Smart Waste Bin Monitoring with QAM Inspired Complex Representation for Scalar Sensor Data: Bridging IoT, Signal Processing, and Communication

R Janaki, G R Shivaranjani, V Shreya
Department of Electronics and Communication Engineering
VIT CHENNAI

***Abstract*—Traditional analysis of scalar IoT sensor data such as that from smart dustbins which process fill level of waste, number of usages per day, and automated lid opening relies on direct time domain processing (averaging, differentiation, or thresholding). However, these scalar only approaches fail to capture the interplay between instantaneous state and its rate of change. This paper proposes a novel Quadrature Amplitude Modulation (QAM) inspired complex representation that maps scalar sensor data into the complex plane, enabling a new class of hybrid analysis techniques combining signal processing, machine learning, and communication theory. A complete Smart Waste Bin prototype was developed using ESP32, ultrasonic and IR sensors, and in-built WiFi module for message transmission. MATLAB-based post-processing of sensor readings using the QAM-inspired mapping revealed enhanced insight into waste bin dynamics and system behavior.**

***Index Terms*—Smart Waste Bin, QAM, IoT, Signal Processing, Complex Representation, Data Modulation, Communication Systems, Scalar Sensor Data, ESP32, MATLAB, Ultrasonic Sensor, Machine Learning, Phase Analysis, Feature Extraction, IoT Analytics.**

I. INTRODUCTION

IoT-based smart waste management has gained immense importance as cities transition toward automation and environmental sustainability. Conventional waste bin systems rely on scalar sensor data such as fill-level and usage count for operation, but traditional methods like thresholding and time-domain averaging cannot capture dynamic variations over time. These scalar approaches lack a multidimensional feature space to describe both instantaneous and temporal behavior.

Inspired by Quadrature Amplitude Modulation (QAM) from communication theory, this research introduces a complex- valued analytical representation of scalar sensor data. The proposed representation transforms each data sample into a complex number, combining magnitude and rate-of-change information. This hybrid domain bridges IoT sensing with modulation theory, paving the way for a new signal-symbol analytic framework.

II. LITERATURE REVIEW

Numerous studies have explored IoT-based smart bin systems and advanced sensor analytics over the past decade. However, despite progress in sensor design, network communication, and data integration, most approaches remain limited to scalar analysis, overlooking the intrinsic temporal–complex relationships in sensor behavior. In [1], Sklar outlined foundational digital modulation techniques that shaped modern signal representation and communication theory, later inspiring analytical frameworks across domains. The rapid growth of the Internet of Things (IoT) and its wide-ranging applications were reviewed in [2], emphasizing the transformative role of ubiquitous connectivity and sensor-driven automation.

Early IoT-based smart waste management systems, such as those developed in [3]–[5], relied mainly on ultrasonic sensors integrated with cloud platforms for monitoring fill levels. In [6], Wi-Fi modules were added for real-time notifications, improving system responsiveness. Broader discussions in [7] and [8] highlighted the need for intelligent data interpretation beyond basic thresholding and reporting, pointing to the untapped potential of sensor signal analytics in connected environments.

Signal processing perspectives on sensor data were introduced by Kumar et al. [9], showing how mathematical transformations improve accuracy and reliability. Similarly, [10] demonstrated feature transformation mechanisms for temporal data, while [11], [12] emphasized the use of derivative- based models for real-time responsiveness. Works such as [13] and [14] applied concepts from communication theory to encode information dynamically, drawing analogies between modulation schemes and feature extraction in sensor analytics. The role of machine learning in IoT data processing was reviewed comprehensively in [15], and dynamic feature encoding strategies were discussed in [16]. Hardware-focused implementations using ESP32 and Wi-Fi modules for real world waste monitoring were demonstrated in [17], [18], providing scalable and low-cost architectures. The idea of representing sensor behavior geometrically and analytically was further advanced in [19], [20], showing that multidimensional modeling enhances interpretability and sensitivity to subtle changes. In parallel, complex-valued neural computation was explored in [21] and extended by [22] into complex-domain and learning, reinforcing the value of representing both magnitude phase in dynamic systems.

