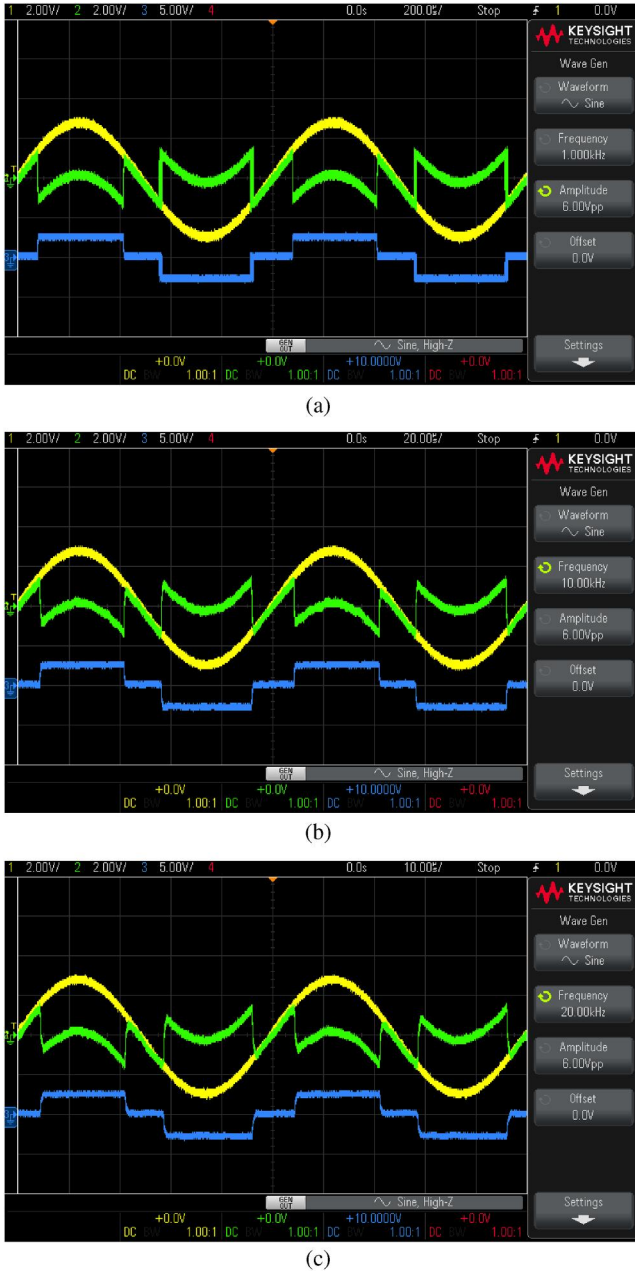
(a)

(b)

(c)

FIGURE 6 Analysis of the frequency response of the modulo hardware. (a–c) are real-time captures of an oscilloscope capturing the 1, 10, and 20 kHz, respectively, the sinusoidal input signal (yellow), its folded output (green), and the direct-voltage generator signal (blue).

output of the modulo operator (shown in black) still remains outside the dynamic range $[-\lambda, \lambda]$, which results in a clipped modulo operator output. The proposed hardware solution addresses this issue and avoids the undesired clipping as shown in Figures 5 and 6.

Our solution uses the fact that with a finite time delay, the amount of overshoot of a smooth signal can be bounded. To elaborate, assume that the signal that undergoes the modulo operation is Lipschitz continuous. Specifically, a signal $y(t)$ is Lipschitz continuous if there exists a positive real number L_y such that for any $T > 0$ we have the following equation:
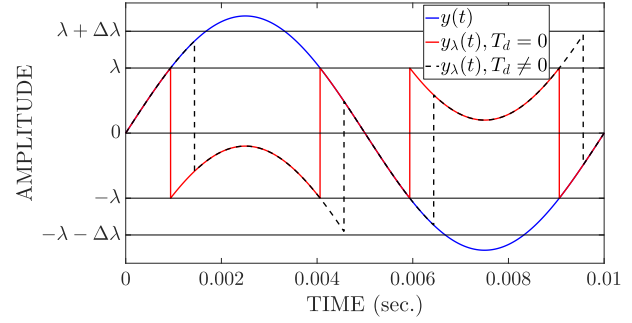

FIGURE 7 Effect of the time delay of the feedback loop on the modulo operation.

$$|y(t) - y(t + T)| \leq L_y T. \quad (2)$$

With the Lipschitz smoothness condition, we note that the amplitude of $y(t)$ cannot change more than $L_y T_d$ between any folding instant and time of its effect to take place. Hence, if we choose the dynamic range of the ADC to be $[-(\lambda + \Delta\lambda), (\lambda + \Delta\lambda)]$ where $\Delta\lambda = L_y T_d$ then the signal will not clip. We show this extended dynamic region in the example in Figure 7. Alternatively, instead of increasing the ADC's dynamic range, one can keep it to be $[-\lambda, \lambda]$ and reduce the threshold values for comparators. In this case, Comp-1 will trigger when the input crosses $\lambda - \Delta\lambda$, and Comp-2 will trigger when the input goes beyond $-\lambda + \Delta\lambda$. In this way, the time delay issue is addressed by the modulo circuit without altering ADC's dynamic range. In our hardware, we choose the former solution. Specifically, we used an oscilloscope to measure and display samples. The dynamic range of the oscilloscope's ADC was sufficiently higher than $[-\lambda, \lambda]$ to sample signals of interest without clipping signals due to the delay in the feedback loop.

