Cloud-Connected Human-Drone Interface for Intuitive Navigation

Atharva Pandhare, Ravi Raghavan, Shreyas Ramachandran Aryan Patil, Vijay Chandhar, Maria Striki, and Sasan Haghani
Department of Electrical and Computer Engineering
Rutgers University, Piscataway, NJ, 08854, USA
{asp265, rr1113, sr1547, aap286, vm527, ms1404, sh867}@rutgers.edu

***Abstract*—As drone technologies become more abundant, efficient and robust, path planning has become increasingly necessary. For ease of use, Human-Machine Interaction (HMI) applications have been developed to simplify the user experience and ensure efficient navigation. These applications enhance the drone's ability to perform various tasks, such as surveillance, delivery, and photography, with reliability and precision. This paper presents an innovative Human-Drone-Interface (HDI) designed to simplify the navigation of drones along user-defined paths. Drone flight paths can be defined by drawing routes or specifying Points of Interest (POIs) on a digital map, streamlining the user experience for users. This work addresses the hardware and software challenges of interpreting user-defined paths in real time, optimizing the drone's route, and generating precise flight commands. Algorithmic computations are delegated to the cloud to enhance battery efficiency. The system incorporates the Long Range (LoRa) protocol to communicate with the drone, which enables a more extended range of operations while providing reliable connectivity in areas with poor coverage and limited broadband access. By making drones easier to operate, the proposed solution seeks to make drones more widely accessible, thus pushing the boundaries of user-friendly drone navigation. Testing results demonstrated the HDI's robust ability to consistently instruct drones along user-defined paths. This capability makes it well-suited for real-world applications such as agriculture, where precise navigation is essential for irrigation or delivery systems, where path optimization ensures timely delivery of packages.**

***Index Terms*—Internet of Things (IoT), Human-Machine Interaction, Path Planning, Long Range (LoRa), Human-Drone Interface (HDI), Drone Navigation**

I. Introduction

Drones and advanced path-planning techniques have revolutionized numerous industries, bringing higher usability and precision to a broader range of applications. For example, drones enable precise pesticide application in agriculture, enhancing yield and sustainability [1]. However, current control solutions have significant tradeoffs between accessibility and user-friendliness. Traditional Human-Machine Interaction (HMI) drone applications are based on remote controls or complex programming interfaces. Drones conventionally communicate over 2.4 GHz Wi-Fi; in its place, various Internet of Things (IoT) technologies have become increasingly prevalent in drone systems. Specifically, RF technology innovations change how systems are controlled and monitored [2], [3].

This work was supported by a grant from Novo Nordisk, Inc.

Recently, as drone technology has become more popular, HMIs have become increasingly prevalent. For Instance, DJI Flighthub provides a Graphical User Interface (GUI) for managing and operating drones, catering to surveillance and detailed 3D mapping tasks. Flighthub [4] provides a suite of features, including real-time monitoring and mission planning. The platform includes a technical user interface, allowing users to "master their mission environment" [4] with features like point-based route planning and terrain mapping. While Flighthub [4] offers a comprehensive and robust GUI, the platform's highly technical nature and proprietary compatibility list may pose an entry barrier for novice users who may lack the technical expertise. Experienced users of FLighthub [4] would need technical knowledge, including an understanding of drone telemetry, and advanced command of systems analysis, as they would need to fine-tune a large number of parameters that are heavily interconnected.

To simplify the user experience of drone navigation and path planning, this paper proposes a cloud-connected (HDI) that enables users to define a drone path using a mobile application. The proposed solution is novel because it primarily offers a higher degree of control and unique path optimization capabilities expanding the reach of drone technologies to wider audiences, as outlined below.

