U-FLASH: RELIABLE UNDERWATER COMMUNICATION USING SMARTPHONES

Improving Underwater Optical Communication by the Scattering Effect

Figma Link: https://www.figma.com/design/J2s4cY8vrIYeqUgGEslain/Untitled?node-id=1-2&t=iA8tPvINUfs758SC-1

TEAM: BrahmandGurus

Amrit Ghoshal (2024A7PS0484G)

Akshat Agrawal (2024AAPS0787G)

Saarang Agarwal (2024A7PS0587G)

Aman Kumar (2024A4PS0414G)

PROBLEM: THE UNDERWATER BARRIER

01

Hardware Constraints

Acoustic communication requires dedicated, large, and expensive equipment. RF technologies (Wi-Fi, LTE) fail due to extreme underwater attenuation.

02

Limited Optical Range

Current smartphone optical systems (using the screen as a source) are highly restricted, often limited to a mere 0.15 m maximum communication distance.

03

Severe Channel Dynamics

Turbulent water flow, particulates, and ambient sunlight cause rapid, unpredictable variations in image brightness and signal detectability.

04

Hardware Restrictions

How do we effectively increase range and reliability using only the built-in smartphone components (Flashlight and Camera) without external mods?

SOLUTION: EMBRACING THE ELEMENTS

01

Exploiting the Rolling Shutter

We utilize the CMOS sensor's rolling shutter effect to capture incredibly fast LED pulses as alternating bright and dark spatial stripes across a single frame.

02

Benefiting from Scattering

Instead of fighting water turbidity, we embrace it. Rayleigh scattering expands the light's Region of Interest, allowing more data bits to be captured per frame.

03

Advanced Software Pipeline

A robust software architecture handles encoding (PPM, RLL) and dynamic decoding (Adaptive Thresholds, Viterbi) strictly through software without external hardware.

