NoisyCode: A Lightweight Binary Protocol for Communication Under Full-Band Jamming

Omar Adel Ibrahim and Roberto Di Pietro
King Abdullah University of Science and Technology (KAUST)
Saudi Arabia
{omar.badreldin, roberto.dipietro}@kaust.edu.sa

***Abstract*—Wireless systems are increasingly exposed to jamming, and traditional anti-jamming techniques such as Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS), while effective against narrowband interference, can fail under full-band, high-power stochastic jamming. We introduce *NoisyCode*, a binary communication protocol that turns this regime into an advantage. *NoisyCode* encodes binary '1' as a deterministic waveform (e.g., a sine wave) and binary '0' as zero-mean noise—additive white Gaussian noise (AWGN) being the canonical instance. The protocol exploits a well-known statistical property: when multiple message repetitions are received and averaged, any zero-mean, uncorrelated interference diminishes while coherent waveforms are preserved. Consequently, the '1' waveform is reinforced while both the '0' symbols and the jammer's noise are progressively canceled, enabling reliable decoding even when the jammer's power at the receiver exceeds the legitimate signal by more than 20 dB. Matching the per-symbol energies of '1' and '0' yields a covertness bonus: the on-air signal exhibits no on/off energy pattern, hiding the bit pattern from a passive energy-detection adversary. We also discuss a silence-for-'0' variant that trades covertness for transmit-energy efficiency. We formalize the protocol, implement it on Universal Software Radio Peripheral (USRP) devices, and evaluate it experimentally under varying jamming conditions: *NoisyCode* achieves approximately 91% throughput with 20 message repetitions and up to 97% with 40, against around 1% for single-shot BPSK. *NoisyCode* offers a lightweight, hardware-minimal solution for critical low-rate communications such as IoT alerts, drone control messages, and spectrum-denied environments.**

***Index Terms*—Anti-jamming, Stochastic interference, Gaussian noise, Wireless security, Wireless communication**

I. Introduction

Wireless communication systems are deeply embedded in modern infrastructure, but their broadcast nature exposes them to jamming attacks that undermine availability at the physical layer and cannot be countered by encryption or authentication [1], [2]. Conventional anti-jamming solutions such as Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) spread transmissions over a wider band [3], but require strict synchronization and additional bandwidth, and they collapse against wide-band stochastic jamming that saturates the entire spectrum (with Additive White Gaussian Noise (AWGN) as the canonical worst case at a given power budget) [1].

To overcome these limitations, we present *NoisyCode*, a lightweight binary protocol that exploits the statistical properties of zero-mean stochastic interference (Gaussian noise being the most common and analytically convenient instance). *NoisyCode* encodes binary '1' as a deterministic waveform (e.g., a sine wave) and binary '0' as zero-mean noise; throughout the paper we instantiate the '0' symbol as AWGN, but the mechanism applies to any zero-mean, finite-variance process uncorrelated across repetitions. When multiple identical transmissions are received and averaged, the random zero-mean components—both the jammer's interference and the '0' symbols—tend to cancel out, while coherent waveform symbols are preserved at the '1' positions. This creates a clear separation between '1' and '0', allowing reliable decoding with minimal computational complexity, even under heavy interference. A second, less obvious benefit follows from the same encoding choice: when the per-symbol energies of '1' (sinusoid) and '0' (noise) are matched, the transmitted signal exhibits no on/off energy pattern, so a passive adversary monitoring received power cannot recover the bit pattern without ensemble-averaging N aligned repetitions and analyzing the FFT at the carrier frequency—essentially the legitimate receiver itself. *NoisyCode* thus combines jamming resilience with a basic form of physical-layer covertness against passive energy detection.

NoisyCode is therefore well suited to scenarios where short, critical messages must be delivered in contested environments—e.g., industrial monitoring or military signaling—at the price of a reduced bit rate. We also analyze a silence-for-'0' variant that trades covertness for transmit-energy efficiency, and show that *NoisyCode* should be viewed as a small family of operating modes along an energy/decoding/covertness trade-off rather than a single point. Using Universal Software Radio Peripheral (USRP) devices, we validate the protocol in real-world tests: even when the jammer arrives at the receiver more than 20 dB stronger than the legitimate signal, *NoisyCode* achieves about 91% throughput with 20 repetitions and up to 97% with 40, against roughly 1% for conventional communication.

