

On-Board Two-Channel Modulo ADC System with Folding and Reconstruction

Zeyuan Li¹ Wenyi Yan² Lu Gan² Guoquan Li³ Hongqing Liu³

¹School of Integrated Circuits, Anhui University, China

²Dept. of Electrical and Electronic Engineering, Brunel University, London, UK

³School of Communications and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China

Abstract—This paper presents an FPGA-based real-time implementation of the robust Chinese remainder theorem (RCRT) for signal reconstruction in two-channel modulo analog-to-digital converters (ADCs) systems. A hardware-oriented formulation of the RCRT algorithm is developed, removing modular inversion to enable a fully deterministic reconstruction pipeline under practical hardware constraints, suitable for FPGA deployment. Building on this formulation, we realise a fully hardware-based Modulo ADC system that performs both signal folding and reconstruction on the same board, without any software involvement. The proposed design achieves accurate real-time signal recovery, demonstrating the feasibility and efficiency of RCRT-based architectures for wide-dynamic-range acquisition in hardware-limited sampling and quantization scenarios.

Index Terms—Modulo samplers, Analog-to-digital converters, Chinese remainder theorem, Wide-dynamic-range acquisition.

I. INTRODUCTION

Conventional analog-to-digital converters (ADCs) are fundamentally limited by their input dynamic range. When the input amplitude exceeds the ADC reference range, signal clipping occurs, causing irreversible distortion and information loss. Increasing the dynamic range typically requires higher resolution or multi-stage gain control, both of which lead to significant increases in power consumption, circuit complexity, and conversion latency [1], [2]. These constraints make it difficult for traditional ADCs to efficiently capture wide-dynamic-range signals.

To overcome these limitations, modulo ADCs prevent saturation by folding the input instead of clipping [3]. This non-linear operation preserves all information within each folded sample. Existing reconstruction methods, including difference-based residual estimation [3], [4], frequency-domain schemes within the unlimited sensing framework (USF) [5], [6], and line-spectral estimation algorithms [7], [8], have confirmed the feasibility of modulo-based recovery, with applications spanning radar, imaging, and communications [9]–[11].

However, these existing methods are computationally demanding and largely limited to software-based or *offline processing*. To our knowledge, no existing system achieves a fully integrated, software-free hardware realization that performs both signal folding and reconstruction within a unified, deterministic pipeline under practical hardware constraints.

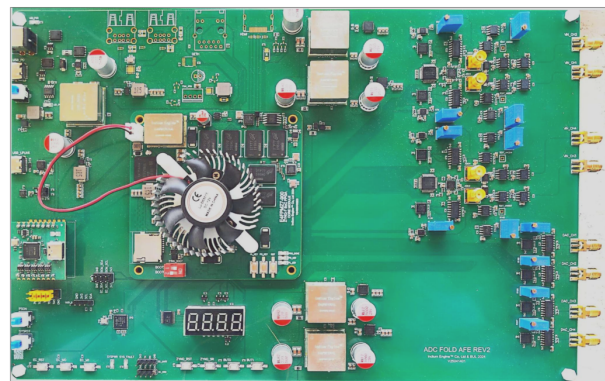


Fig. 1. Multi-channel Modulo ADCs hardware prototype.

To address this, this work adopts a two-channel Modulo ADC architecture combined with a reconstruction algorithm based on the robust Chinese remainder theorem (RCRT) [12]–[16]. By developing a hardware-oriented RCRT formulation, we eliminate the need for modular inversion and iterative refinement, enabling a fixed-latency reconstruction pipeline suitable for real-time FPGA deployment. Building on this foundation, we demonstrate the fully hardware-implemented, software-free reconstruction pipeline, where both folding and recovery are executed on-board (Fig. 1). The main contributions of this work are summarised as follows:

- **Hardware-oriented RCRT algorithm:** A reformulated RCRT for efficient hardware implementation, eliminating modular inversion and iteration for fully deterministic reconstruction.
- **Fully hardware modulo ADC system:** The **first** complete hardware realisation performing both folding and reconstruction *without software involvement*.
- **System-level experimental validation:** Hardware experiments on a two-channel FPGA prototype verify accurate real-time reconstruction, confirming the robustness and practicality of the proposed design.