In order to apply the solution, the signal must be Lipschitz continuous. In our design, the modulo operation input signal is always a bandlimited signal satisfying the Lipschitz smoothness condition [27]. For a bandlimited signal $y(t)$, its Lipschitz constant L_y is directly proportional to its bandwidth ω_c [27]. Hence, for a given value $\Delta\lambda$ and T_d (both depend on the modulo circuit), $L_y = \Delta\lambda/T_d$ is fixed and this restricts the maximum frequency of the input signal that can be faithfully folded. In the current design, we choose to implement the counter management using a TEENSY microcontroller, resulting in a 1 μs time delay. Then for a sinusoidal signal $y(t) = A \sin(2\pi f_0 t)$, the Lipschitz constant is given as $L_y = 2\pi A f_0$. For $A = 8\lambda$, $\Delta\lambda = 0.5\lambda$, and $T_d = 1$ μs, we note that the maximum operating frequency is 10 kHz which is in line with the experimental results discussed in Figure 6.

Specifications of the proposed hardware prototype are summarised in Table 3.

4 | RESULTS

In this section, we demonstrate the modulo hardware's signal reconstruction capability. We focus on the folding and reconstruction of bandlimited and FRI signals. In the hardware, the

T A B L E 3 Specifications of the modulo analog-to-digital converter.

Parameter	Value
Input signal	±10 v
Input bandwidth	10 kHz
Folding threshold (λ)	1.25 v
Modulo-input dynamic range	±10 v
Modulo-output dynamic range	±1.25 v

folded measurements are generally contaminated by different noises, including quantisation noise. Since the performance of an unfolding algorithm depends on the noise levels, we first consider simulated results to assess the performance of the B^2R^2 algorithm used for unfolding. Then we demonstrate the results from the hardware.

4.1 | Simulated results

In this section, we compare our B^2R^2 algorithm with the *higher-order differences* (HOD) approach [7, 28] and *Chebyshev polynomial filter*-based method [8]. Although a comparison of these methods is analysed in Ref. [10], the settings are different here. Importantly, quantisation noise is not considered in our previous work.

We consider noisy measurements as follows:

$$\tilde{y}_\lambda(nT_s) = y_\lambda(nT_s) + v(nT_s), \qquad (3)$$

where $v(nT_s)$ is the noise term. In the simulations, λ is set to be 1.25 as in the hardware. We normalise the bandlimited signals to have a maximum amplitude of 10. The SNR is calculated as $\text{SNR} = 20\log\left(\frac{\|y_\lambda(nT_s)\|}{\|v(nT_s)\|}\right)$. Reconstruction accuracies of different methods are compared in terms of the normalised mean-squared errors (MSEs) as $\frac{\sum |y(nT_s)-\hat{y}(nTs)|^2}{\sum |y(nT_s)|^2}$, where $\hat{y}(nT_s)$ is an estimate of $y(nT_s)$. For different noise settings and over-sampling factors (OFs), we considered 100 independent noise realisations and calculated the average MSE for them. We first treat quantisation noise and then present results for unbounded noise.

In the first simulation, the unfolding algorithms are applied to quantised folded samples. The MSE in the estimation of bandlimited signals for a different number of bits and OFs is shown in Figure 8. We observe that for a given OF, B^2R^2 algorithm results in the lowest MSE for <5 bits. For more than five bits, all the algorithms, except HOD with OF = 3, perform equally well. The results show that low-resolution quantisers can be used with the B^2R^2 algorithm for unfolding, which saves both power and memory requirements.

Next, for unbounded noise, we assume that the noise samples $v(nT_s)$ are independent and identically distributed Gaussian random variables with zero means. The variance of $v(nT_s)$ is set to achieve the desired SNR. We compare the
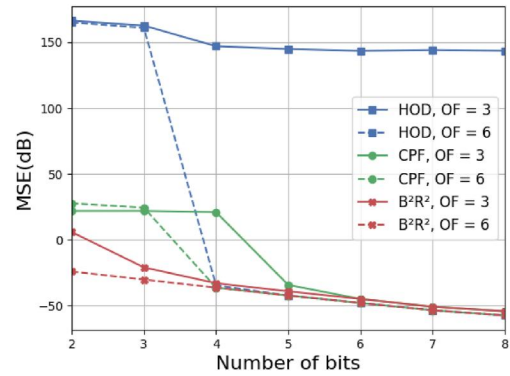