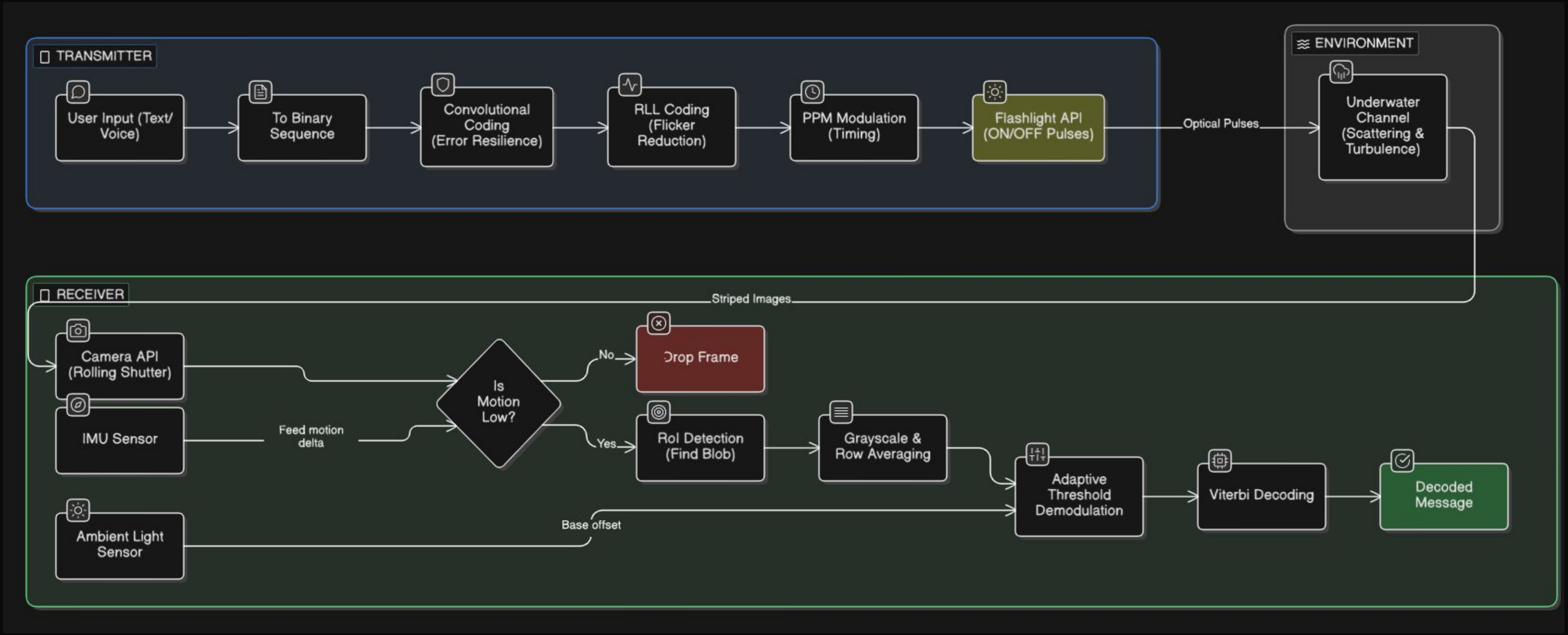
04

Sensor-Assisted Reliability

Utilizing the phone's IMU and Ambient Light sensors to dynamically drop blurry frames and intelligently adjust detection thresholds on the fly.

END-TO-END PIPELINE

A high-level overview of how raw text is converted to light pulses (TX), passes through the highly variable water channel, and is successfully decoded back to text via the smartphone camera (RX).



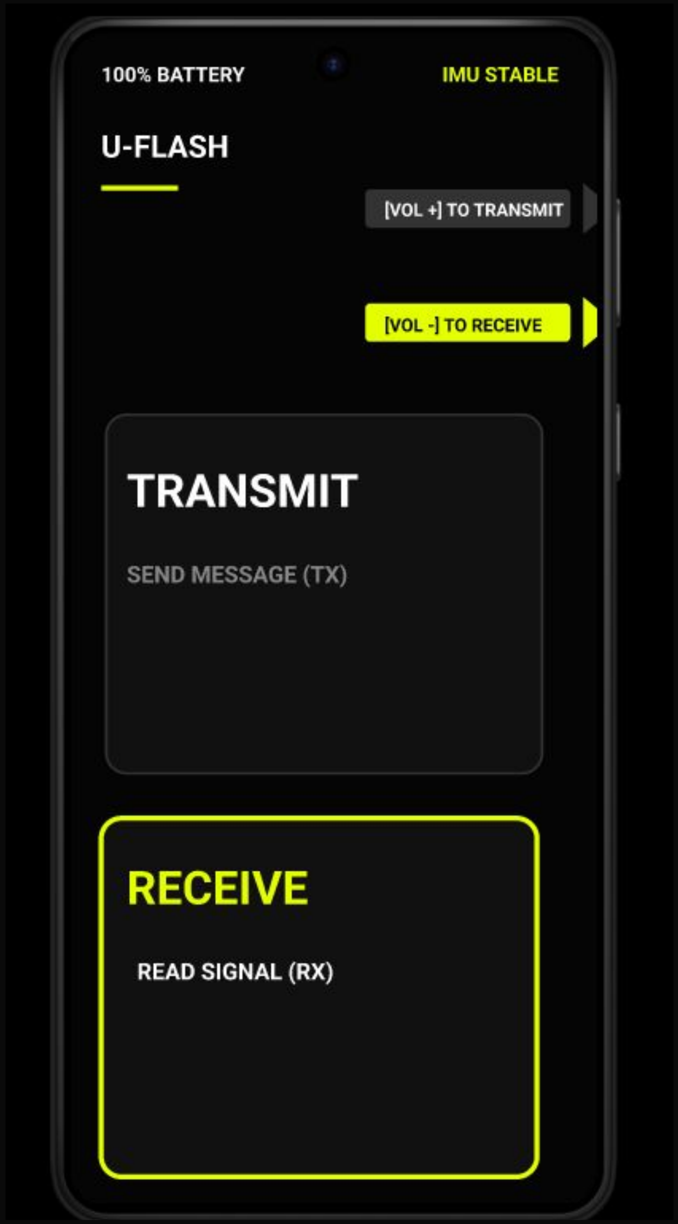
UI: DESIGNED FOR THE DEEP

Touchless Navigation

Capacitive touchscreens fail under water pressure. The user interface is 100% driven by the physical Volume Up and Volume Down hardware keys for foolproof operation.

High Visibility

A deep background combined with highly saturated Neon Yellow accents ensures maximum readability and contrast through a foggy or tinted diving mask.



ENHANCED UX:
RAPID RESPONSE

01 **Pre-Saved Quick Phrases**

Typing underwater is heavily restricted. The app features a customizable library of high-probability commands (e.g., "Low Air", "Ascend"). Divers simply scroll and transmit instantly without typing.