Contributions. The main contributions of this work are:

- **Protocol Design:** We introduce *NoisyCode*, a binary protocol where '1' is a deterministic waveform and '0' is zero-mean noise (instantiated as AWGN, the canonical case), integrated with FFT-based energy detection, cross-correlation re-synchronization, and a tunable repetition parameter. While ensemble averaging and noise-based encoding are individually classical techniques, the con-

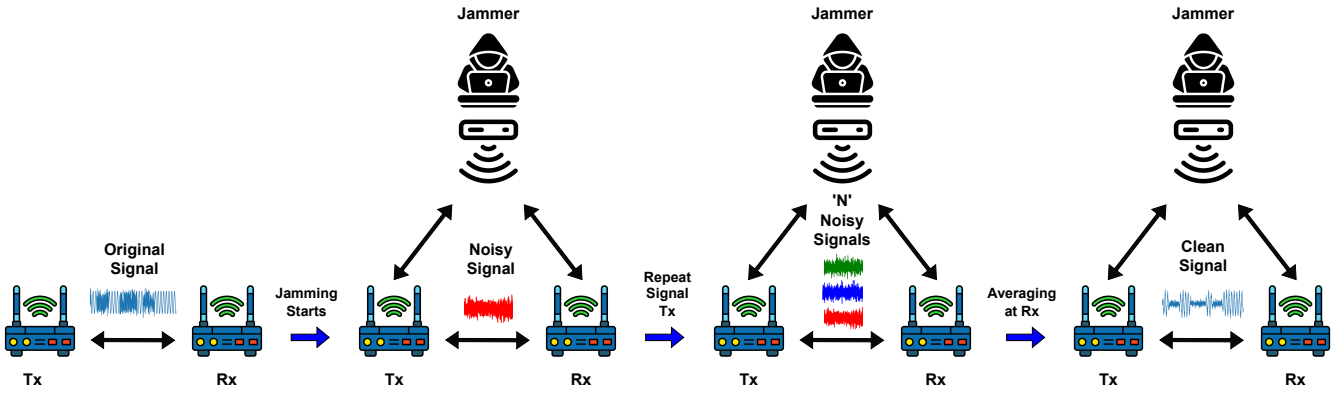


Fig. 1: Scenario and system overview of *NoisyCode*.

tribution of *NoisyCode* is their integration into a complete, hardware-minimal protocol that survives full-band stochastic jamming and provides covertness against passive energy detection, all without carrier-phase recovery or chip-level synchronization.

- **Experimental Validation:** We implement *NoisyCode* on USRP devices and evaluate it under different jamming conditions, showing substantial gains over conventional modulation.
- **Trade-Off Analysis:** We analyze the repetition overhead required for reliability and the silence-vs-noise encoding trade-off, showing that *NoisyCode* is particularly effective for short, critical communications where availability outweighs throughput.

II. Communication Protocol: *NoisyCode*

This section details the scenario, the design, and operation of the proposed communication protocol *NoisyCode*.

A. Scenario and Adversary Model

In many scenarios, reliable delivery of short, low-rate messages matters more than sustaining high-throughput streams: industrial monitoring, environmental sensing, machine-to-machine links, remote alerts, and military signaling (e.g., battlefield commands or drone coordination) all rely on brief but vital updates. In such settings, availability takes precedence over efficiency, making robustness the primary design goal.

We assume a strong adversary equipped with a wideband jammer that emits high-power zero-mean noise continuously across the spectrum (with AWGN as the canonical and most common instance), possibly with constant or reactive strategies. We additionally consider passive adversaries that monitor the channel and attempt to recover the transmitted bit pattern from received energy alone—a relevant threat in covert-communication settings.