First, this HDI has a unique feature that enables users to chart entire flight paths by drawing the path on a digital map. This gives the user far more granular control over the features of the flight path than what current implementations offer [4]–[7]. Existing HDIs, such as the one presented [5], enable the user to control the drone by specifying the flight destination and adjusting basic control parameters such as altitude and speed. Such existing solutions significantly limit the user's ability to control the drone's route. In contrast, the proposed HDI allows users to draw nuanced paths easily. Specifically, it provides for creating intricate flight paths with detailed curvature, enabling drones to smoothly follow the contours of various terrains and adapt to dynamic environmental conditions. This level of control enables users to better navigate around familiar terrain, ultimately leading to improved performance in real-world deployments.

Second, the HDI proposed in this paper also enables users to define a drone's flight path by providing a list of POIs. Unlike the HDI presented in [4], [6], [7], the order in which the POIs are marked does not determine the visitation order. In this paper's implementation, the user-specified POIs are used for computation, determining the optimal route using a shortest path algorithm. Some heuristic algorithms for optimal path planning have already been developed, such as the one presented in [8], which proposes a path-planning and obstacle avoidance algorithm based on a rapidly exploring random tree methodology. However, such algorithms [5] only determine the path between the start point and destination and do not enable users to select POIs along the drone's flight path. The HDI presented in this paper allows the users to specify POIs along a drone's flight path, ensuring optimized routes for scenarios where the visitation order is not the most crucial, like in delivery applications.

Third, the HDI uses the LoRa protocol [9] to relay instructions to the drone. While drone communication over Wi-Fi provides high bandwidth, it has a significant limitation: its operation radius is only 1 Km [9]. The LoRa protocol is a low-power, long-range communication protocol popular in IoT applications [10], achieving up to 15 km depending on the environment [11]. LoRa can be used in place of Wi-Fi as shown in [12], where LoRa, through a LoRaWAN gateway, allows the drone and a ground controller to establish a connection. When using a LoRaWAN gateway, any network traffic on the gateway limits the already low available bandwidth. However, this paper's proposed solution enables direct communication between LoRa transceivers, eliminating the need for gateways and streamlining the HDI implementation. This allows the HDI to be set up even in remote areas without the need for additional gateways or extenders.

Finally, in a HDI, leveraging cloud services for computations offers significant advantages. By offloading processing tasks to the cloud, the system can harness the cloud's extensive computational power and storage capacity, far exceeding the limitations of onboard drone hardware. Cloud-based computation ensures that the HDI can scale without the need for constant hardware upgrades. Additionally, reducing the amount of on-device computations will improve the battery efficiency of the drone.

Overall, this paper presents a novel HDI that aims to broaden the accessibility of long-range drone technologies by simplifying the user experience. Additionally, it extends the range of drone communication through the use of the LoRa protocol. Furthermore, the HDI presented in this paper includes advanced path optimization features, minimizing the overall distance covered by the drone's flight path.

The rest of the paper is organized as follows. Section II presents an overview of the LoRa-enabled HDI design. Section III presents a technical description of the underlying hardware and software implementation. Section IV provides an outline of the field testing procedure. Finally, Section V discusses potential future improvements and summarizes this paper's contributions.

II. System Overview

To improve HMIs and make them more accessible, the HDI uses the workflow shown in Fig 1. The user can either draw the drone's path or specify POIs on the mobile application, allowing the HDI software to compute the necessary drone instructions. A robust 2-tier system is used for data storage. After drawing the path, there are two options: save in Redis, an in-memory key-value database [13], or MongoDB, a NoSQL relational database [14]. If the path usage is immediate, it is stored in the Redis database; otherwise, it is stored in MongoDB for future use. Once a path is ready to be executed, it is uploaded to Redis, computed into drone instructions, and pulled by the router. Using LoRa, the router sequentially forwards the instructions to the drone for execution.