Integration of IoT data into hybrid analytical frameworks was examined in [23], linking communication theory and data science. More recent studies [24], proposed feature fusion techniques inspired by modulation principles, showing how ideas from digital communications can inform richer data interpretation. Collectively, these works highlight a major research gap: most IoT sensor systems treat data as discrete, isolated points rather than as

continuous signals evolving over time. This limits the ability to capture dynamic variations such as transients or oscillatory trends. To overcome this, the present paper introduces a QAM-inspired analytical framework that models scalar sensor readings as complex-valued signals. This enables temporal–complex feature extraction, offering a richer and more nuanced interpretation of sensor dynamics within IoT systems.

III. Hardware Implementation

The prototype system comprises multiple hardware modules for sensing, control, and communication.

A. System Overview

The system employs an ultrasonic sensor to detect bin fill- level, an IR sensor to sense user proximity, and a servo motor for lid control. An ESP32 microcontroller manages sensor interfacing and data communication, while a Wi-Fi module transmits notifications when the bin reaches full capacity. The sensor readings form a time series input for QAM-inspired complex data analysis.

B. Hardware Components

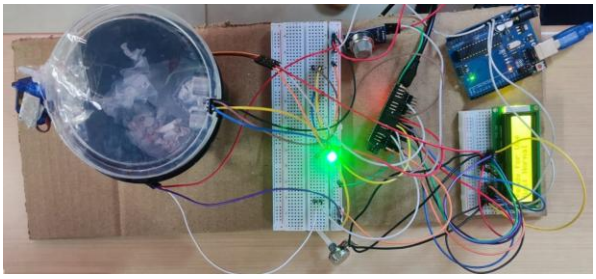
- **Ultrasonic Sensor (HC-SR04):** Measures fill-level using time-of-flight.
- **IR Sensor (LM393):** Detects object proximity for lid automation.
- **Servo Motor:** Opens/closes bin lid automatically.
- **ESP32:** Central controller handling data computation and transmission.
- **In-built Wi-Fi module:** Sends alerts to registered users.
- **LCD Display:** Displays bin status and operational data.
- **Power Supply Module:** Regulated 5V DC power delivery.
- **Gas Sensor:** Alerts when toxic gas is detected.

C. Pin Connections

Component	Pins	ESP32 GPIO
IR Sensor	Output	34
Servo Motor	Signal	13
Ultrasonic Sensor	TRIG / ECHO	25 / 33
Gas Sensor	Analog Out	32
LEDs	Green / Red	26 / 27
LCD (RS, EN)	RS / EN	23 / 22
LCD (Data Lines)	D4 / D5 / D6 / D7	18 / 19 / 21 / 5

TABLE I: Pin Connections for ESP32 Smart Dustbin

D. Physical Setup of Smart Dustbin



IV. Methodology

The methodology integrates hardware data acquisition and QAM-inspired analytical mapping to convert scalar readings into QAM Constellations translations.

A. Algorithmic Steps

1) Acquire scalar data from the ultrasonic sensor.
2) Normalize readings for uniform scaling.
3) Compute discrete derivative: $\Delta x = x_n - x_{n-1}$.
4) Form complex representation: $z_n = x_n + j\Delta x$.
5) Extract magnitude $|z_n|$ and phase ϑ_n.
6) Visualize the trajectory of z_n in the complex plane.

