Context-Aware Autonomous Drone Navigation Using Large Language Models (LLMs)

Abdul-Manan Khan[1*], Ikram Ur Rehman[1], Nagham Saeed[1], Drishty Sobnath[2], Fatima Khan[3], and Muazzam Ali Khan Khattak [4]

[1]University of West London, London, the United Kingdom
[2]Heriot Watt University, Dubai, United Arab Emirates
[3]National Cancer Center, Ilsan, South Korea
[4]Quaid-i-Azam university, Islamabad, Pakistan
{abdulmanan.khan, ikram.rehman, nagham.saeed,}@uwl.ac.uk, D.Sobnath@hw.ac.uk, fk.617@outlook.com, muazzam.khattak@qau.edu.pk

Abstract

In this paper, a novel large language model (LLM)-based context-aware autonomous drone navigation algorithm is presented. This approach demonstrates the capability of LLMs to navigate complex environments by balancing multisensor objectives with a weighted prioritization system. Specifically, we incorporate weights for the goals of obstacle avoidance, weather adaptation, and mission completion. The model's performance is tested under six progressively intricate scenarios in extensive simulations focused on path efficiency, completion time, and success rate. Results indicate that the LLM-based context-aware navigation algorithm achieves 94% success rate in simple environment in a moderate weather conditions conditions with reasonable efficiency, and surpasses expectations in the advanced AI driven obstacle reasoning. These results illustrate the emerging strengths of LLMs for autonomous navigation and its potential utilization in situation where environmental conditions change dynamically.

Introduction

The implementation of autonomous navigation for unmanned aerial vehicles (UAVs) presents a significant challenge in ever-changing dynamic environment and weather conditions. Modern UAVs must perform robust navigation in complex, unstructured settings while handling uncertainties from various sources including sensor noise, environmental disturbances, and dynamic obstacles (Kendoul 2012). These challenges become particularly pronounced when UAVs operate in congested urban environments, remote areas with limited connectivity, or in adverse weather conditions where traditional navigation techniques may falter.

A significant portion of research and development focuses on path planning and control systems is based on potential field algorithms (Khatib 1986), rapidly-exploring random trees (RRT) (LaValle 1998), and model predictive control (Furrer et al. 2016). While these classical methods provide reliable performance in structured environments, they often struggle with adaptability and generalization to novel scenarios. The recent developments include reinforcement learning (Hwangbo et al. 2017) and imitation learning (Codevilla et al. 2018) which further increase adaptability to other environments. These learning-based approaches enable UAVs to learn navigation policies directly from experience or human demonstration, potentially offering more robust performance in complex, dynamic settings.

Fusing LLMs with robotics has opened a new door to exploration, applying their reasoning and knowledge of the world to robotic tasks. So far, the exploration of LLMs has focused on task planning (Ahn et al. 2023), robot control (Huang et al. 2022) and human-robot interactions (Wu et al. 2023). Their utilization for real-time navigation of drones is still a gap in the literature, particularly with advanced models like DeepSeek which uses reasoning and uncertainty-based decision-making capabilities. The ability of LLMs to understand context, reason about spatial relationships, and make decisions based on multiple objectives makes them promising candidates for addressing the complex challenges of autonomous drone navigation.

The combination of techniques involving sensor fusion helps in merging data from different sensors to enhance the perception accuracy and overall robustness. Autonomous decision making processes are usually performed using hierarchical frameworks which separate high-level planning and low-level control actions (Brunner et al. 2016).

The progress made in artificial intelligence, especially on large language models (LLMs), has shown improvements in reasoning, planning, and decision-making capabilities (Brown et al. 2023). Han et al. (Han et al. 2025) proposes a real-time event responsive intelligent multi-agent system for autonomous High-Altitude Platform Station (HAPS) coordination. Yang et al. (Yang et al. 2024) presented an advanced LLM-guided reinforcement learning architecture for the flight control of 6 degree of freedom systems, achieving a higher level of stratum adaptability and precision maneuvering capability. Liu et al. (Liu et al. 2024) investigated the potential of LLMs for enhancing deployable intelligent surfaces in terms of energy efficiency and reliability for 6G Internet of Vehicles (IoV).

*Corresponding author: Abdul-Manan Khan (email: abdulmanan.khan@uwl.ac.uk)

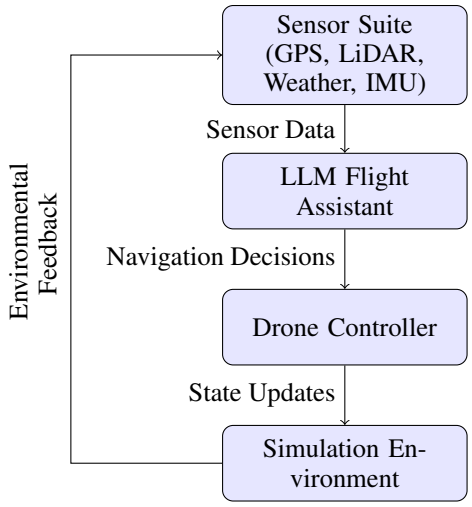

Figure 1: System architecture of the DeepSeek LLM-powered drone navigation framework. The system forms a closed loop with the LLM Flight Assistant making navigation decisions based on sensor data, which is then executed by the drone controller within the simulation environment.