02 **Zero Unlock Hardware**

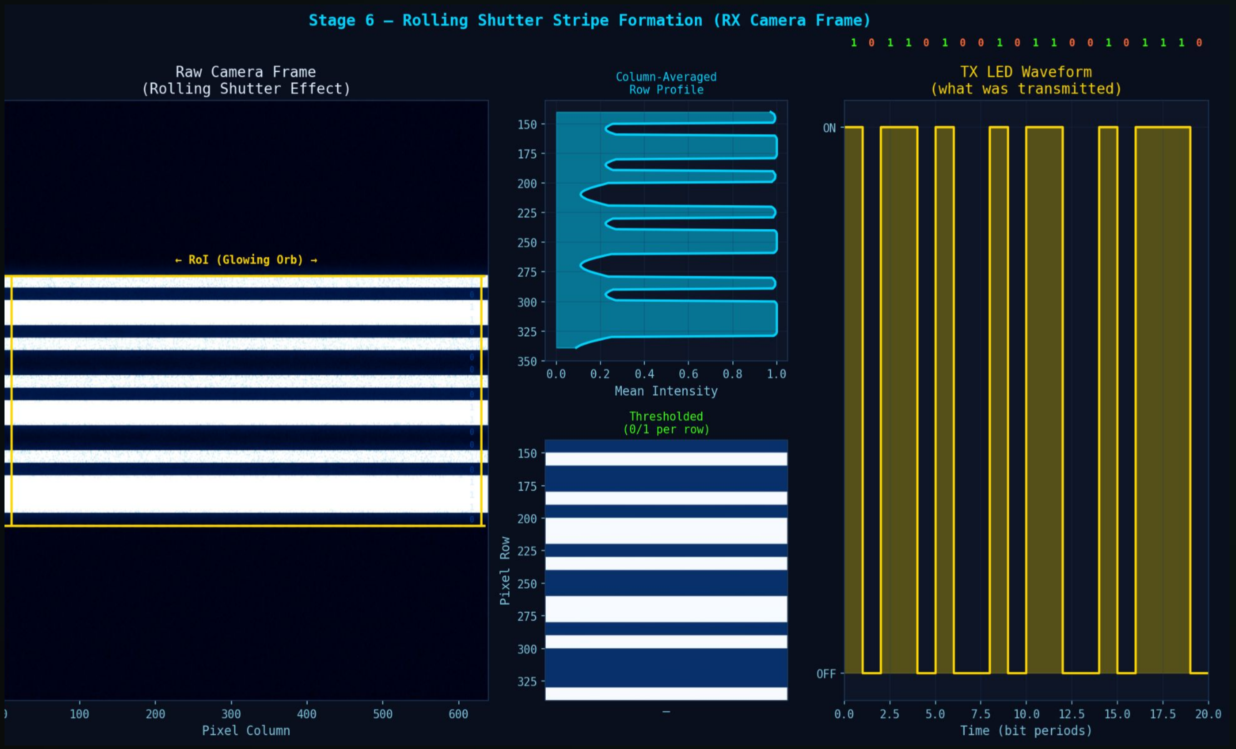
To bypass touch screen limitations, the app is "armed" before entry. A single press of the physical Power Button instantly brings U-FLASH to the foreground, displaying the Quick-Select menu.