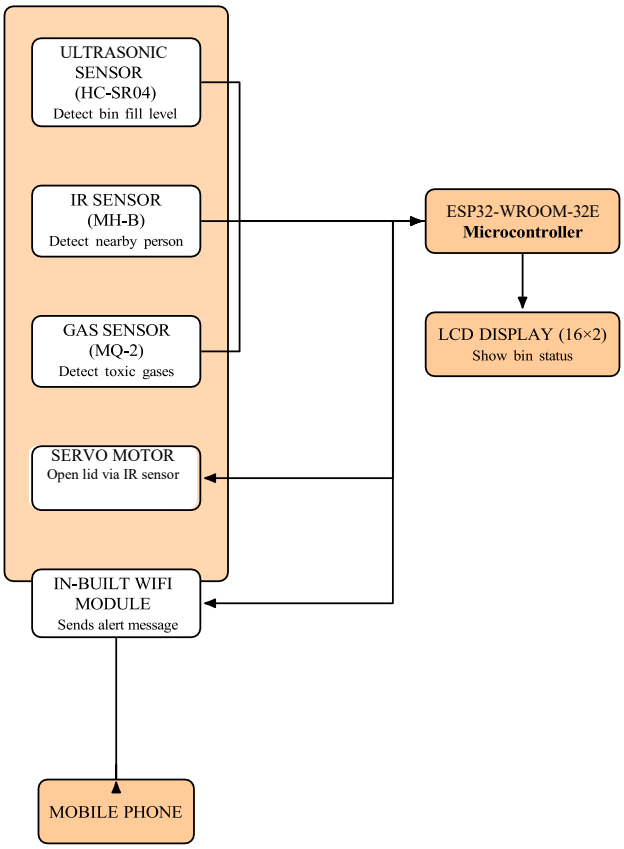


Fig. 1. Hardware Block Diagram of Smart Waste Bin Monitoring System

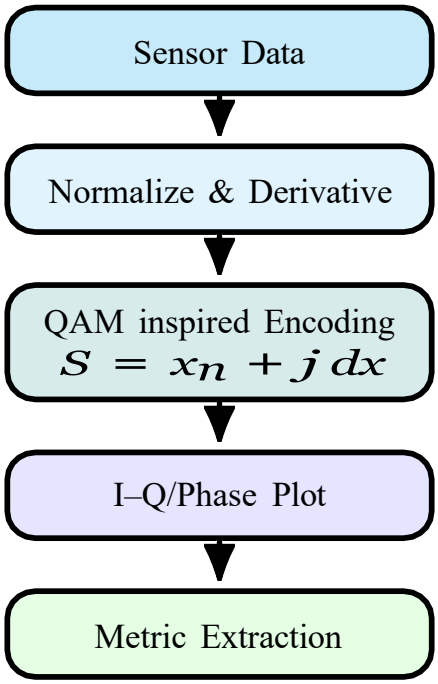


Fig. 2. QAM inspired Sensor Data Analysis Software Flow

V. Results and Discussion

The implementation of the proposed system was carried out using the ESP32-WROOM-32E module in the Arduino IDE. The corresponding code, along with the hardware setup, was tested to validate the functionality of the system. The following images illustrate the code snippets and the observed results from the setup.