B. NoisyCode *in a Nutshell*

NoisyCode is a binary protocol that leverages the statistical properties of zero-mean stochastic noise (with AWGN as the most common and canonical instance). A '1' is represented by a deterministic waveform (e.g., a sine wave), while a '0' is represented by zero-mean noise; in our implementation and experiments we use AWGN, but any zero-mean, finite-variance process uncorrelated across repetitions yields the same ensemble-averaging behavior. Messages are transmitted repeatedly so that the receiver (Rx) can ensemble-average across instances sent by the transmitter (Tx). After averaging, the random zero-mean components—both the jammer's interference and the '0' symbols—diminish, while the coherent '1' waveform accumulates. This separation enables reliable decoding even under wideband jamming, albeit at reduced throughput, and (with matched per-symbol energies) yields a covertness property treated in Section IV-D. The protocol is illustrated in Figure 1.

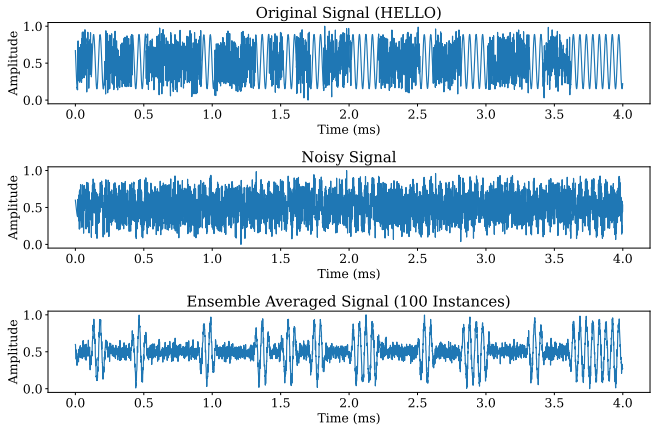


Fig. 2: Effect of ensemble averaging: (top) original HELLO signal; (middle) noisy transmission under AWGN jamming; (bottom) ensemble average of 100 repetitions, where both the jammer's noise and the '0' symbols diminish.

Figure 2 shows a USRP example encoding the word *HELLO*: averaging 100 repetitions recovers the clean signal, with both jammer noise and '0' symbols diminished. In practice, as few as 20 repetitions suffice for high decoding accuracy, depending on jamming power.

C. Message Generation

A bit '1' is represented by the complex sinusoid $s_1(t) = Ae^{j2\pi f_0 t}$, where $A = 1$ is the amplitude and $f_0 = 900$ MHz is the carrier frequency. A bit '0' is modeled as zero-mean stochastic noise; in our implementation we instantiate it as complex AWGN, $s_0(t) = n(t)$ with $n(t) = A(n_{\text{real}}(t) + jn_{\text{imag}}(t))$, where n_{real} and n_{imag} are zero-mean Gaussian with standard deviation $\sigma = 1$, chosen so that the per-symbol energy of '0' approximately matches that of '1'.

Remark on generality. The ensemble-averaging mechanism on which *NoisyCode* relies does *not* require Gaussianity. Any zero-mean, finite-variance process uncorrelated across the N repetitions decays as $1/\sqrt{N}$ under averaging (by the Law of Large Numbers / Central Limit Theorem). We adopt AWGN as the standard model for thermal noise and wideband jammers and the maximum-entropy worst case at a given power budget; the boundary cases (correlated or biased interference) are discussed in Section IV-F, and a silence-for-'0' variant is analyzed in Section IV-D. The legitimate signal $s(t)$ is corrupted by jamming noise:

$$r(t) = s(t) + n_{\text{jam}}(t)$$

where $n_{\text{jam}}(t)$ is any zero-mean, finite-variance interference process; in our experiments $n_{\text{jam}}(t) \sim \mathcal{N}(0, \sigma)$ with $\sigma = 1$.

D. Message Decoding

The receiver computes the ensemble average $M_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} r_i(t)$ across N aligned repetitions, then takes the FFT magnitude $Mag = |\text{FFT}(M_{\text{avg}})|$. After averaging, any zero-mean interference component—the jammer's noise and the '0'-symbol noise alike—contributes a residual broadband floor that scales as $1/\sqrt{N}$, while a deterministic sinusoid at '1' positions produces a sharp peak at f_0 that is preserved under averaging. Sinusoidal signals exhibit distinct peaks at their fundamental frequency, whereas broadband zero-mean noise manifests as a wide and irregular distribution of energy; this structural difference is what the receiver exploits. We min-max normalize the magnitude to $[0, 1]$,

$$Mag_{\text{norm}} = \frac{Mag - \min(Mag)}{\max(Mag) - \min(Mag)},$$

and decode each bit segment by thresholding: if the fraction of Mag_{norm} samples exceeding t (we use $t = 0.5$) is large enough, the segment is '1', otherwise '0'. Algorithm 1 summarizes the steps of *NoisyCode*.