F I G U R E 8 Comparison of HOD, CPF, and B^2R^2 algorithms after the quantisation process in terms of MSE when recovering a bandlimited signal from modulo samples with $\lambda = 1.25$ and OF = 3,6. For a given number of bits, B^2R^2 has the lowest MSE. CPF, *Chebyshev polynomial filter*; HOD, *higher-order differences*; MSE, mean-squared error; OF, over-sampling factor.
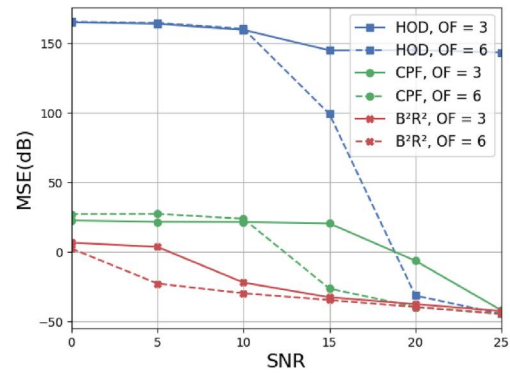

F I G U R E 9 Comparison of HOD, CPF and B^2R^2 algorithms (with unbounded noise) in terms of MSE when recovering a bandlimited signal from modulo samples with $\lambda = 1.25$ and OF = 3,6. For a given SNR and OF, B^2R^2 has the lowest MSE. CPF, *Chebyshev polynomial filter*; HOD, *higher-order differences*; MSE, mean-squared error; OF, over-sampling factor.

methods for different values of SNR and OFs with $\lambda = 1.25$. Figure 9 shows the MSE of the different algorithms for OF = 3 and 6. We note that our algorithm results in the lowest error for a given OF and SNR.

Given the advantages of the B^2R^2 algorithm over the other approaches, we present the hardware results in the next section by using this method.

4.2 | Hardware results for bandlimited signals

In this section, we first present results for bandlimited signals. For generating bandlimited or lowpass signals, we used an Arduino microcontroller (see Figure 10), which converts the digital signal to analog signal via a DAC. The digital signals were generated using MATLAB software. Two examples of 1 kHz bandlimited signals are presented in Figure 11a,b. The modulo hardware folds the signals to stay within the dynamic

FIGURE 10 Bandlimited signal generator.
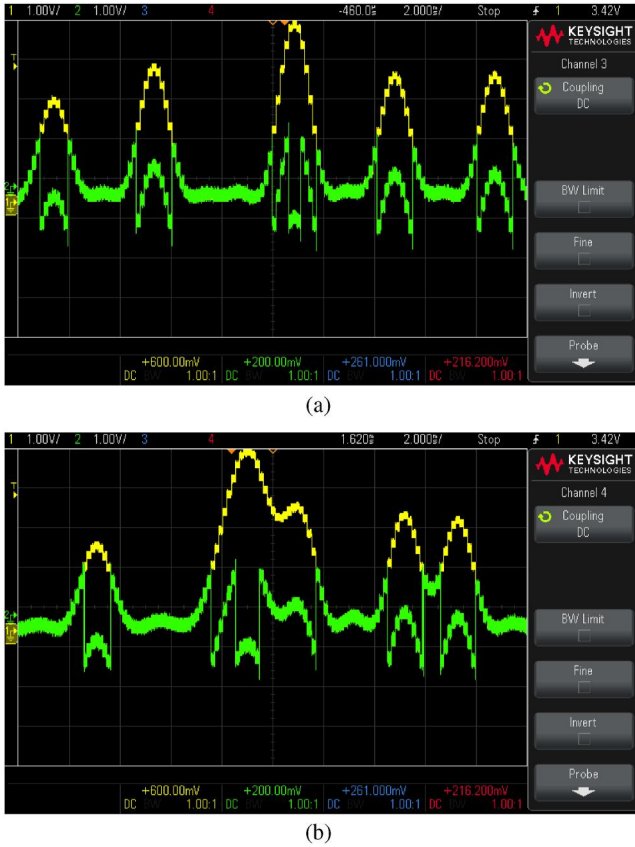

(a)

(b)

FIGURE 11 Screenshots of the oscilloscope capturing the two bandlimited input signals (yellow) and their folded output signals (green). (a) Example-1 of bandlimited signal and (b) Example-2 of bandlimited signal.

range, as shown in Figure 11. The signals are sampled with an oversampling factor of five (OF = 5), and the B^2R^2 algorithm is applied for unfolding. The unfolded or reconstructed signals are shown in Figure 12. We observe that the reconstruction is close to the true signals except for an amplitude scaling factor, which is the result of scaling within the hardware.