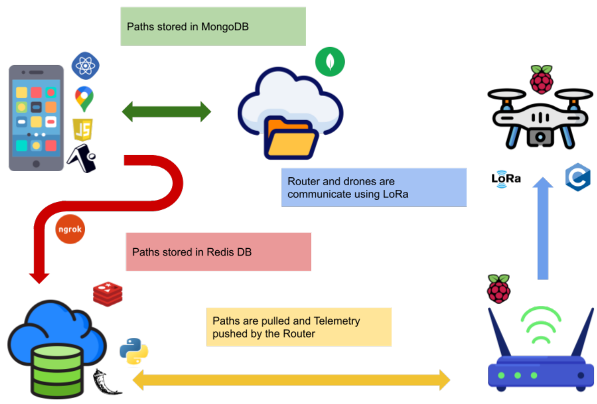

Fig. 1. Cloud Connected HDI System Workflow.

III. Hardware and Software Implementation

A. Hardware Design

The HDI provides a robust control system that leverages LoRa's features and reliability. It is comprised of two hardware components: a router and a drone. The router is based on a Raspberry Pi [15] with a GPS module and a HopeRF RFM95W LoRa transceiver [16]. The router's primary role is to control any communications between the software component and the drone. By accessing the Redis database, the router pulls drone instructions data and stores any telemetry information. It communicates with the drone using LoRa, a "spread-spectrum modulation technique derived from Chirp Spread Spectrum (CSS) technology" [11]. It sends out "chirps" or packets of minimal size, spreading it over a much larger spectrum band, giving it a more extended range and significant signal-to-noise ratio than the more common Frequency-shift Keying (FSK) modulation. The HDI leverages the chirps by sending data in batches; once the router is ready with the path's instruction set, each instruction is sent sequentially as a packet.

A custom Flight Controller (FC) based on a Raspberry Pi Pico [17], a Raspberry Pi and a HopeRF LoRa Transceiver are mounted on the drone. The FC uses Pulse-Width-Modulation (PWM) to control the motors and is programmed with specific instructions such as turning, going straight, etc.

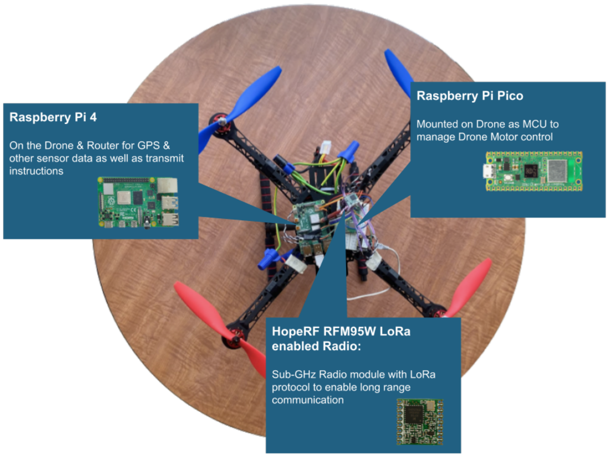

Fig. 2. Drone Prototype and Hardware Overview.

Fig. 3. Mobile Device UI for Drawing and Path Optimization.

The Raspberry Pi, using the LoRa transceiver, receives instructions from the router and, using universal asynchronous receiver/transmitter (UART), instructs the FC accordingly.

B. Software Design

The application has three major components: a React Native frontend, a Flask backend, and a 2-part data storage system. To streamline the storage and transmission of telemetry and drone flight path data, the system uses a 2-part data storage system composed of Redis and MongoDB. Since Redis is an in-memory database, stored data is temporary and will be lost upon system restart. The HDI uses Redis as an intermediary between the router and the user to cache drone flight paths and telemetry. By caching the path information before transmission, the overall system is protected against data loss. If the transmission to the drone is interrupted, the path can easily be re-transmitted. Additionally, the user can use stored telemetry data to gain insight into the performance and behavior of the drone during flight. On the other hand, MongoDB is a persistent database that enables the user to load and store paths for long-term usage.