A. ESP-32 Implementation Code

```
#include <ESP32Servo.h>
#include <LiquidCrystal.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// ---------- WiFi & Telegram Setup ----------
const char* ssid = "OnePlus Nord CE 3 Lite 5G";
const char* password = "w5xdea98";
#define BOTtoken "8089785003:AAF4XH3yBBO3pAVtasUMxXfTHXRMBfRxnjo"
#define CHAT_ID "8212541315"

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// ---------- Pin Definitions ----------
const int IR_PIN = 34;
const bool IR_ACTIVE_STATE = LOW;
const int SERVO_PIN = 13;

// Ultrasonic
const int TRIG_PIN = 25;
const int ECHO_PIN = 33;

// LED pins
const int GREEN_LED = 26;
const int RED_LED   = 27;

// Gas sensor
const int GAS_PIN = 32;
const int GAS_THRESHOLD = 1800;

// LCD connections
const int RS = 23;
const int EN = 22;
const int D4 = 18;
const int D5 = 19;
const int D6 = 21;
const int D7 = 5;
// ---------- Parameters ----------
const unsigned long openDuration = 3000UL;// Lid open time (ms)
const int binFullDistance = 4;// Distance (cm) to detect bin full

// ---------- Globals ----------
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);
Servo servo;

bool lidOpen = false;
unsigned long lidOpenedAt = 0;
bool binWasFull = false;
bool gasAlertSent = false;

void setup() {
  Serial.begin(115200);

  pinMode(IR_PIN, INPUT);
  pinMode(GAS_PIN, INPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);

  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(RED_LED, LOW);

  servo.attach(SERVO_PIN);
  servo.write(0);

  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("Connecting WiFi");

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  client.setInsecure();
  Serial.println("\nWiFi Connected!");
  lcd.clear();
  lcd.print("Smart Bin Ready");
  lcd.setCursor(0, 1);
  lcd.print("LID: CLOSED");

  Serial.println("System initialized. Waiting for IR detection...");
}

void loop() {
  long distance = getDistance();
  int irVal = digitalRead(IR_PIN);
  int gasValue = analogRead(GAS_PIN);

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print(" cm | IR: ");
  Serial.print(irVal == IR_ACTIVE_STATE ? "DETECTED" : "CLEAR");
  Serial.print(" | Gas: ");
  Serial.println(gasValue);

  // ----- BIN FULL -----
  if (distance > 0 && distance <= binFullDistance) {
    servo.write(0);
    lidOpen = false;
    digitalWrite(RED_LED, HIGH);
    digitalWrite(GREEN_LED, LOW);

    if (!binWasFull) {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Bin is Full!");
      lcd.setCursor(0, 1);
      lcd.print("Please empty bin");
      Serial.println("Bin full! Lid locked.");
      bot.sendMessage(CHAT_ID, "⚠️ Bin is FULL! Please empty the bin.", "");
      binWasFull = true;
    }

    delay(300);
    return;
  }
    // ----- BIN NOT FULL -----
    else {
      // when bin transitions from full → not full
      if (binWasFull) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Bin Cleared :)");
        lcd.setCursor(0, 1);
        lcd.print("LID: CLOSED");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Ready for Use");
        lcd.setCursor(0, 1);
        lcd.print("LID: CLOSED");
        binWasFull = false;
      }

      // normal IR-based opening
      if (irVal == IR_ACTIVE_STATE && !lidOpen) {
        openLid();
      }

      if (lidOpen && (millis() - lidOpenedAt >= openDuration)) {
        closeLid();
      }
    }

    // ----- GAS SENSOR STATUS -----
    if (gasValue < GAS_THRESHOLD) {
      // GAS NORMAL
      lcd.setCursor(0, 1);
      lcd.print("Gas: Normal   ");
      digitalWrite(GREEN_LED, HIGH);
      digitalWrite(RED_LED, LOW);
      gasAlertSent = false; // reset for next detection
    }
  else {
    // GAS ABNORMAL
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("⚠ GAS ALERT ⚠");
    lcd.setCursor(0, 1);
    lcd.print("Check bin area!");

    digitalWrite(RED_LED, HIGH);
    digitalWrite(GREEN_LED, LOW);

    if (!gasAlertSent) {
      Serial.println("⚠ WARNING: Gas Level HIGH!");
      bot.sendMessage(CHAT_ID, "⚠️ Abnormal Gas Detected! Please check immediately.", "");
      gasAlertSent = true;
    }
  }

  delay(500);
}

void openLid() {
  servo.write(90);
  lidOpen = true;
  lidOpenedAt = millis();

  Serial.println("Lid opened");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("LID: OPEN");
}

void closeLid() {
  servo.write(0);
  lidOpen = false;

  Serial.println("Lid closed after timeout");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Thank you :)");
  lcd.setCursor(0, 1);
    lcd.print("Have a great day!");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("LID: CLOSED");
  }

  long getDistance() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    unsigned long duration = pulseIn(ECHO_PIN, HIGH, 30000);
    if (duration == 0) return -1;

    long distance = (duration * 0.0343) / 2.0;
    return distance;
  }
```