4.3 | Hardware results for FRI signals

Before presenting the results for FRI signals, we briefly discuss the FRI signal model and its sampling and reconstruction
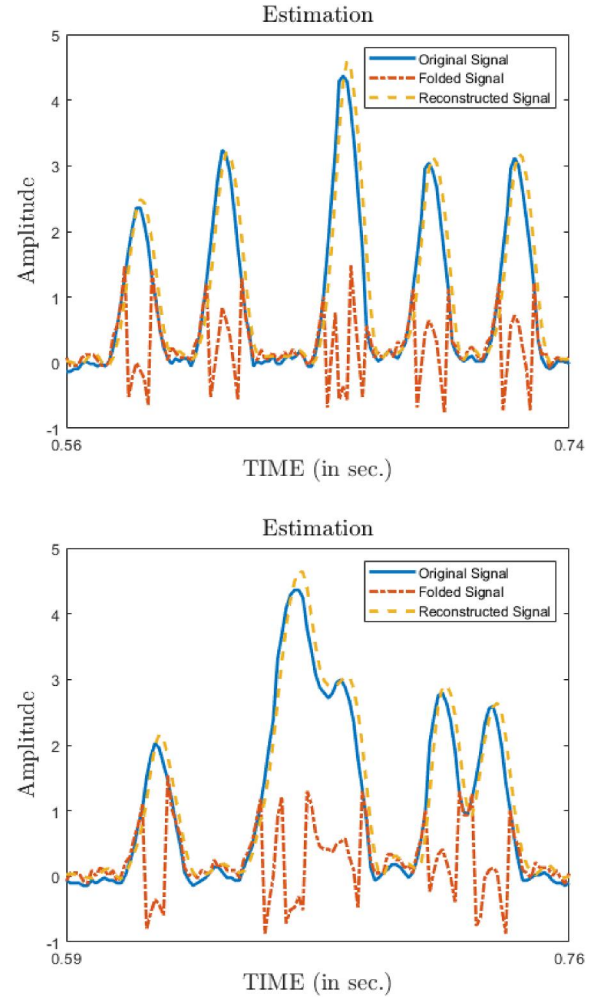

FIGURE 12 Hardware results for bandlimited signals. The B^2R^2 algorithm is used to unfold $y_\lambda(t)$ (measured at the output of hardware), and the unfolded signal $\hat{y}(t)$ is plotted with bandlimited signal $y(t)$.

mechanism. Consider an FRI signal consisting of a stream of L pulses:

$$f(t) = \sum_{\ell=1}^{L} a_\ell h(t - t_\ell), \qquad (4)$$

where the pulse $h(t)$ a real-valued known pulse. We assume that $\{a_\ell\}_{\ell=1}^{L}$ are real-valued and $\{t_\ell\}_{\ell=1}^{L} \subset (0, T_0] \subset \mathbb{R}$ for a known T_0.

The FRI signal model in Equation (4) is encountered in several scientific applications such as radar imaging [29–31], ultrasound imaging [24, 32, 33], light detection and ranging [34], time-domain optical coherence tomography [35], and other time-of-flight imaging systems. In these applications, $h(t)$ is the transmitted pulse and $\{a_\ell h(t - t_\ell)\}_{\ell=1}^{L}$ constitute the reflections from L point targets. The amplitudes $\{a_\ell\}_{\ell=1}^{L}$ depend on the sizes of the targets and the delays $\{t_\ell\}_{\ell=1}^{L}$ are proportional to the distances of the targets from the transmitter. Here T_0 denotes the maximum time delay of the targets. The signal $f(t)$ is specified by $\{a_\ell, t_\ell\}_{\ell=1}^{L}$ and can be

reconstructed from its sub-Nyquist measurements acquired using an appropriate sampling kernel [23–25, 36]. Given their widespread application, here we consider sampling and reconstruction of FRI signals by using our hardware prototype.

FRI signals can be perfectly reconstructed by applying high-resolution spectral estimation methods, such as the annihilating filter or Prony's method and its variants [37–43] to the Fourier measurements:

$$S(k\omega_0) = \frac{F(k\omega_0)}{H(k\omega_0)} = \sum_{\ell=1}^{L} a_\ell e^{-jk\omega_0 t_\ell}, k \in \{-K, \dots, K\}, \quad (5)$$