III. *NoisyCode* Experimental Evaluation

In this section, we describe the experimental setup of *NoisyCode*, explain how synchronization of noisy signal instances is recovered using cross-correlation, and present the results obtained under different jamming powers.

Algorithm 1 *NoisyCode*

1: Encode '1' as a sine waveform and '0' as zero-mean noise (AWGN in our implementation)
2: **if** Jammer is detected **then**
3: Tx transmits N repetitions: $M_1, M_2, \ldots, M_N$
4: **else**
5: Tx transmits a single instance of M; **goto** step 8
6: **end if**
7: **if** communication is synchronized **then**
8: Rx stacks N received repetitions R
9: **else**
10: Apply Algorithm 2 to align repetitions
11: **end if**
12: Average repetitions to obtain M_{avg}
13: Compute and normalize FFT magnitude Mag_{norm} of M_{avg}
14: **for** each bit segment of M_{avg} **do**
15: **if** percentage of $Mag_{norm} > t$ **then**
16: Decode as '1'
17: **else**
18: Decode as '0'
19: **end if**
20: **end for**
21: Output decoded message M

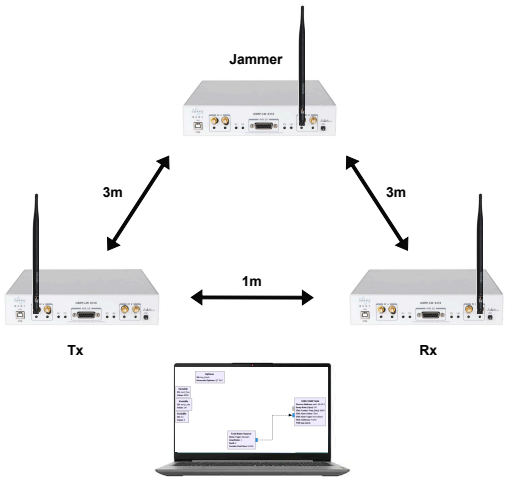


Fig. 3: Experimental setup with transmitter (Tx), receiver (Rx), and jammer positions.

A. Experimental Setup

Figure 3 shows the experimental setup: three USRP devices configured as Tx, Rx, and jammer, in line of sight in an open area, with Tx and Rx 1 m apart and the jammer 3 m from both. GNURadio is used to configure the radios and record the signals. The transmitted message is the word "HELLO," encoded with *NoisyCode* (binary '1' as a sine wave, binary '0' as AWGN). We use a carrier frequency of 900 MHz and a sampling rate of 1 MSps, collecting 100 million samples per scenario. Each bit spans 100 samples (0.1 ms), so the 40-bit

"HELLO" (01001000010001010100110001001100010011111) is transmitted in 4 ms. In separate experiments, the jammer emits normalized powers of 0.6 (2.8 dBm, 1.9 mW), 0.8 (9 dBm, 7.94 mW), and 1 (15 dBm, 31.62 mW).

B. Synchronization of Noisy Signal Instances

Under strong jamming, offsets can prevent correct alignment of message boundaries. *NoisyCode* uses a predefined fixed message length and a set number of repetitions whenever jamming is detected [4], letting the receiver segment and average incoming signals despite interference. When framing is disrupted further, cross-correlation—a standard time-delay estimation technique used in radar [5], telecommunications [6], and speech processing [7]—realigns noisy repetitions. As shown in Algorithm 2, for a message of length x bits, received signals are segmented into $2x$-sample blocks, cross-correlated for best alignment, and zero-padded before averaging, so that each averaged block contains at least one complete instance of the message.