Fig. 3. ESP-32 code implemented in Arduino IDE

B. Hardware Model Results

Fig. 4. Initialization of the Smart Bin

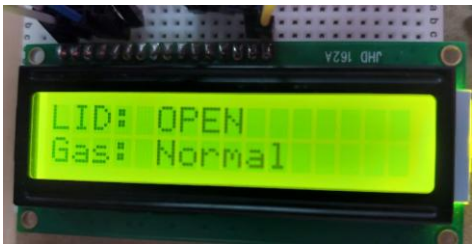


Fig. 4. Lid opened when hand movement is detected by the IR sensor

Fig. 5. Lid closed 3s after opening

Fig. 6. Bin full message when fill level < 4cm

Fig. 6. Toxic gas detected by gas sensor and message displayed when level>1800

Fig. 7. Telegram message sent to registered no. for bin full and toxic gas alert

C. MATLAB Implementation

The MATLAB implementation produced separate plots of magnitude and phase variations. Flat readings (constant fill-levels) exhibited near-zero phase change, while sudden increases in bin usage produced sharp phase shifts.

Input data (x) = [6 6 6 6 6 8 8 8 0 0 0 2 3 2 3 3 2 4 3 3 3 3 3 3 0 3 3 12 12 12 12 12 12 12];

TABLE II
DYNAMIC METRICS FROM QAM-INSPIRED SENSOR REPRESENTATION

Metric	Value	Meaning	Interpretation
Trajectory Length	6.5598	Total path traversed by the complex signal $S(t)$ in the IQ domain.	Represents overall activity of the sensor signal — indicates moderate variability in bin-level data.
Average Phase Direction	−0.0235 rad	Mean angular orientation of the complex trajectory.	A near-zero mean phase shows a steady overall trend with minimal directional deviation.
Stability-Index	σ_θ = 0.4721	Standard deviation of the instantaneous phase.	Quantifies phase stability; moderate spread suggests balanced dynamic behavior.
Activity-Index	$\lvert dx/dt \rvert_{avg}$ = 0.0810	Average rate of change in normalized sensor data.	Indicates signal responsiveness — low-to-moderate activity across sensor readings.
Event Count	7	Number of significant signal changes ($\lvert dx \rvert > 0.1$).	Represents the frequency of major transitions — seven primary fill/empty events detected.

The dynamic metrics obtained from the QAM-inspired representation provide useful insights into the bin's operation. The trajectory length and average phase direction describe the temporal behavior of sensor readings, while the stability and activity indices measure variations in fill-level and lid movements. The event count indicates significant changes in the system, such as frequent bin usage or abrupt lid actions. Overall, this data helps in understanding the bin's usage patterns and supports timely alerts for efficient waste management.

i. Graph Interpretation

MATLAB implementations reveal that the normalized sensor data shows clear step-like changes, corresponding to distinct fill and empty events reflected as sharp peaks and troughs in the derivative (rate of change) plot.

In the complex trajectory plot (Fig 5), The x-axis (In-phase component, I) represents the normalized sensor value, corresponding to the physical magnitude of the measured quantity (e.g., fill level or height). The y-axis (Quadrature component, Q) represents the rate of change of the sensor reading, capturing how rapidly the physical state varies over time. The complex trajectory and phase plots indicate moderate variability with a generally stable phase direction, con- firming that the system experiences intermittent but controlled fluctuations in sensor readings. The moderate spread of phase angles suggests a well-balanced signal behavior, indicating reliable and repeatable sensor dynamics over time.