where we assume that $H(k\omega_0) \neq 0$. Here $K \geq L$ and $\omega_0 = \frac{2\pi}{T_0}$ [44]. The Fourier measurements $\{S(k\omega_0)\}_{k=-K}^{K}$ can be determined from the samples $(f \times g)(nT_s)$ where $g(t)$ is an ideal lowpass filter with bandwidth $[-K\omega_0, K\omega_0]$ and $T_s = \frac{2\pi}{(2K+1)\omega_0}$. In practice, the duration of the pulse $h(t)$ is very short, and hence $f(t)$ has a wide bandwidth. This results in a large sampling rate (or Nyquist rate) if $f(t)$ is sampled directly. However, the filtered signal $y(t) = (f \times g)(t)$ is bandlimited to $[-K\omega_0, K\omega_0]$, which is much smaller than that of $h(t)$ and the sampling rate is much lower than the Nyquist rate.
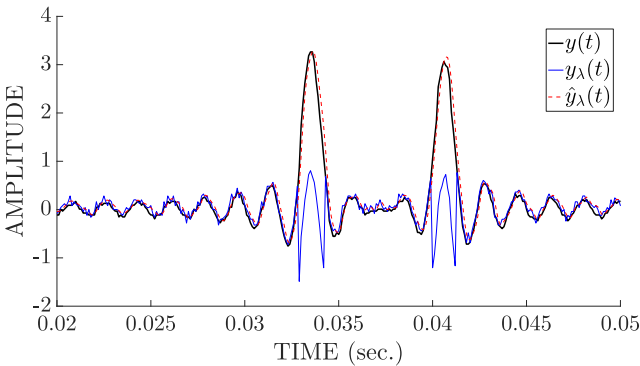
As in the bandlimited signal model, a modulo operation can be applied to the filtered signal $y(t)$ to avoid clipping. Then $y_\lambda(t)$ is sampled. Then to determine the Fourier samples
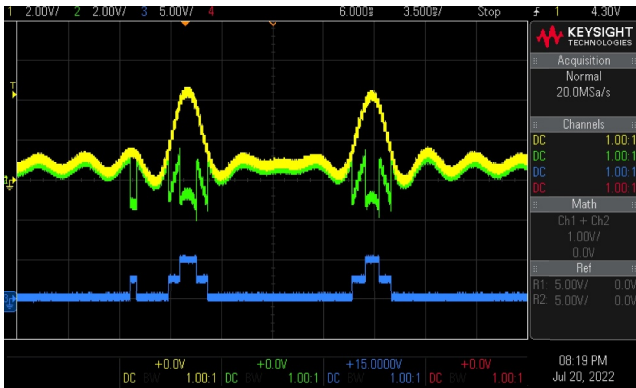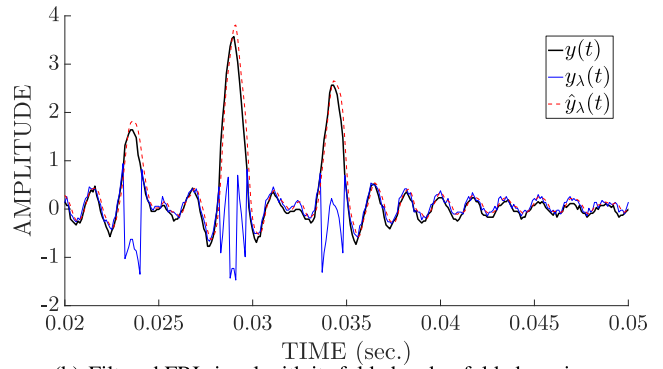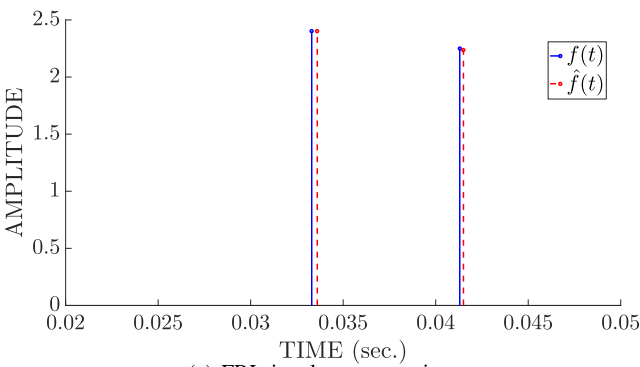

(a) Screenshot of the oscilloscope capturing filtered FRI signal (yellow), its folded output (green), and the DVG signal (blue).
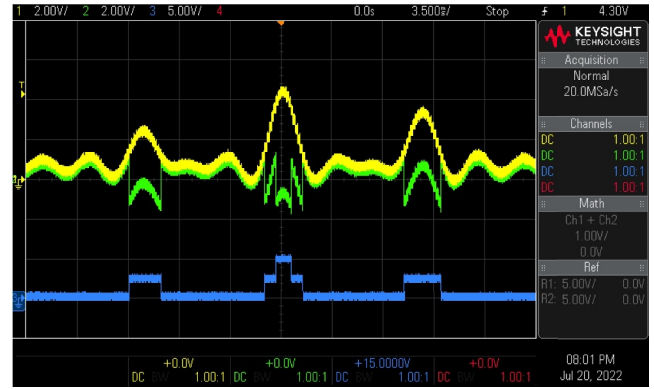
(b) Filtered FRI signal with its folded and unfolded versions.

(c) FRI signal reconstruction.

FIGURE 13 Reconstruction of an FRI signal ($L = 2$) via the modulo hardware.

(a) Screenshot of the oscilloscope capturing filtered FRI signal (yellow), its folded output (green), and the DVG signal (blue).
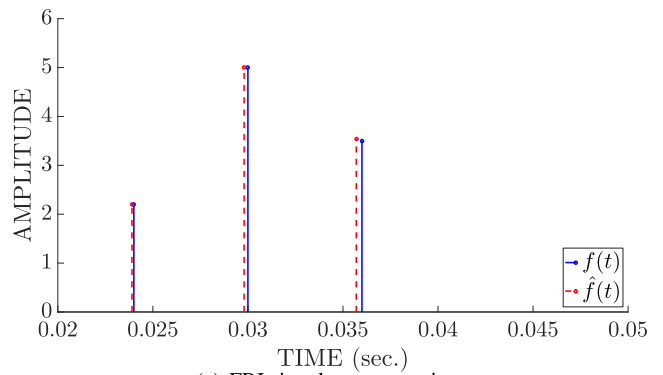

(b) Filtered FRI signal with its folded and unfolded versions.
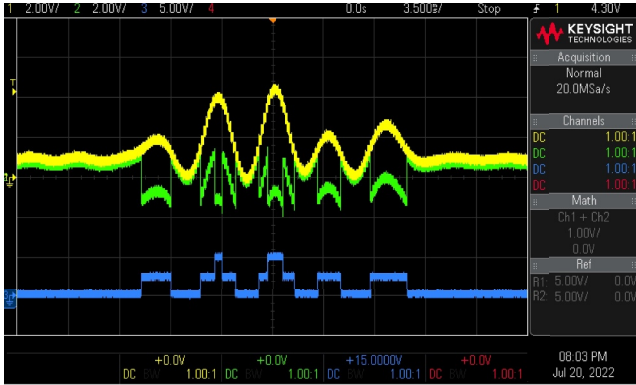

(c) FRI signal reconstruction.