The Flask backend controls the Redis and MongoDB databases by implementing HTTP endpoints. These endpoints facilitate data retrieval, storage, and updates, allowing seamless communication with Redis for swift, in-memory

Fig. 4. UI for Loading and Saving Path.

operations and MongoDB for robust, document-based storage. The Flask server can be hosted either on a local network or on a cloud platform such as Amazon Web Services (AWS) [18], Microsoft Azure [19], or Google Cloud Platform (GCP) [20]. Additionally, since all services and applications within the HDI can be self-hosted on a local network, the HDI can function in areas without internet access.

As shown in Fig. 3, the React Native frontend presents the user with an intuitive digital map interface to define features of a drone's flight path. Users can either draw the entire path or specify POIs. As denoted by the circle, users can only work within the router's range to ensure reliable communication between the router and the drone. To draw an entire flight path, the user needs to press the "Draw" button and begin drawing on the digital map. To discard the drawn path, users can simply press the "Delete" button. While the user is drawing the path internally, the path is represented as a list of Global Positioning System (GPS) coordinates. To ensure smoothness while preserving sufficient details about flight curvature for execution, the drone's flight path is discretized at a lower resolution. Once the user has finished drawing the flight path, they have two options. By clicking "Save", as illustrated in Fig. 4, the GPS coordinates of the drone's path are sent to the backend and then saved in MongoDB, giving users the option to reuse this drawn path. The name of the path, which acts as the path's ID, can only have a maximum of 20 characters. To access saved paths, users can click the "Load" button and then click the desired path to render it onto the digital map. Paths can be sorted by name: swiping left or right sorts them in ascending or descending order respectively. Users can delete paths by holding the path block and selecting the "Delete" option. On the other hand, by clicking "Submit," the GPS coordinates of the drone's path are uploaded to Redis via the backend for execution.

In general, enabling users to draw flight paths on digital maps presents a transformative tool for drone navigation. This can be useful for agriculture particularly in optimizing irrigation practices. Tailored irrigation strategies can be implemented with precision by allowing users to specify flight paths over designated areas within the agricultural grid ultimately contributing to sustainable agricultural practices.

Alternatively, on the digital map users can define flight

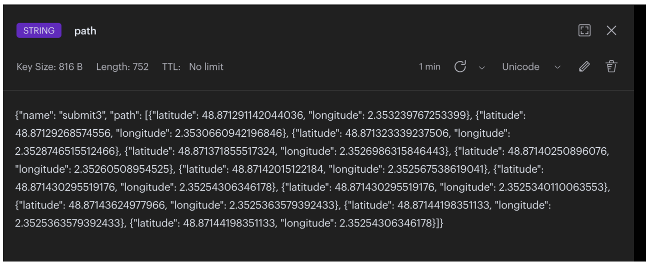

Fig. 5. Redis GUI showing deployment of the path for execution.

path features by specifying the starting point of the drone as well as POIs along the drone's flight path. To indicate the starting point, the user needs to click the "Origin" button and place the corresponding marker on the digital map. Likewise, to indicate POIs, the user needs to click the "Stops" button and place the corresponding markers on the digital map. After specifying this information, the user has two options: "Save" and "Submit" which have the same functionality as described above. By clicking "Save", the GPS Coordinates of the starting point and POIs are sent to the backend and then saved in MongoDB, giving users the option to reuse the flight path features. By clicking "Submit", the mobile application will transmit the GPS coordinates of the starting point and POIs to the backend application. Then, the backend application will use the traveling salesman algorithm, as provided by the NetworkX Python package, to determine the shortest route from the starting point that visits all POIs [21]. Once the shortest route has been determined, a series of GPS coordinates outlining the sequence in which the POIs will be visited will be transmitted to Redis through the backend for execution.

Overall, path optimization is crucial as it enhances efficiency by minimizing flight time and energy consumption, ensuring longer operational periods, which would be extremely valuable for delivery applications in ensuring efficient and timely deliveries.