Algorithm 2 Cross-correlation for Unsynchronized Messages

Input: Noisy repetitions $R = \{M_1, \ldots, M_n\}$
Output: Aligned messages list R
Select the first message M_r as reference
for each message $M \in R$ **do**
    Compute correlation $c \leftarrow \text{correlate}(M_r, M)$
    Find lag $l \leftarrow \text{argmax}(c) - (\text{len}(M) - 1)$
    Align M by rolling it with lag l
    if $l > 0$ **then**
        Zero out first l samples
    else if $l < 0$ **then**
        Zero out last $|l|$ samples
    end if
    Append aligned M to R
end for
Return aligned R to Step 4 of Algorithm 1

C. Experimental Results

We focus on the unsynchronized case, where cross-correlation aligns noisy instances before averaging; with synchronized Tx/Rx, fewer repetitions suffice. In Figure 4a, the top subplot shows the noisy "HELLO" message under jamming power 0.6, the second the cleaned message after averaging 20 repetitions, and the third the per-bit FFT magnitude with the decoding threshold applied. Figures 4b and 4c repeat the analysis at 0.8 and 1: lower jamming leaves parts of the signal partially visible (e.g., the last four '1's at 0.6), whereas higher jamming completely obscures the message until enough repetitions are averaged.

After averaging, the jammer's interference and the '0'-symbol noise both decay while the coherent '1' sinusoid remains prominent. Quantitatively, on 100 M samples per power level (300 M total) divided into 80,000-sample segments (20 repetitions each), *NoisyCode* achieves 99% accuracy at jamming power 0.6, 98% at 0.8, and 91% at 1; doubling to 40 repetitions raises accuracy at power 1 to 97%, and 60 repetitions reach 98%.

D. Experiments with NoisyCode Protocol

Using the setup of Section III-A, we evaluated the impact of message repetitions on Bit Error Rate (BER) under different jamming powers, comparing *NoisyCode* against standard BPSK. Figure 5 shows that *NoisyCode* achieves substantially lower BER, validating its resilience to strong AWGN interference. For jamming powers 0.6, 0.8, 1, BPSK yields BERs of 0.02, 0.90, 0.99—i.e., it collapses under full-band jamming. *NoisyCode*, in contrast, holds at 0.01/0.02/0.09 with 20 repetitions, 0.01/0.02/0.03 with 40, and 0.01/0.01/0.02 with 60. Higher jamming powers thus require more repetitions for near-perfect decoding, but ensemble averaging keeps critical low-rate messages decodable even where conventional protocols fail.

IV. Discussion

A. Bit Rate (bps) Analysis

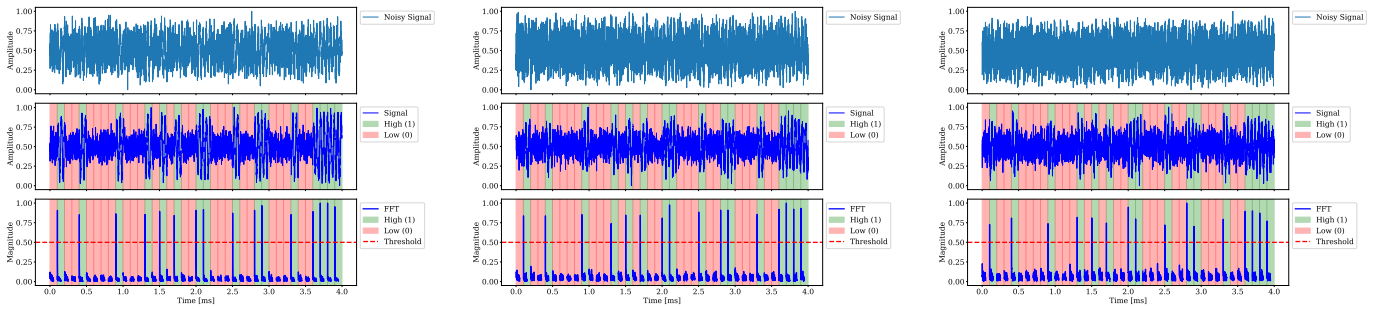
Repetitions trade throughput for robustness. Table I reports effective rates with 20 or 60 repetitions across representative link technologies. The resulting rates suffice for alarms, keep-alives, and short commands in low-rate deployments (LoRa/NB-IoT/SCADA); in high-rate systems, *NoisyCode* can opportunistically drop repetitions when the channel is clean and increase them only when jamming is detected.