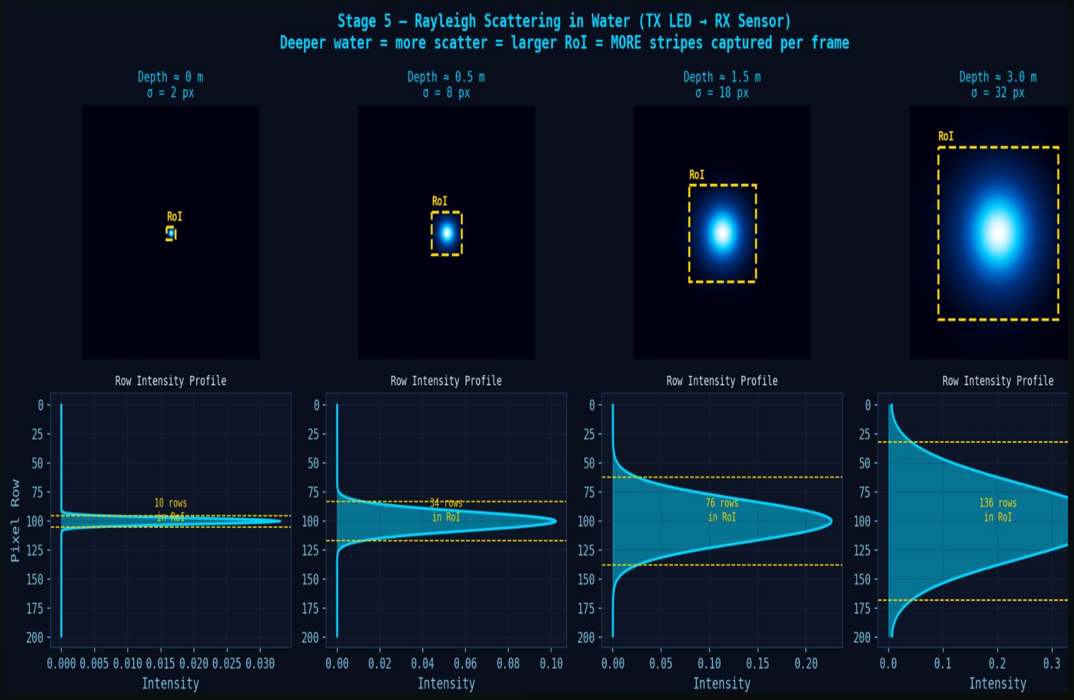
03 **Tactical Mode Switching**

For specific, non-standard messages, a secondary hardware trigger (e.g., Double-tapping Power or long-pressing Volume Down) seamlessly toggles the interface to "Custom Input" mode.

PHYSICS: ROLLING SHUTTER & RAYLEIGH SCATTERING

- **Transmitter:** The flashlight toggles ON/OFF rapidly to encode data.
- **Receiver:** The camera captures these temporal pulses as spatial stripes.
- **The Multiplier:** Rayleigh scattering disperses the light, enlarging the glowing orb on the sensor so we can extract more bits per frame.

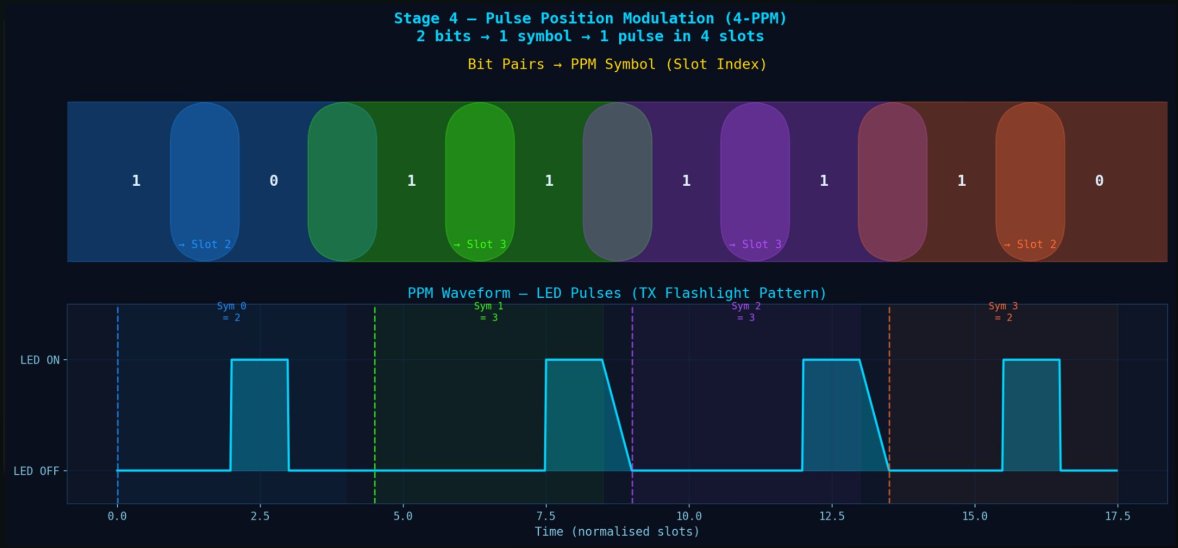




TX PIPELINE: COMBATING FLICKER

> **Run-Length Limited (RLL) Coding:** Caps the maximum number of consecutive 0s or 1s to prevent the flashlight from visibly flickering to the human eye, maintaining a steady appearance.