II. SYSTEM OVERVIEW

Fig. 2 illustrates the overall architecture of the proposed two-channel Modulo ADC system. From a functional perspective, the system comprises two main parts: a **folding**

front-end, responsible for real-time modulo folding of the analog input signal, and a **reconstruction back-end**, which performs digital recovery of the unfolded signal using the proposed RCRT algorithm. From a circuit perspective, the overall design is a mixed-signal implementation that combines analog circuitry with FPGA-based digital logic. Specifically, the analog front-end together with the FPGA generates the modulo signal through closed-loop folding, while the ADC and FPGA jointly perform sampling and real-time reconstruction.

A. Folding Front-end

From a mathematical perspective, the modulo ADC mitigates saturation by folding the input signal back into a bounded ADC dynamic range Δ , defined for $x \in \mathbb{R}$ as [3]:

$$\langle x \rangle_{\Delta} = \Delta \left(\left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor - \frac{1}{2} \right), \quad \llbracket x \rrbracket \stackrel{\text{def}}{=} x - \lfloor x \rfloor. \quad (1)$$

Consider a bandlimited input $g(t)$ processed by a two-channel Modulo ADC system, where each channel employs a distinct folding range:

$$\Delta_{\ell} = \tau_{\ell} \epsilon, \quad \ell = 1, 2, \quad (2)$$

with $\tau_1, \tau_2 \in \mathbb{Z}^+$ being coprime integers satisfying $2 \leq \tau_1 < \tau_2$, and $\epsilon > 0$ a common scaling factor. The modulo outputs from the two-channel folding front-end are expressed as

$$y_{\ell}(t) = \langle g(t) \rangle_{\Delta_{\ell}}, \quad \ell = 1, 2. \quad (3)$$

In our hardware prototype (Fig. 1), the modulo operation is implemented by a closed-loop analog front-end integrated with a digital control circuit. The front-end comprises a bipolar threshold generator, a high-speed window comparator, and a loop-control DAC with programmable gain, forming a feedback loop that folds the input voltage into $\pm \Delta_{\ell}/2$ and maintains it within the ADC's dynamic range. An FPGA-based digital controller synchronises the timing and feedback of both channels to prevent clipping [17]. Due to page limitations, the detailed circuit implementation of the analog folding front-end will be provided in a forthcoming journal paper.

B. Reconstruction Back-End

In our hardware prototype (Fig. 1), the modulo signal $y_{\ell}(t)$ is digitized by an AD9288 dual-channel 8-bit ADC, whose input range of ± 0.5 V corresponds to a digital output range of $[-128, 127]$. A Xilinx Zynq-7000 FPGA serves as the digital processing unit, providing precise clock control, sampling synchronization, and real-time reconstruction.

Suppose sampling with period T produces two folded discrete-time sequences, given by

$$\hat{y}_{\ell}[k] = y_{\ell}(kT) + e_{\ell}[k], \quad k \in \mathbb{Z}, \quad (4)$$

where $e_{\ell}[k]$ denotes quantization and circuit noise.

RCRT forms the core of the digital signal reconstruction back-end. It operates on modulo samples to estimate the correct folding indices and reconstruct the original signal. The reconstructed signal $\hat{g}[k]$ can thus be expressed as

$$\hat{g}[k] = \Delta_{\ell} \hat{n}_{\ell}[k] + \hat{y}_{\ell}[k], \quad (5)$$

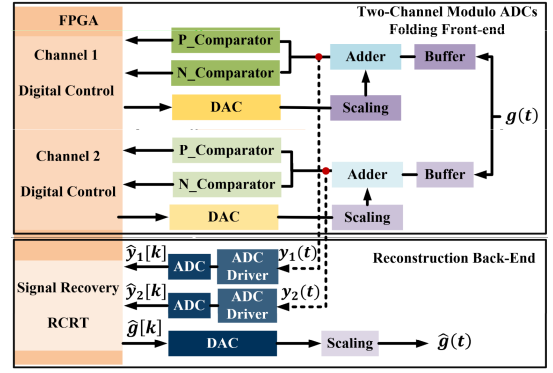


Fig. 2. Overall architecture of the proposed two-channel Modulo ADC sampling and reconstruction system.