B. Accuracy vs. Latency Trade-off

More repetitions trade latency for reliability: N=20 already delivers high accuracy, while larger N approaches perfect recovery under severe jamming. Because *NoisyCode* repeats correlated copies, the post-averaging message stays coherent; sending more distinct bits, as in conventional schemes, instead scatters the payload.

C. Comparison with Coherent Schemes

The BPSK curve in Fig. 5 is single-shot and shows what conventional non-repeated detection delivers under full-band jamming. A coherent BPSK receiver could in principle exploit N repetitions via matched-filter combining, gaining 3 dB in E_b/N_0 over noise-vs-waveform detection thanks to BPSK's antipodal $2A$ versus *NoisyCode*'s effective A separation. *NoisyCode*'s contribution lies along three different axes: (i) the receiver needs only an FFT magnitude and a threshold—no carrier-phase recovery or matched filter; (ii) energy detection in the f_0 FFT bin tolerates phase drift across repetitions; (iii) the noise-for-'0' encoding hides the bit pattern from a passive energy detector, a property no symmetric coherent modulation provides. *NoisyCode* thus trades a small theoretical BER margin for a simpler transceiver and basic covertness.



(a) Jamming power = 0.6. (b) Jamming power = 0.8. (c) Jamming power = 1.

Fig. 4: "HELLO" message sent using *NoisyCode* under different jamming powers, reconstructed via ensemble averaging of 20 repetitions with FFT-based decoding.

TABLE I: Impact of 20 and 60 repetitions on different communication systems and protocols.

Category	**Protocol**	**Data Rate (Baseline)**	**Effective Rate (20 / 60 Repetitions)**
High-Rate Systems	Wi-Fi 6, Advanced LTE/5G	1–5 Gbps	50–250 Mbps / 16.7–83.3 Mbps
	Basic Wi-Fi, Legacy LTE	1–100 Mbps	50 kbps–5 Mbps / 16.7 kbps–1.67 Mbps
Low-Rate Systems	Bluetooth Classic, BLE	125 kbps–2 Mbps	6.25 kbps–100 kbps / 2.08 kbps–33.3 kbps
	LoRa, NB-IoT, SCADA	11–50 kbps	550 bps–2.5 kbps / 183 bps–833 bps

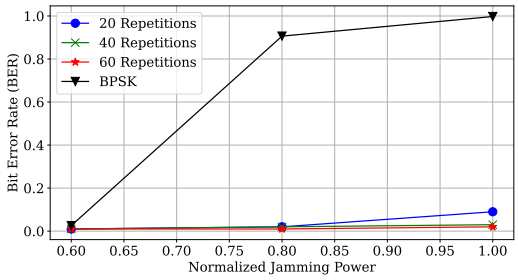


Fig. 5: Bit Error Rate (BER) of *NoisyCode* and BPSK under varying jamming powers, showing the relationship between jamming power, BER, and message repetitions.

D. The Silence/AWGN Trade-off and Covertness

Encoding '0' as AWGN rather than silence appears to waste transmit energy, but it preserves covertness. With AWGN power matched to the sinusoid ($\sigma_{tx}^2 \approx A^2$), per-bit received power is $A^2 + \sigma_j^2$ at both symbol positions, so a passive total-power detector cannot recover the bit pattern. Structurally distinguishing the symbols requires narrowband detection at f_0, knowledge of the repetition cadence, and ensemble averaging over N aligned repetitions—operations the legitimate receiver already performs, giving it a $\sqrt{N}$ advantage.

A silence-for-'0' variant ($s_0(t) = 0$) trades covertness for a lower post-averaging residual variance (σ_j^2/N instead of $(\sigma_j^2 + \sigma_{tx}^2)/N$) and zero transmit energy on '0' bits, but exposes the on/off energy pattern to passive detection. *NoisyCode* is therefore a family of operating modes along an energy/decoding/covertness trade-off: AWGN-for-'0' when covertness matters, silence-for-'0' when it does not, or a hybrid that pseudorandomly selects between them per '0' bit using a pre-shared key, so the adversary cannot predict which bits appear silent. A formal security analysis of these variants, possibly combined with a lightweight cipher above the physical layer, is left for future work.