> **Pulse Position Modulation (4-PPM):** Maps the encoded bits to specific, predefined time slots to combat and mitigate the effects of chromatic dispersion as light travels through water.



RX PIPELINE: ADAPTIVE THRESHOLDING

A hard-coded detection threshold instantly fails underwater due to rippling water surfaces and dynamic ambient sunlight.

Instead, we utilize a sliding window technique to calculate the local mean and variance. This allows the system to dynamically adjust the 0/1 decision threshold for every single pixel row on the fly.

SENSORS: ENVIRONMENTAL ADAPTATION

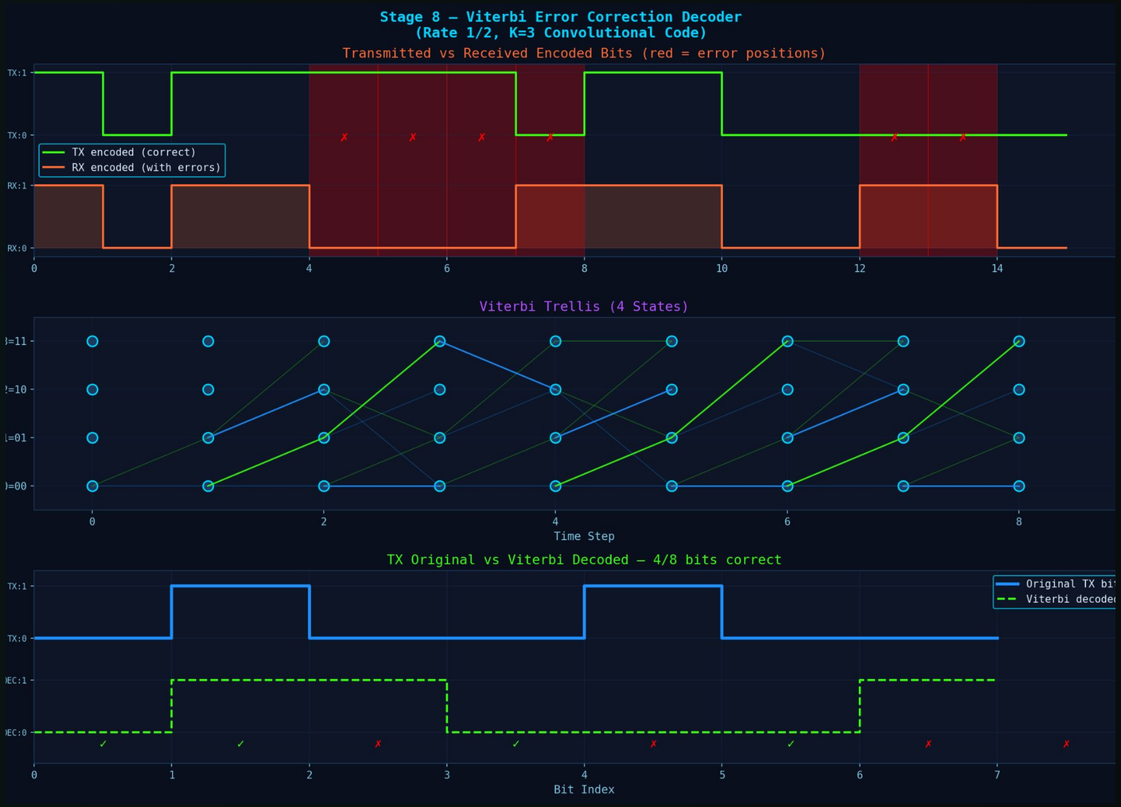
> **IMU Motion Gating:**
> The smartphone's gyroscope detects excessive hand shaking and intentionally drops the corrupted frame before processing, preventing severe inter-symbol interference.

> **Ambient Light Feedback:**
> The built-in light sensor detects sudden spikes in background illumination (like a passing sun ray) and offsets the baseline decision threshold accordingly.

VITERBI DECODING FOR BURST ERRORS

The highly volatile underwater channel frequently causes clustered "burst" errors that overwhelm basic parity checks.

By applying an underwater-optimized Convolutional Code at the transmitter, our software Viterbi decoder maps out a trellis to mathematically find the most likely original path, successfully correcting heavily corrupted bits.