where $\hat{n}_{\ell}[k] \in \mathbb{Z}$ denotes the unknown folding index associated with the ℓ th channel. RCRT performs *point-to-point recovery*, where each sample is independently reconstructed from its corresponding residues, enabling fully parallel and streaming operation. The algorithm provides a closed-form solution based on modular arithmetic, mainly additions, multiplications, and rounding, without iterative refinement or matrix inversion. Its computational complexity per sample remains constant, i.e., $\mathcal{O}(1)$, allowing deterministic and low-latency processing suitable for real-time FPGA deployment.

However, conventional RCRT requires modular inverse computation, which increases arithmetic complexity and introduces additional logic for modular multiplication on hardware. Although such inverses can be precomputed, their use still complicates datapath design and timing control. To address this, the next section presents a hardware-oriented reformulation that obviates online modular inversion and streamlines the reconstruction logic for efficient FPGA implementation.

III. ALGORITHM AND HARDWARE DESIGN

A. Hardware-Oriented Algorithm

In the RCRT algorithm, different parameter selection strategies can be adopted for τ_{ℓ} in (2) [13], [18], [19]. In practical implementations, however, the reconstruction algorithm should maximise error tolerance to counteract real-world nonidealities such as quantisation error, harmonic distortion, and overshoot at folding edges. Following Corollary 1 in [13], this maximum robustness can be achieved by selecting two consecutive coprime integers as the moduli, $\tau_2 = \tau_1 + 1$. With this parameter choice, the following corollary shows that modulo inversion is no longer required, forming the basis of the simplified reconstruction.

Corollary 1. *Let τ_1 and τ_2 be defined in (2) with $\tau_2 = \tau_1 + 1$. Since $\tau_1 \tau_1^{-1} \equiv 1 \pmod{\tau_2}$, it follows that τ_1 is its own modular inverse modulo τ_2 ; that is, $\tau_1^{-1} \equiv \tau_1 \pmod{\tau_2}$.*

Proof: As $\gcd(\tau_1, \tau_2) = 1$, τ_1 admits a unique inverse. Noting that $\tau_1 \equiv -1 \pmod{\tau_2}$, we obtain $\tau_1^2 \equiv (-1)^2 \equiv 1 \pmod{\tau_2}$, which proves the claim. ■

Algorithm 1 summarises the proposed hardware-oriented RCRT, designed to meet hardware constraints by eliminating

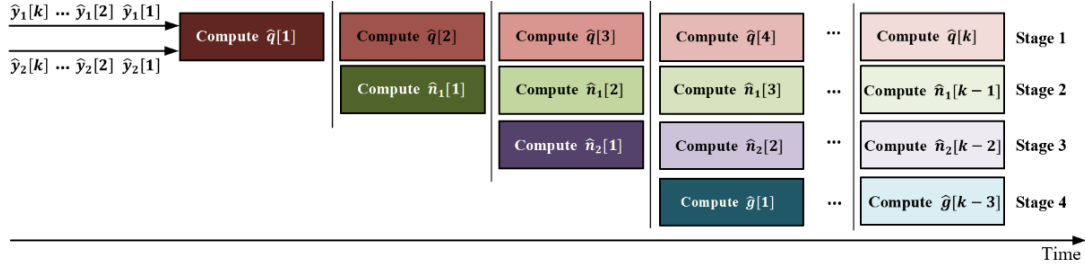


Fig. 3. Functional schedule and pipeline flow of the FPGA-based RCRT reconstruction.