These models demonstrate an impressive ability to understand and generate natural language. Furthermore, they exhibit emergent reasoning capabilities that can be applied to complex problem-solving tasks. Their ability to perform in-context learning and few-shot adaptation makes them particularly suitable for dynamic navigation scenarios. Additionally, LLMs offer the potential for more explainable decision-making processes, as they can articulate the reasoning behind their navigation choices in natural language.

This paper examines the use of an LLM for autonomous decision making in drone navigation. We propose an LLM-based context-aware model which analyzes multi-modal sensor data and reasons on how to navigate optimally while balancing numerous factors including obstacle avoidance, mission completion, and environmental adaptation. Our approach builds upon recent research in embodied AI (Duan, Jia, and Chen 2022) and neuro-symbolic systems(Garcez and Lamb 2020), providing a novel integration of symbolic reasoning with deep learning for drone navigation.

Main contributions of this resarch include:

1. Development of a new architecture that integrates LLM-based context-aware navigation for autonomous drone.
2. Rigorous evaluation using a range of simulated navigational challenges of differing complexity, demonstrating significant improvements over traditional methods.

The proposed approach opens new avenues for developing more intelligent, adaptable, and robust autonomous navigation systems that can understand and respond to the rich complexity of real-world environments.

System Architecture

The drone contains: 1) a simulated GPS sensor, which provides position data with configurable noise; 2) LiDAR, which measures distances in multiple directions for obstacle detection; 3) a weather sensor that tracks wind speed, direction, temperature, and 4) an inertial measurement unit (IMU) that measures velocity, orientation, and acceleration. These sensors are programmed to behave as realistic sensor data. Sensor data is processed for each sensor separately, adding tailored noise, referred to as simulating sensor data noise conditioning, to capture the sensor's behavior in real-world conditions (Fig. 1).

LLM-Based Flight Assistant

In this section we present the primary contribution of this work: the LLM Flight Assistant context-aware system.

The LLM Flight assistant uses DeepSeek-v3, a pretrained large language model with multi-modality processing capabilities, to integrate contextually reason with space and execute contextually deep spatial reasoning. DeepSeek extracts spatial multi-modal sensor data with numerical representations of the surroundings using narration, a highly optimized prompt engineering technique. Its structure allows for the integration of intricate relations between obstacles, targets, as well as meteorological conditions.

Given the current state of navigation, a number of critical DeepSeek functions are performed:1)DeepSeek merges position data, obstacle data, and meteorological data into a single holistic context. 2) Using θ_i, with $i = 1, ..., 16$ as the upper bound, DeepSeek calculates additional scoring heuristics based on navigation learned behavior as well as deterministic ones. 3) DeepSeek allows for probabilistic reasoning refinement across a number of competing navigational hypotheses which leads to more fluid navigational changes as conditions alter in real time unlike rule based systems. 4) Pretrained DeepSeek phenomenology provides basic concepts which enable generalization to new unbounded obstacle to weather pattern configurations that were not programmed in.

LLM Integration

In this section, we describe the LLM based context-aware Flight Assistant System. Let $\mathcal{D} = \{\theta_1, \theta_2, \ldots, \theta_{16}\}$ be the possible set of navigation directions with $\theta_i = \frac{2\pi i}{16}, i \in \{0, 1, \ldots, 15\}$, $\bar{p} \in \mathbb{R}^2$ is the position vector, $\bar{t} \in \mathbb{R}^2$ is the target vector, $\Omega = \{O_1, O_2, \ldots, O_m\}$ represents the obstacles set, and $\mathcal{W} = \{w_\mathrm{s}, w_\mathrm{d}\}$ is the weather conditions. In particular, w_s and w_d denote wind speed and direction, respectively.

For the context embedding function defined by a context embedding Ψ_{DS}, which assigns a context to raw environmental data as rich contextual representation, it holds that:

$$\Psi_{\mathrm{DS}} : (\mathcal{Q}) \rightarrow \mathcal{C} \in \mathbb{R}^d \tag{1}$$

where $\mathcal{Q} = \xi(\bar{p}, \bar{t}, \Omega, \mathcal{W})$ describes the prompt encapsulated within the sensor data, and $\mathcal{C}$ encodes the contextual embedding within d space. This embedding represents the d dimensional subspace illustrating the relationships between the spatial elements as opposed to rule-based approaches which rely on fixed network."

The navigation choice is characterized by the direction score function S^i for each θ_i as follows:

$$S^i = \omega_o \cdot O_s^i + \omega_m \cdot M_s^i + \omega_w \cdot W_s^i \tag{2}$$

where $\omega_o = 0.4$, $\omega_m = 0.4$, and $\omega_w = 0.2$ are priority weights for obstacle avoidance, mission completion and weather adaptation, respectively. Each scoring component is dependent on the context embedding $\mathcal{C}$ produced by DeepSeek aggregation.