IV. Testing and Results

The testing phase demonstrates the functioning of the HDI and visualizes the communication flow between the UI and the drone. To test the HDI, the data throughput is monitored using SSH terminals on the hardware and the respective applications for the backend and database services. To set up the HDI for testing the drone, both the React Native and Flask applications were hosted locally.

The testing procedure started with a custom-drawn path submitted in the GUI. The application converts the drawn path into GPS coordinates. The coordinates are then saved to the Redis database as seen in Fig 5. On the Router terminal in Fig. 6, a key-value pair with key "commands" is visible, representing the drone's flight instructions that are generated by the backend as a result of pulling the path. The value section of the key-value pair represents a list containing drone instructions that are needed to follow the flight path. Each number in the list is an encoded instruction corresponding to a drone action. As seen in Fig. 6, before

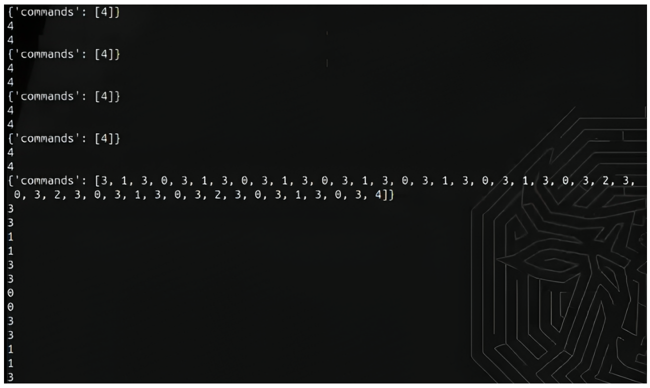

Fig. 6. Router Pulling Path and Transmitting to Drone.

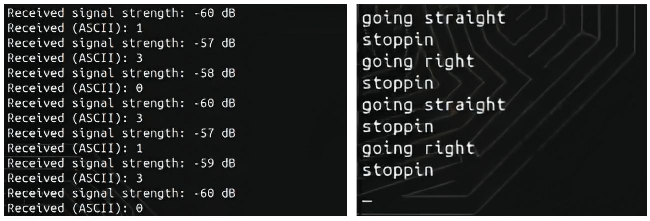

Fig. 7. Drone Receiving Instructions & Signal Strength (left). Readouts from Flight Controller (right).

a path is pulled, the Redis holds simply the command '4" corresponding to the stop instruction; the same is true after the path is pulled as it clears the data. In Fig. 6, the subsequent lines after the path show the accessing and sending of the instructions sequentially to the drone.

The left terminal in Fig. 7 shows the drone receiving the instruction and the connection's signal strength. Finally, the terminal on the right of Fig. 7 represents the feedback from the FC, representing the drone responding appropriately to received instructions.

The testing procedure was repeated multiple times for both use cases with various levels of complexity. In every test, the HDI proved to be robust by consistently generating and transmitting the path.

V. Future works and Conclusion

Potential avenues to further develop this proposed HDI includes adaptive routing capabilities, computer vision techniques, and enhanced long-range communication. Upgrading the Pico to a more established flight controller can allow for further experimentation and deployment of this HDI. It could give the HDI more control and capabilities in different conditions and setups, proving useful in a broader range of applications. Another enhancement is regarding the LoRa. Altering the spreading factor of the LoRa transceiver can allow for different bit rates, as seen in table I [11]. Additionally, dynamically adjusting this can allow the HDI to minimize latency based on range and packet size, as shown in [22]. Such improvements to the HDI can provide the user with higher degrees of freedom for drone flight paths and also allow for reliable execution, with support for low-latency path corrections.

TABLE I
IMPACT OF LORA SPREADING FACTORS ON BITRATE & RANGE [11].