Algorithm 1 Hardware-Oriented Two-Channel RCRT

Data: Threshold parameter τ_1, τ_2, ϵ , and modulo samples $\hat{y}_\ell[k], \ell = 1, 2$

Result: Reconstructed estimate $\hat{g}[k]$

- 1: Compute $\hat{q}[k] = \left\lfloor \frac{y_2[k] - y_1[k]}{\epsilon} \right\rfloor$
- 2: Compute $\hat{n}_1[k] = \langle \hat{q}[k] \tau_1 \rangle_{\tau_2}$
- 3: Compute $\hat{n}_2[k] = \frac{\tau_2}{\hat{n}_1[k] \tau_1 - \hat{q}[k]}$
- 4: Compute $\hat{g}[k] = \frac{1}{2} \sum_{\ell=1}^2 (\hat{n}_\ell[k] \Delta_\ell + \hat{y}_\ell[k])$

modular inversions. In contrast, the conventional RCRT [13] requires computing a modular inverse between τ_1 and τ_2 .

B. On-Chip RCRT Implementation

FPGA is chosen for its deterministic timing and parallel processing capability. It ensures clock-level synchronisation between the two modulo channels and enables a fully pipelined architecture for fixed-latency computation. The proposed RCRT reconstruction adopts such a pipelined design (Fig. 3), where point-to-point recovery on a continuous stream of modulo samples allows concurrent processing of successive data, maximising throughput with constant latency per sample. According to Algorithm 1, the reconstruction is divided into four sequential stages: (1) difference $\hat{q}[k]$ computation, (2) compute $\hat{n}_1[k]$, (3) compute $\hat{n}_2[k]$, and (4) signal recovery.

1) *Difference $\hat{q}[k]$ Computation:* The $\hat{q}[k]$ computation module performs a fixed-point subtraction followed by an integer division with rounding (Fig. 4a). The subtraction stage is realized using a ripple-carry adder (RCA) configured with one inverted operand, providing a compact structure and predictable timing closure on FPGA fabric. The divider adopts a radix-2 sequential architecture with a 12-stage fully pipelined datapath, yielding both quotient and remainder outputs after a 12-clock-cycle latency. A rounding unit follows the divider to implement an integer round-half-up operation based on the integer quotient Q_{int} , remainder R , and divisor D :

- For $R > 0$: output Q_{int} if $|R| < D/2$, otherwise $Q_{\text{int}} + 1$.
- For $R < 0$: output Q_{int} if $|R| < D/2$, otherwise $Q_{\text{int}} - 1$.

2) *Computation of $\hat{n}_1[k]$:* The $\hat{n}_1[k]$ computation module performs modular reduction of the intermediate value $\hat{q}[k]$ based on the symmetric modulo operation in (1). As shown