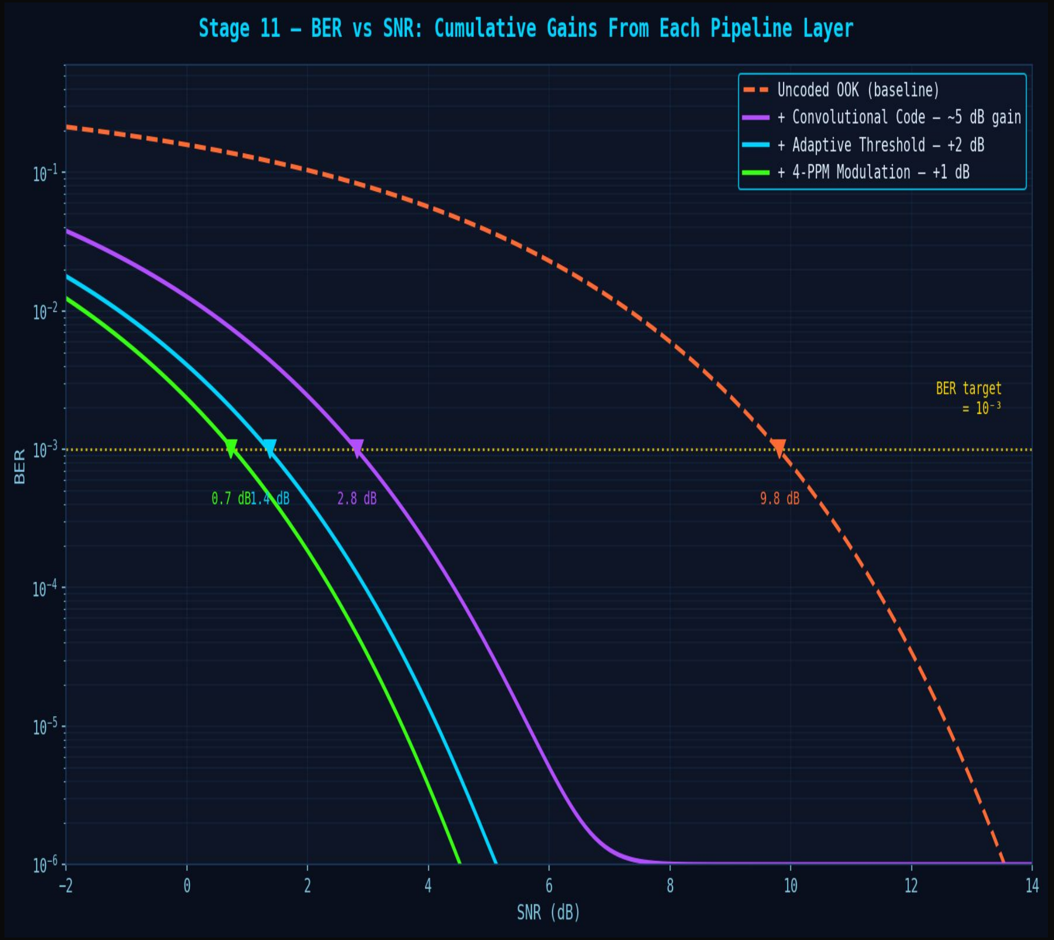


CUMULATIVE RELIABILITY GAINS

This graph demonstrates the significant reduction in the Bit Error Rate (BER) across different signal-to-noise ratios.

Each software layer—from standard OOK implementation to Convolutional Coding to our Adaptive Thresholds—dramatically shifts the performance curve.

This proves that robust software engineering alone can drastically improve communication reliability without relying on external lenses or bulky equipment.



TECHNOLOGIES BEING USED

Camera2 API

Grants low-level Android access to lock Auto-Exposure and Auto-Focus, forcing the extremely fast shutter speeds required to capture temporal stripes.

C++ & NDK

Forms the backbone of the heavy signal processing. C++ handles real-time matrix operations and OpenCV tracking without dropping sensor frames.

Flutter

Powers the reactive UI layer and utilizes Method Channels to seamlessly route Voice-to-Text inputs and raw hardware sensor data (IMU motion gating, Ambient Light threshold offsets).

Figma

Enabled rapid, modern UI prototyping specifically tailored for harsh visual environments, focusing on extremely high contrast and touchless navigation.