E. Robustness to Channel Impairments and Smarter Jammers

The reported experiments use a static line-of-sight setup. In multipath or mobile environments, phase rotation across repetitions degrades the coherent build-up; mitigations include a '1' waveform short relative to the coherence time, non-coherent power accumulation, or hopping f_0 so each repetition fades independently. Reactive or waveform-aware jammers can exploit *NoisyCode*'s predictable repetition cadence; randomizing inter-repetition timing, varying f_0, or interleaving *NoisyCode* with a lightweight FHSS layer are natural defenses. These directions, together with stress tests of the cross-correlation realignment under heavy jamming, are left for follow-up work.

F. Boundary of Applicability: Same-Slice Replay and Biased Interference

NoisyCode's averaging suppresses any interference that is zero-mean and uncorrelated across the N repetitions; two regimes fall outside this assumption. In the *same-slice replay* limit, the jammer's contribution is identical across all repetitions and survives averaging as a deterministic term—precisely the regime where classical interference-cancellation techniques (notch filtering, adaptive subtraction, interference-cancellation receivers) apply; they are *orthogonal and complementary* to *NoisyCode* and can be composed with it as a pre-processing stage. For interference with a *time-varying non-zero mean*, averaging converges to the instantaneous mean rather than to zero, leaving a residual bias that must be removed (e.g., by a DC/mean-tracking pre-filter). A formal characterization under colored, correlated, and biased interference is left for future work.

V. Related Work

Ensemble averaging is a classical technique for improving effective SNR by combining repeated observations [8], applied for instance to denoising repeatedly captured images [9] and revealing Gaussian-noise-obfuscated Radio-frequency (RF) fingerprints [10]. *NoisyCode* adapts this principle at the symbol level: repetitions suppress any zero-mean, uncorrelated interference while reinforcing deterministic waveforms.

Jamming attacks degrade SNR through spot, sweep, barrage, and reactive or deceptive strategies [11], [12], [13], [14], with AWGN jamming particularly prevalent [1] and the canonical worst case at a given power budget. Common countermeasures—frequency hopping, spectrum spreading, adaptive filtering, MIMO defenses, channel coding, and learning-based classifiers [3], [15], [16], [17], [18], [19]—work in their target regimes but can demand substantial complexity, bandwidth, or hardware. By contrast, *NoisyCode* embraces stochastic interference: encoding '0' as zero-mean noise and '1' as a coherent waveform makes averaging both suppress interference and hide the bit pattern from passive energy detection.

Repetition coding is far from capacity-achieving: since Shannon and Hamming, the community has developed channel codes (BCH, convolutional, turbo, Low-Density Parity-Check (LDPC), polar) that approach capacity with much lower redundancy. We do not claim that *NoisyCode* is information-theoretically optimal. Rather, its integration of waveform-vs-noise encoding, ensemble averaging, and FFT thresholding gives a complete, hardware-minimal protocol that survives full-band stochastic jamming and provides physical-layer covertness on off-the-shelf SDRs without carrier-phase recovery. Relative to repetition-coded coherent BPSK, it trades a small BER margin for a simpler receiver and covertness (see Section IV-C); relative to FHSS/DSSS, it needs no synchronized hopping or chip sequence; relative to capacity-approaching codes, it needs no decoder beyond an FFT and a threshold. Combining *NoisyCode* with a lightweight outer code is left for future work.