FIGURE 14 Reconstruction of an FRI signal ($L = 3$) via the modulo hardware.

$\{S(k\omega_0)\}_{k=-K}^{K}$, unfolding is first applied. Since the filtered signal is bandlimited, we use the proposed hardware for modulo folding and can apply the B^2R^2 algorithm from unfolding.
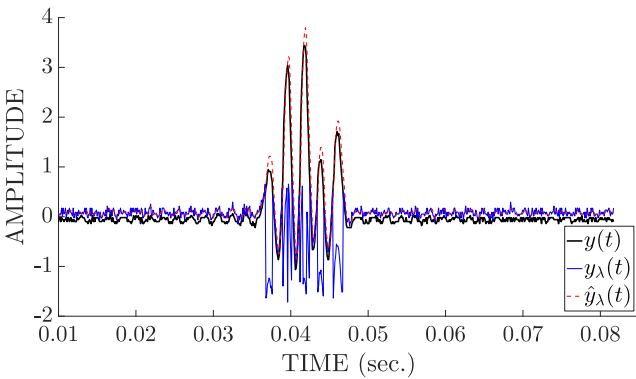
In our setup, to generate the FRI signals, we consider $h(t)$ to be a short pulse of bandwidth 30 kHz (Nyquist rate = 60 kHz). We consider three examples with $L = 2$, 3, and 5. The amplitudes and time delays are generated randomly. The maximum time delay is $T_0 = 0.1$ s. Once generated, the FRI signal is lowpass filtered with a cutoff frequency of 1 kHz. MATLAB is used to generate the samples of filtered FRI

(a) Screenshot of the oscilloscope capturing filtered FRI signal (yellow), its folded output (green), and the DVG signal (blue).

(b) Filtered FRI signal with its folded and unfolded versions.

(c) FRI signal reconstruction.

FIGURE 15 Reconstruction of an FRI signal ($L = 5$) via the modulo hardware.

signals and then an Arduino microcontroller is used to generate the analog counterpart of them. The signal is then folded using the hardware, and the folded signals are sampled. The sampling rate is 10 kHz which is five times higher than the sampling rate of the lowpass signal. Still, the rate is six times lower than the Nyquist rate, and hence the system operates at a sub-Nyquist rate.
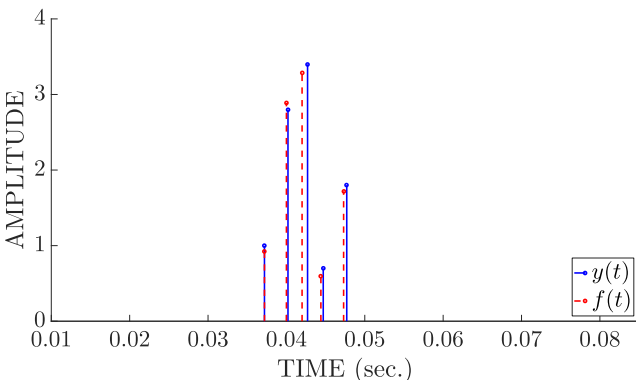
We first applied the B^2R^2 algorithm to unfold the signal and then used ESPRIT [43] to estimate the time delays and amplitudes of the FRI signals. In Figures 13–15, we show sampling and reconstruction of FRI signals with $L = 2$, 3, 5, respectively. The FRI signals followed by a lowpass filter are presented in Figures 13a, 14a, and 15a. The reconstruction of the lowpass signals displayed in Figures 13b, 14b, and 15b, where $y(t)$ described the LPF output, $y_\lambda(t)$ is the folded signal (output of the modulo hardware), and the unfolded signals are given by $\hat{y}(t)$. Figures 13c, 14c, and 15c show location and amplitude of the true signal $f(t)$ and estimated FRI signal $\hat{f}(t)$. The maximum error in the estimation of time delay is −15 dB which shows that the system can be used in applications like radar and ultrasound imaging.

5 | CONCLUSIONS

We presented a hardware prototype for the modulo folding system and showed that for different bandlimited and FRI signals, the hardware is able to fold the signal faithfully. In particular, we were able to sample signals with 8 times the dynamic range of the ADC roughly. We also addressed the time delay issue of the modulo system and presented a hardware solution. The overall system operates five times below the Nyquist rate, which enables one to use low-rate, low-dynamic range, power-efficient ADCs.