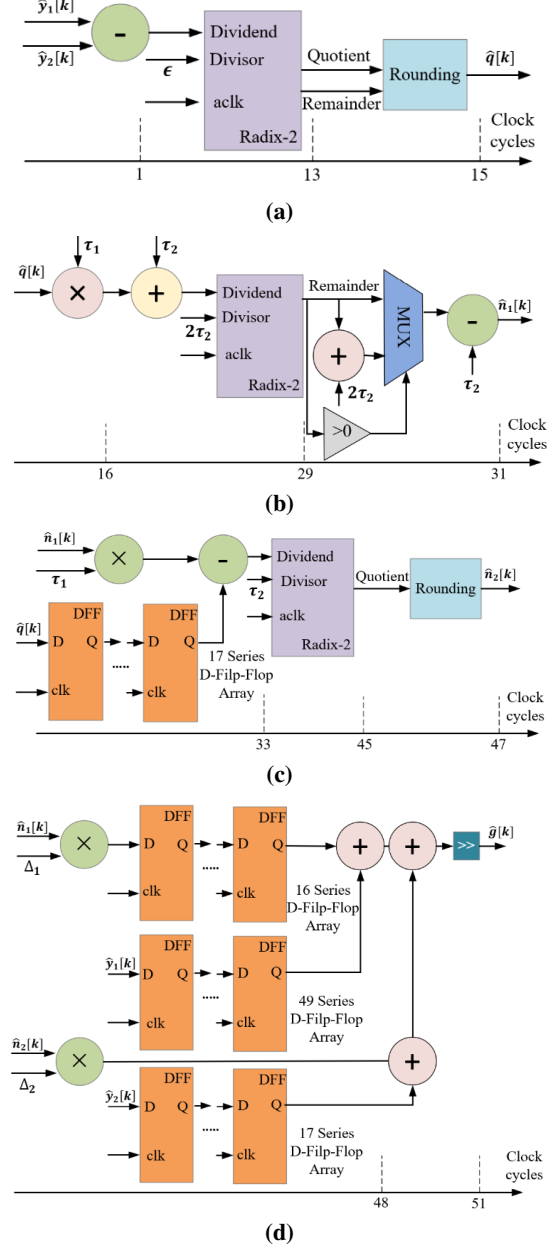


Fig. 4. Pipeline architecture of the proposed RCRT reconstruction system: (a) $\hat{q}[k]$ difference computation; (b) $\hat{n}_1[k]$ modular reduction; (c) $\hat{n}_2[k]$ computation with pipeline alignment; (d) Reconstruction $\hat{g}[k]$.

in Fig. 4b, it consists of a multiplier, an adder, a radix-2 based divider, and a symbol-processing circuit implemented on FPGA DSP and logic resources. The symbol-processing circuit evaluates the sign of the divider remainder to enforce the symmetric modulo rule: if the remainder is negative, an offset of $2\tau_2$ is added to restore it to the correct range; otherwise, the value is directly passed to the next stage.

3) *Computation of $\hat{n}_2[k]$* : The $\hat{n}_2[k]$ computation module generates the second folding index from $\hat{n}_1[k]$ and the delayed $\hat{q}[k]$, as shown in Fig. 4c. Due to the pipelined structure, $\hat{q}[k]$ is delayed by 17 clock cycles using a shift-register array to align with $\hat{n}_1[k]$, ensuring both operands reach the multiplier simultaneously. This timing alignment eliminates inter-stream delay and preserves full pipeline throughput, completing in 47 clock cycles.

4) *Signal Reconstruction*: The $\hat{g}[k]$ computation module performs the final weighted accumulation of both channels after precise timing alignment, as shown in Fig. 4d. Since $\hat{n}_1[k]$, $\hat{n}_2[k]$, $\hat{y}_1[k]$, and $\hat{y}_2[k]$ arrive from different pipeline depths, dedicated delay lines are inserted to achieve clock-level synchronization. The 16-cycle and 49-cycle shift-register arrays align both channels before accumulation, ensuring deterministic timing and eliminating phase skew between signal paths. After alignment, the two channel results are summed and averaged to generate $\hat{g}[k]$ with fixed 51-cycle latency and continuous streaming throughput.

Discussion. Existing modulo ADC reconstruction methods, such as LASSO-based sparse recovery [20], higher-order difference (HoD) [21], and iterative schemes (e.g., ITER-SIS) [22], achieve high accuracy in simulation but rely on signal sparsity or spectral priors and are sensitive to noise and sampling errors. In HoD-based methods [21], the sampling error can propagate through subsequent steps, causing cumulative waveform distortion, while iterative approaches require floating-point operations and matrix updates, hindering real-time hardware deployment. In contrast, the proposed RCRT-based reconstruction has a closed-form algebraic structure with constant per-sample complexity.

C. Resource Utilization and Power Consumption

We now summarise the resource utilisation and power consumption of the proposed design. The folding front-end controller uses 18 LUTs and 19 registers, while the reconstruction back-end occupies 1182 LUTs, 1813 registers, and two DSP slices, accounting for less than 3% of the available resources on the XC7Z100 device. The total power consumption is approximately 2.16 W, of which 1.53 W originates from the static power consumption of the unused PS7 subsystem. The effective dynamic power of the proposed design is therefore below 0.7 W, confirming its lightweight and energy-efficient real-time operation.