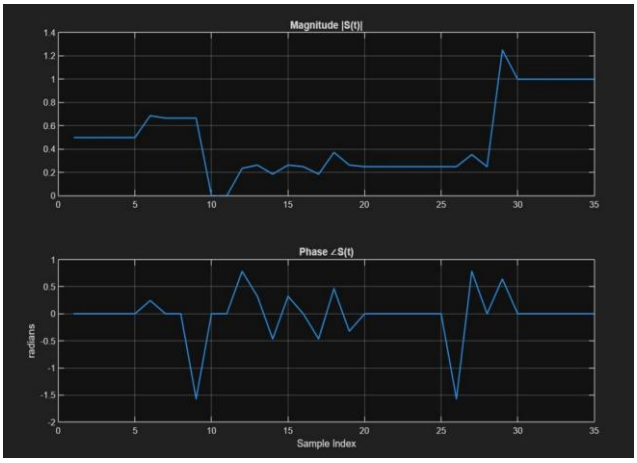


Fig. 8. Magnitude and Phase plots after QAM based processing

```
clc; clear; close all;
% Example sensor data: trash height
x = [6 6 6 6 6 8 8 8 0 0 0 2 3 2 3 3 2 4 3 3 3 3 3 3 3 0 3 3 12 12 12 12
    12 12 12];
% Normalize
x_norm = (x - min(x)) / (max(x) - min(x));
% Derivative (rate of change)
dx = [0 diff(x_norm)];
% Complex representation
S = x_norm + 1j * dx;
% Visualization
figure('Name','QAM-Inspired Representation','NumberTitle','off');
subplot(3,1,1); plot(x_norm,'b','LineWidth',1.5); grid on;
title('Normalized Sensor Data'); ylabel('x_{norm}(t)');
subplot(3,1,2); plot(dx,'r','LineWidth',1.5); grid on;
title('Derivative (Rate of Change)'); ylabel('dx/dt');
subplot(3,1,3); plot(real(S),imag(S),'-o','LineWidth',1.5); grid on;
title('QAM-Inspired Complex Trajectory');
xlabel('Normalized Value (I)'); ylabel('Rate of Change (Q)');
axis equal;
% Magnitude and phase
figure('Name','Magnitude and Phase','NumberTitle','off');
subplot(2,1,1); plot(abs(S),'LineWidth',1.5);
title('Magnitude |S(t)|'); grid on;
subplot(2,1,2); plot(angle(S),'LineWidth',1.5);
title('Phase ∠S(t)'); ylabel('radians'); xlabel('Sample Index'); grid on;
% Quantitative metrics
trajectory_length = sum(abs(diff(S)));
avg_phase = mean(angle(S));
stability_index = std(angle(S));
activity_index = mean(abs(dx));
event_count = sum(abs(dx) > 0.1);

fprintf('\n--- Dynamic Metrics from QAM-Inspired Representation ---\n');
fprintf('Trajectory Length : %.4f\n', trajectory_length);
fprintf('Average Phase Direction : %.4f radians\n', avg_phase);
fprintf('Stability Index (σθ) : %.4f\n', stability_index);
fprintf('Activity Index (|dx|avg) : %.4f\n', activity_index);
fprintf('Event Count (>|0.1|) : %d\n', event_count);
```