Spreading Factor	Bit Rate	Range
SF10	980 bps	8km
SF9	1760 bps	6km
SF8	3125 bps	4km
SF7	5470 bps	2km

Moreover, using Redis allows for higher scalability when deploying drone fleets. Redis can accommodate multiple paths for different drones by implementing a drone ID. This would enable the router to distinguish which drone receives the instructions. This, paired with LoRa's immunity to noise and even other spreading factors as described in [11], can allow for superior flight planning for fleets.

In summary, this paper presents an intuitive and robust HDI that enables the user to control a drone's flight path through a user-friendly mobile interface. The system features advanced path optimization capabilities, ensuring an optimal flight path that minimizes distance. The HDI's usage of the LoRa protocol for communication provides robust data transmission and extends the drone's flight range.

REFERENCES

[1] A. Amarasinghe, L. Jayaratne, and V. B. Wijesuriya, "A path planning drone solution to safe pesticide usage in arable lands," in *2022 2nd International Conference on Advanced Research in Computing (ICARC)*, 2022, pp. 55–60.

[2] K. Intwala, S. Jatav, and K. Kolhe, "System to capture WiFi based Drones using IoT," in *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, 2022, pp. 1–6.

[3] V. Herdel, L. J. Yamin, and J. R. Cauchard, "Above and beyond: A scoping review of domains and applications for human-drone interaction," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, 2022.

[4] "DJI FlightHub 2 - Drone Software and Management Platform - DJI Enterprise." [Online]. Available: https://enterprise.dji.com/photo

[5] C.-I. Lin, Y.-Y. Chang, and K.-T. Lai, "A Web-based UAV Control System for Intuitive Human-Drone Interaction," in *2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, 2023, pp. 135–136.

[6] AirHub, "AirHub - Drone Operations Software - AirHub," 5 2024. [Online]. Available: https://airhub.app/

[7] "DroneDeploy Aerial: drone mapping software — Drone Surveying." [Online]. Available: https://dronedeploy.com/product/dronedeploy-aerial

[8] A. Chatterjee and H. Reza, "Path Planning Algorithm to Enable Low Altitude Delivery Drones at the City Scale," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 750–753.

[9] S. Devalal and A. Karthikeyan, "LoRa Technology - An Overview," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 284–290.

[10] G. P. Reddy and Y. V. Pavan Kumar, "Demystifying LoRa Wireless Technology for IoT Applications: Concept to Experiment," in *2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Dec. 2021, pp. 01–06. [Online]. Available: https://ieeexplore.ieee.org/document/9668500

[11] *LoRa and LoRaWAN: A technical overview*, Dec. 2019. [Online]. Available: https://loradevelopers.semtech.com/

[12] A. Rahmadhani, Richard, R. Isswandhana, A. Giovani, and R. A. Syah, "LoRaWAN as Secondary Telemetry Communication System for Drone Delivery," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 116–122.

[13] "Redis - The Real-time Data Platform." [Online]. Available: https://redis.io/

[14] "MongoDB: The Developer Data Platform." [Online]. Available: https://www.mongodb.com

[15] "Raspberry Pi 4 Model B," Mar. 2024. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[16] "RFM95W - HOPERF - Reliable original manufacturer of IoT key components." [Online]. Available: https://www.hoperf.com/modules/lora/RFM95W.html

[17] "Raspberry Pi Pico W product brief." [Online]. Available: https://datasheets.raspberrypi.com/picow/pico-w-product-brief.pdf

[18] "Cloud Computing Services - Amazon Web Services (AWS)." [Online]. Available: https://aws.amazon.com/

[19] "Cloud Computing Services | Microsoft Azure." [Online]. Available: https://azure.microsoft.com/en-us

[20] "Cloud Computing Services." [Online]. Available: https://cloud.google.com/

[21] A. Hagberg, P. J. Swart, and D. A. Schult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[22] Z. B. Arslanbenzer, B. Bayram, T. T. Sarı, and G. Seçinti, "Utilizing LoRa for Control Link in Software-defined Aerial Networks: Analysis and Implementation," in *2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2023, pp. 164–169.