IV. HARDWARE RESULTS

A sinusoidal input signal with amplitude $A = 1.14$ V (corresponding to 289 digital counts) and frequency $f = 10$ kHz was applied to the modulo-ADC front-end. The modulo-ADC

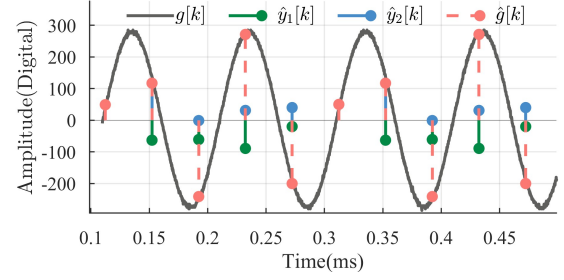


Fig. 5. Two-channel modulo sampling and FPGA-based RCRT reconstruction of a 10 kHz sinusoid (2.28 V_{pp}, 578 counts).

thresholds were set to $\Delta_1 = 0.72$ V and $\Delta_2 = 0.96$ V. For a fair comparison, both the ground-truth input $g[k]$ and the reconstructed waveform $\hat{g}[k]$ were simultaneously captured inside the FPGA. The ground-truth signal $g[k]$ was obtained at $50\times$ the Nyquist sampling rate to provide a high-resolution reference, whereas the reconstructed waveform $\hat{g}[k]$ was recovered from folded samples $\hat{y}_\ell[k]$ acquired at $1.55\times$ the Nyquist rate using the proposed RCRT algorithm implemented in the hardware prototype shown in Fig. 1.

As shown in Fig. 5, the proposed two-channel RCRT reconstruction algorithm successfully recovers the original sinusoidal waveform from the folded samples. The reconstructed waveform $\hat{g}[k]$ closely follows the ground-truth signal $g[k]$. To further quantify the reconstruction accuracy, the *matrix pencil method* was employed to estimate the amplitude $\hat{A}$ and frequency $\hat{f}$ from the discrete reconstructed samples [23]. The estimated parameters were $\hat{A} = 276$ (digital counts) and $\hat{f} = 10.165$ kHz, corresponding to deviations of approximately 4.5% in amplitude and 0.165 kHz (1.65%) in frequency from the ground truth, thereby confirming the high fidelity of the proposed reconstruction.

Remark. The estimated amplitude $\hat{A}$ exhibits deviations primarily attributable to the matrix pencil method's sensitivity to sampling alignment relative to signal peaks, rather than the RCRT pipeline's reconstruction accuracy.

Discussion. Unlike HoD-based methods [3] necessitating significantly higher sampling rates, our two-channel RCRT framework theoretically enables perfect recovery at the Nyquist rate [13]. Experiments confirm that our FPGA-based RCRT maintains high accuracy even at $1.55\times$ the Nyquist rate, validating its efficiency near theoretical limits.

V. CONCLUSION AND FUTURE WORK

This paper presents the first fully integrated, software-free hardware realization of a two-channel Modulo ADC system, performing both folding and reconstruction on-board. By eliminating modular inversion and iteration through a hardware-oriented RCRT formulation, we enable a deterministic FPGA reconstruction pipeline. Experiments verify accurate signal recovery, demonstrating the design's practicality under real-world hardware constraints. Future work will quantify the impact of noise and non-idealities, refine hardware calibration, and extend the architecture to multi-channel configurations for enhanced dynamic range.