VI. Conclusion

This paper presented *NoisyCode*, a binary protocol for jamming-resilient transmission that encodes '1' as a deterministic waveform and '0' as zero-mean noise (with AWGN as the canonical case). Ensemble averaging at the receiver suppresses both the jammer's interference and the '0'-symbol noise while reinforcing the coherent '1' waveform, enabling reliable decoding under full-band jamming with low complexity. Matching the per-symbol energies of '1' and '0' adds a covertness property: the on-air signal exhibits no on/off energy pattern, hiding the bit stream from passive energy-detection adversaries. A silence-for-'0' variant trades that covertness for transmit-energy efficiency, so *NoisyCode* is best viewed as a small family of operating modes; when interference is correlated or biased, *NoisyCode* composes naturally with classical interference-cancellation techniques. USRP experiments validated the design: *NoisyCode* achieved 91% accuracy under 15 dBm jamming with 20 repetitions, rising to 97–98% with 40–60. The tunable repetition count provides a direct reliability–latency trade-off, making *NoisyCode* a simple, low-cost, scalable option for delivering critical low-bandwidth messages under interference.

References

[1] R. Poisel, *Modern communications jamming principles and techniques*. Artech house, 2011.

[2] O. A. Ibrahim and R. Di Pietro, "Mag-auth: Authenticating wireless transmitters and receivers on the receiver side via magnetic emissions," in *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2023, pp. 305–316.

[3] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[4] O. A. Ibrahim and R. Di Pietro, "Mag-jam: Jamming detection via magnetic emissions," in *European Symposium on Research in Computer Security*. Springer, 2024, pp. 167–186.

[5] L. Pallotta and G. Giunta, "Accurate delay estimation for multisensor passive locating systems exploiting the cross-correlation between signals cross-correlations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 2568–2576, 2021.

[6] S. Mikki, D. Sarkar, and Y. Antar, "Near-field cross-correlation analysis for mimo wireless communications," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 7, pp. 1357–1361, 2019.

[7] J. Benesty, J. Chen, and Y. Huang, "Time-delay estimation via linear interpolation and cross correlation," *IEEE Transactions on speech and audio processing*, vol. 12, no. 5, pp. 509–519, 2004.

[8] C. L. Wilkins, S. P. Perone, C. E. Klopfenstein, R. C. Williams, and D. E. Jones, *Ensemble Averaging of Repeatable Noisy Signals*. Boston, MA: Springer US, 1975, pp. 121–128. [Online]. Available: https://doi.org/10.1007/978-1-4615-8720-0_11

[9] G. Gisbert, N. Dey, H. Ishikawa, J. Schuman, J. Fishbaugh, and G. Gerig, "Improved denoising of optical coherence tomography via repeated acquisitions and unsupervised deep learning," *Investigative Ophthalmology & Visual Science*, vol. 61, no. 9, pp. PB0035–PB0035, 2020.

[10] O. A. Ibrahim and R. Di Pietro, "Echo: On the limitations of noise-based radio fingerprinting obfuscation," in *2025 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2025.

[11] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 24, no. 2, pp. 767–809, 2022.

[12] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46–57.

[13] M. Schulz, F. Gringoli, D. Steinmetzer, M. Koch, and M. Hollick, "Massive reactive smartphone-based jamming using arbitrary waveforms and adaptive power control," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 111–121.

[14] P. Wu, H. Calatrava, T. Imbiriba, and P. Closas, "Jammer classification with federated learning," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2023, pp. 228–234.

[15] T. Karhima, A. Silvennoinen, M. Hall, and S.-G. Haggman, "Ieee 802.11 b/g wlan tolerance to jamming," in *IEEE MILCOM 2004. Military Communications Conference, 2004.*, vol. 3. IEEE, 2004, pp. 1364–1370.

[16] M. Ding, W. Chen, D. Weng, and X. Mi, "Adaptive jamming mitigation in single-antenna receivers with spectral analysis and switchable filtering," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[17] Z. Wang, J. Zhang, H. Du, D. Niyato, S. Cui, B. Ai, M. Debbah, K. B. Letaief, and H. V. Poor, "A tutorial on extremely large-scale mimo for 6g: Fundamentals, signal processing, and applications," *IEEE Communications Surveys & Tutorials*, 2024.

[18] K. Davaslioglu, S. Soltani, T. Erpek, and Y. E. Sagduyu, "Deepwifi: Cognitive wifi with deep learning," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 429–444, 2019.

[19] S. Di Pinto, P. Locatelli, P. Spadaccino, and F. Cuomo, "Detection and mitigation of jamming attacks in lorawan using machine learning," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2025, pp. 1–6.