AUTHOR CONTRIBUTIONS

Satish Mulleti: Conceptualisation, Data curation, Formal analysis, Investigation, Methodology, Project administration, Supervision, Validation, Visualisation, Writing – original draft, Writing – review & editing. Eliya Reznitskiy: Investigation, Methodology, Software, Validation. Shlomi Savariego: Methodology, Resources, Software, Validation. Moshe Namer: Data curation, Methodology, Resources, Supervision, Validation. Nimrod Glazer: Conceptualisation, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing. Yonina C. Eldar: Conceptualisation, Funding acquisition, Investigation, Project administration, Supervision, Visualisation, Writing – review & editing.

CONFLICT OF INTEREST STATEMENT

We have conflict of interests with the following personnels: (1) Ayush Bhandari, Imperial College London. (2) Felix Krahmer, Tech Univ Munich. (3) Dorian Florescu, Imperial College London.

DATA AVAILABILITY STATEMENT

The results in the paper are based on the data generated from the proposed hardware prototype. Data is available on request from the authors.

PERMISSION TO REPRODUCE MATERIALS FROM OTHER SOURCES

None.

ORCID

Satish Mulleti https://orcid.org/0000-0002-3995-9070

REFERENCES

1. Marks, R.: Restoring lost samples from an oversampled band-limited signal. IEEE Trans. Acoust. Speech Signal Process. 31(3), 752–755 (1983). https://doi.org/10.1109/tassp.1983.1164101
2. Marks, R., Radbel, D.: Error of linear estimation of lost samples in an oversampled band-limited signal. IEEE Trans. Acoust. Speech Signal Process. 32(3), 648–654 (1984). https://doi.org/10.1109/tassp.1984.1164370
3. Pérez, J.P.A., Pueyo, S.C., López, B.C.: Automatic Gain Control. Springer (2011)
4. Mercy, D.: A review of automatic gain control theory. Radio Electron. Eng. 51(11.12), 579–590 (1981). https://doi.org/10.1049/ree.1981.0084
5. Landau, H.J.: On the recovery of a band-limited signal, after instantaneous companding and subsequent band limiting. Bell Syst. Tech. J. 39(2), 351–364 (1960). https://doi.org/10.1002/j.1538-7305.1960.tb01604.x
6. Landau, H.J., Miranker, W.L.: The recovery of distorted band-limited signals. J Math. Anal. Appl. 2(1), 97–104 (1961). https://doi.org/10.1016/0022-247x(61)90047-6
7. Bhandari, A., Krahmer, F., Raskar, R.: On unlimited sampling and reconstruction. IEEE Trans. Signal Process. 69, 3827–3839 (2020). https://doi.org/10.1109/tsp.2020.3041955
8. Romanov, E., Ordentlich, O.: Above the Nyquist rate, modulo folding does not hurt. IEEE Signal Process. Lett. 26(8), 1167–1171 (2019). https://doi.org/10.1109/lsp.2019.2923835
9. Bhandari, A., Krahmer, F., Poskitt, T.: Unlimited sampling from theory to practice: Fourier-Prony recovery and prototype ADC. IEEE Trans. Signal Process. 70, 1131–1141 (2022). https://doi.org/10.1109/tsp.2021.3113497
10. Azar, E., Mulleti, S., Eldar, Y.C.: Residual recovery algorithm for modulo sampling. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5722–5726 (2022)
11. Azar, E., Mulleti, S., Eldar, Y.C.: Robust unlimited sampling beyond modulo. arXiv preprint arXiv:220614656 (2022)
12. Bhandari, A., Krahmer, F., Raskar, R.: Unlimited sampling of sparse signals. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4569–4573 (2018)
13. Prasanna, D., Sriram, C., Murthy, C.R.: On the identifiability of sparse vectors from modulo compressed sensing measurements. IEEE Signal Process. Lett. 28, 131–134 (2021). https://doi.org/10.1109/lsp.2020.3047584
14. FernándezMenduiña, S., et al.: DoA estimation via unlimited sensing. In: Proceedings of European Signal Processing Conference (EUSIPCO), pp. 1866–1870 (2021)
15. Bhandari, A., Beckmann, M., Krahmer, F.: The modulo Radon transform and its inversion. In: Proceedings of European Signal Processing Conference (EUSIPCO), pp. 770–774 (2021)
16. Ji, F., Pratibha, P., Tay, W.P.: On folded graph signals. In: Proceedings of Global Conference on Signal and Information Processing (GlobalSIP), pp. 1–5 (2019)
17. Park, D., Rhee, J., Joo, Y.: A wide dynamic-range CMOS image sensor using self-reset technique. IEEE Electron. Device Lett. 28(10), 890–892 (2007). https://doi.org/10.1109/led.2007.905396
18. Sasagawa, K., et al.: An implantable CMOS image sensor with self-reset pixels for functional brain imaging. IEEE Trans. Electron. Dev. 63(1), 215–222 (2016). https://doi.org/10.1109/ted.2015.2454435
19. Yuan, J., et al.: An activity-triggered 95.3 db DR −75.6 db THD CMOS imaging sensor with digital calibration. IEEE J. Solid State Circ. 44(10), 2834–2843 (2009). https://doi.org/10.1109/jssc.2009.2027929
20. Krishna, A., et al.: Unlimited dynamic range analog-to-digital conversion. arXiv preprint:191109371 (2019)
21. Bhandari, A.: Back in the US-SR: unlimited sampling and sparse super-resolution with its hardware validation. IEEE Signal Process. Lett. 29, 1047–1051 (2022). https://doi.org/10.1109/lsp.2022.3161865
22. Florescu, D., Krahmer, F., Bhandari, A.: The surprising benefits of hysteresis in unlimited sampling: theory, algorithms and experiments. IEEE Trans. Signal Process. 70, 616–630 (2022). https://doi.org/10.1109/tsp.2022.3142507
23. Vetterli, M., Marziliano, P., Blu, T.: Sampling signals with finite rate of innovation. IEEE Trans. Signal Process. 50(6), 1417–1428 (2002). https://doi.org/10.1109/tsp.2002.1003065
24. Tur, R., Eldar, Y.C., Friedman, Z.: Innovation rate sampling of pulse streams with application to ultrasound imaging. IEEE Trans. Signal Process. 59(4), 1827–1842 (2011). https://doi.org/10.1109/tsp.2011.2105480
25. Mulleti, S., Seelamantula, C.S.: Paley–Wiener characterization of kernels for finite-rate-of-innovation sampling. IEEE Trans. Signal Process. 65(22), 5860–5872 (2017). https://doi.org/10.1109/tsp.2017.2733484
26. Mulleti, S., Eldar, Y.C.: Modulo sampling of FRI signals. arXiv preprint arXiv:220708774 (2022)
27. Papoulis, A.: Limits on bandlimited signals. Proc. IEEE 55(10), 1677–1686 (1967). https://doi.org/10.1109/proc.1967.5960
28. Bhandari, A., Krahmer, F., Raskar, R.: On unlimited sampling. In: Proceedings of International Conference on Sampling Theory and Applications (SampTA), pp. 31–35 (2017)
29. Bar-Ilan, O., Eldar, Y.C.: Sub-Nyquist radar via Doppler focusing. IEEE Trans. Signal Process. 62(7), 1796–1811 (2014). https://doi.org/10.1109/tsp.2014.2304917
30. Bajwa, W.U., Gedalyahu, K., Eldar, Y.C.: Identification of parametric underspread linear systems and super-resolution radar. IEEE Trans. Signal Process. 59(6), 2548–2561 (2011). https://doi.org/10.1109/tsp.2011.2114657
31. Rudresh, S., Seelamantula, C.S.: Finite-rate-of-innovation-sampling-based super-resolution radar imaging. IEEE Trans. Signal Process. 65(19), 5021–5033 (2017). https://doi.org/10.1109/tsp.2017.2721917
32. Wagner, N., Eldar, Y.C., Friedman, Z.: Compressed beamforming in ultrasound imaging. IEEE Trans. Signal Process. 60(9), 4643–4657 (2012). https://doi.org/10.1109/tsp.2012.2200891
33. Mulleti, S., et al.: Ultrasound image reconstruction using the finite-rate-of-innovation principle. In: Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 1728–1732 (2014)
34. Castorena, J., Creusere, C.D.: Sampling of time-resolved full-waveform LIDAR signals at sub-Nyquist rates. IEEE Trans. Geosci. Rem. Sens. 53(7), 3791–3802 (2015). https://doi.org/10.1109/tgrs.2014.2383839
35. Blu, T., Bay, H., Unser, M.: A new high-resolution processing method for the deconvolution of optical coherence tomography signals. In: Proceedings of First IEEE International Symposium on Biomedical Imaging: Macro to Nano, vol. III, pp. 777–780 (2002)
36. Dragotti, P.L., Vetterli, M., Blu, T.: Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix. IEEE Trans. Signal Process. 55(5), 1741–1757 (2007). https://doi.org/10.1109/tsp.2006.890907
37. DeProny, G.R.: Essai experimental et analytique: Sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansive de la vapeur