REFERENCES

- [1] R. Bolaños-Pérez, J. M. Rocha-Pérez, A. Díaz-Sánchez, J. Ramirez-Angulo, and E. Tlelo-Cuautle, "CMOS analog AGC for biomedical applications," *Electronics*, vol. 9, no. 5, pp. 878, 2020.
- [2] J. Xu, A. T. Nguyen, T. Wu, W. Zhao, D. K. Luu, and Z. Yang, "A wide dynamic range neural data acquisition system with high-precision delta-sigma ADC and on-chip EC-PC spike processor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 3, pp. 425–440, 2020.
- [3] A. Bhandari, F. Krahmer, and R. Raskar, "On Unlimited Sampling and Reconstruction," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3827–3839, 2021, Conference Name: IEEE Transactions on Signal Processing.
- [4] A. Bhandari, F. Krahmer, and T. Poskitt, "Unlimited sampling from theory to practice: Fourier-prony recovery and prototype ADC," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1131–1141, 2022.
- [5] R. Guo, Y. Zhu, and A. Bhandari, "Sub-Nyquist USF spectral estimation: k frequencies with $6k + 4$ modulo samples," *IEEE Transactions on Signal Processing*, vol. 72, pp. 5065–5076, 2024.
- [6] Y. Zhu, R. Guo, and A. Bhandari, "Ironing the modulo folds: Unlimited sensing with 100x bandwidth expansion," in *2025 International Conference on Sampling Theory and Applications (SampTA)*, 2025, pp. 1–5.
- [7] Q. Zhang, J. Zhu, F. Qu, and D. W. Soh, "Line spectral estimation via unlimited sampling," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 5, pp. 7214–7231, 2024.
- [8] J. Zhu, J. Ma, Z. Liu, F. Qu, Z. Zhu, and Q. Zhang, "A modulo sampling hardware prototype and reconstruction algorithms evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–11, 2025.
- [9] T. Feuillen, M. Alaee-Kerahroodi, A. Bhandari, M. R. B. Shankar, and B. Ottersten, "Unlimited sampling for FMCW radars: A proof of concept," 2022, Institute of Electrical and Electronics Engineers.
- [10] A. Bhandari and F. Krahmer, "HDR imaging from quantization noise," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 101–105.
- [11] Z. Liu, A. Bhandari, and B. Clerckx, " λ -MIMO: Massive MIMO via modulo sampling," *IEEE Transactions on Communications*, vol. 71, no. 11, pp. 6301–6315, Nov 2023.
- [12] L. Gan and H. Liu, "High dynamic range sensing using multi-channel modulo samplers," *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop*, vol. 2020-June, 6 2020.
- [13] W. Yan, L. Gan, S. Hu, and H. Liu, "Towards optimized multi-channel modulo-ADCs: Moduli selection strategies and bit depth analysis," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 9496–9500.
- [14] W. Wang and X.-G. Xia, "A closed-form robust Chinese remainder theorem and its performance analysis," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5655–5666, 2010.
- [15] L. Xiao, X.-G. Xia, and Y.-P. Wang, "Exact and robust reconstructions of integer vectors based on multidimensional chinese remainder theorem (md-crt)," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5349–5364, 2020.
- [16] X. Li, X.-G. Xia, W. Wang, and W. Wang, "A robust generalized Chinese remainder theorem for two integers," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7491–7504, 2016.
- [17] Z. Li, W. Yan, R. Zhu, L. Gan, and H. Liu, "Live demonstration: Real-time high-amplitude signal acquisition with 2-channel modulo ADC," in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2025, pp. 1–1.
- [18] Y. Gong, L. Gan, and H. Liu, "Multi-channel modulo samplers constructed from gaussian integers," *IEEE Signal Processing Letters*, vol. 28, pp. 1828–1832, 2021.
- [19] W. Yan, L. Gan, and Y. D. Zhang, "Threshold sensitivity in two-channel modulo adcs: Analysis and robust reconstruction," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.
- [20] S. B. Shah, S. Mulleti, and Y. C. Eldar, "LASSO-based fast residual recovery for modulo sampling," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [21] A. Bhandari, F. Krahmer, and R. Raskar, "On unlimited sampling and reconstruction," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3827–3839, 2021.
- [22] R. Guo and A. Bhandari, "ITER-SIS: Robust unlimited sampling via iterative signal sieving," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [23] Y. Hua and T. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 814–824, 1990.