Fig. 9. Matlab Implementation

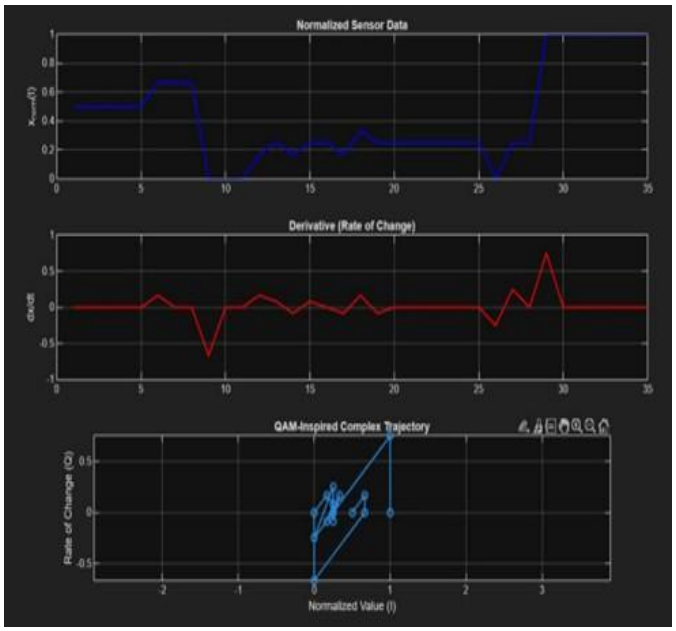


Fig. 10. Final Complex trajectory of the data from the Smart Waste Bin

VI. CONCLUSION

A QAM-inspired analytical framework for scalar data extraction was successfully implemented on a Smart Waste Bin prototype developed for the same. The proposed framework draws direct inspiration from Quadrature Amplitude Modulation (QAM), where amplitude and phase jointly represent information in a complex plane. By mapping sensor magnitude and rate of change to analogous in-phase (I) and quadrature (Q) components, the system achieves a modulation-like encoding of environmental variations.

This analogy enables intuitive visualization of sensor dynamics as trajectories in the complex plane, similar to QAM constellations. Such an approach bridges communication theory and IoT analytics, offering a novel paradigm for interpreting scalar data through a modulation-inspired lens.

REFERENCES

[1] B. Sklar, "Digital Communications: Fundamentals and Applications," *Prentice Hall*, 2001.
[2] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," *Cisco*, 2011.
[3] S. Mahajan *et al.*, "IoT-based Smart Waste Management System," *Int. J. Adv. Res. Comput. Sci.*, 2017.
[4] A. Gupta and P. Singh, "Cloud-connected Smart Dustbin," *IEEE ICI- CICT*, 2018.
[5] M. Patil, "Design of Intelligent Dustbin System," *IJETT*, 2019.
[6] M. Rahman *et al.*, "GSM-based Smart Waste Alert System," *IEEE Access*, 2020.
[7] L. Atzori *et al.*, "The Internet of Things: A Survey," *Comput. Netw.*, 2010.
[8] A. Zanella *et al.*, "IoT for Smart Cities," *IEEE IoT Journal*, 2014.
[9] R. Kumar *et al.*, "Signal Processing in IoT Devices," *Sensors*, 2020.
[10] J. Li *et al.*, "Feature Transformation for IoT Temporal Data," *IEEE Access*, 2021.
[11] P. Singh, "Derivative-based Feature Modeling for IoT," *IJETT*, 2021.
[12] A. Akram, "Real-time Sensing and Control in IoT," *IEEE Trans. Ind. Electron.*, 2022.
[13] X. Chen *et al.*, "Communication-inspired Sensing Framework," *IEEE Commun. Mag.*, 2018.
[14] M. Arif, "Hybrid Modulation for Sensor Analytics," *IEEE ICCE*, 2019.
[15] H. Sun, "Machine Learning in IoT: A Review," *IEEE Access*, 2020.
[16] K. Liu *et al.*, "Dynamic Feature Encoding in IoT Systems," *Sensors*, 2021.
[17] A. Sharma, "ESP32-Based Smart Bin with GSM," *IEEE ICCCA*, 2022.
[18] R. Jain, "IoT-enabled Waste Management," *IEEE Access*, 2023.
[19] A. Gogoi, "Visual Analytics for Sensor Signals," *IET Signal Processing*, 2022.
[20] C. Liu, "Complex Domain Signal Representation," *IEEE Trans. Signal Process.*, 2023.
[21] C. Trabelsi *et al.*, "Deep Complex Networks," *ICLR*, 2018.
[22] J. Bassey, "Complex-Valued Neural Representations," *IEEE Access*, 2021.
[23] S. Nanda, "IoT Data Analytics Using Hybrid Models," *IEEE IoT J.*, 2022.
[24] A. Alvi, "Complex Data Fusion in IoT," *IEEE Trans. Big Data*, 2023.

AUTHORS -

Author 1:
R Janaki

Author 2:
G R Shivaranjani

Author 3:
V Shreya