de l'eau et de la vapeur de l'alcool, à différentes températures. J de l'Ecole polytechnique 1(2), 24–76 (1795)

38. Plonka, G., Tasche, M.: Prony methods for recovery of structured function. GAMM-Mitteilungen 37(2), 239–258 (2014). https://doi.org/10.1002/gamm.201410011

39. Schmidt, R.O.: Multiple emitter location and signal parameter estimation. IEEE Trans. Antenn. Propag. 34(3), 276–280 (1986). https://doi.org/10.1109/tap.1986.1143830

40. Barabell, A.: Improving the resolution performance of eigenstructure-based direction-finding algorithms. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 8, pp. 336–339 (1983)

41. Rao, B.D., Hari, K.V.S.: Performance analysis of root-MUSIC. IEEE Trans. Acoust. Speech Signal Process. 37(12), 1939–1949 (1989). https://doi.org/10.1109/29.45540

42. Hua, Y., Sarkar, T.K.: Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. IEEE Trans. Acoust. Speech Signal Process. 38(5), 814–824 (1990). https://doi.org/10.1109/29.56027

43. Paulraj, A., Roy, R., Kailath, T.: A subspace rotation approach to signal parameter estimation. Proc. IEEE 74(7), 1044–1046 (1986). https://doi.org/10.1109/proc.1986.13583

44. Eldar, Y.C.: Sampling Theory: Beyond Bandlimited Systems. Cambridge University Press (2015)