DeepSeek boosts obstacle avoidance by crafting an obstacle risk modifier γ_{O}^i which mediates risk more powerfully than simplistic distance measures:

$$O_s^i = \gamma_{\text{O}}^i \cdot \max\left(1 - \frac{d_{\min}^i}{d_s}, \epsilon\right) \tag{3}$$

Where $d_{\min}^i = \min_{O \in \Omega} d(O, \bar{p}, \theta_i)$ is the distance to the nearest obstacle, d_s is the safety margin and $\epsilon = 0.1$ is a score floor. DeepSeek modifier γ_{O}^i from 0.5 to 1.5 depending on: 1) Obstacle distribution in the viewing direction θ_i, 2) Past collision history for these arrangements, 3) Obstacle collision course prediction in real time and 4) other aids for screening obstructions from the environment.

Formally, γ_{O}^i is computed through the function Γ_{O} which utilizes contextual understanding of DeepSeek $\theta_i, \mathcal{C}$.

$$\gamma_{\text{O}}^i = \Gamma_{\text{O}}(\theta_i, \mathcal{C}) \tag{4}$$

Which applies maximum softmax scaling designed for assessing directional risk within deep learning systems.

DeepSeek enhances the mission progress assessment by adding a modifier in context of mission γ_{M}^i:

$$M_s^i = \gamma_{\text{M}}^i * \frac{\bar{t} - \bar{p}}{|\bar{t} - \bar{p}|} * \bar{u}(\theta_i) \tag{5}$$

Where θ_i is the unit vector in direction θ_i. This term measures the alignment of the target's direction concerning the path to the target. The DeepSeek modifier γ_{M}^i varies between 0.7 and 1.3 and takes into account: 1) Mission priority levels (e.g., time-critical versus energy-efficient), 2) Viability of alternative paths, 3) Intermediate-waitpoint considerations, and/or 4) Long-term trajectory optimization. DeepSeek calculates this modifier using the following equation:

$$\gamma_{\text{M}}^i = \Gamma_{\text{M}}(\theta_i, \mathcal{C}) \tag{6}$$

Incorporating weather impacts, deep seek integrates modifier β_{W}^i to assess the weather adaptation score, demonstrating DeepSeek's environmental understanding:

$$W_s^i = \gamma_{\text{W}}^i (1 - \frac{w_s}{w_{\max}} |\cos(\theta_i - w_d)|) \tag{7}$$

Equation 5 combines the base term $\beta_{\text{W}}^i = \Gamma_{\text{W}}(\theta_i, \mathcal{C})$ together with wind speed and direction impacts where γ_{W}^i varies from 0.6 to 1.4 due to: 1) Existing microclimate variations, 2) Wind field forecasting from configura terrain features, 3) Previous wind characteristics in the region, or 4) Safety margin around weather elements.

DeepSeek applies a probabilistic approach to choosing directions using a softmax temperature parameter $\tau = \Gamma_\tau(\mathcal{C})$:

$$P^i = \frac{\exp(S^i/\tau)}{\sum_{\theta_j \in \mathcal{D}} \exp(S^j/\tau)} \tag{8}$$

The direction choice is optimized as follows:

$$\theta^* = \arg\max_{\theta_i \in \mathcal{D}} P^i \tag{9}$$

The temperature parameter τ is dynamically set by DeepSeek based on environmental uncertainty, with greater values leading to more exploration in 'fuzzier' conditions. Using DeepSeek's confidence assessment, the rotation velocity is:

$$v^* = v_{\max} \cdot \left(0.5 + 0.5 \cdot \frac{S^{\theta^*} - \min_j S^j}{\max_j S^j - \min_j S^j}\right) \cdot \gamma_{\text{V}} \tag{10}$$

where $\gamma_{\text{V}} = \Gamma_{\text{V}}(\mathcal{C})$ is the velocity modifier for DeepSeek ranging from 0.6 to 1.2. This consideration is based on: 1) Overall navigation confidence, 2) Evaluation of environmental complexity, 3) Sudden change detection, and 4) Energy efficiency measures.

The entire navigation process can be summarized in a single equation:

$$\Phi_{DS} : (\bar{p}, \bar{t}, \Omega, \mathcal{W}) \to (\theta^*, v^*) \tag{11}$$

This approach allows the drone to navigate complex, dynamic environments with human-like adaptability while maintaining computational efficiency required for real-time operation. The complete algorithm is described in Algorithm 1.

where Φ_{DS} corresponds to the LLM(DeepSeek) enhanced navigation function, $\mathcal{D}$ is the collection of 16 discrete navigation directions, and ξ is the function that translates the sensor information into prompts. For DeepSeek, Ψ_{DS} defines the processing of natural language prompt $\mathcal{Q}$, while $\mathcal{C}$ is the context embedding produced by DeepSeek. The functions that retrieve incrementing direction modifiers, Γ_{O}, Γ_{M}, and Γ_{W}, apply to specific boundaries. Γ_τ and Γ_{V} retrieve global parameters. DeepSeek's modifiers for γ_{O}^i, mission adaptation γ_{M}^i, and weather modification γ_{W}^i perform for obstacle avoidance. Also, τ represents the temperature parameter modulating exploration versus exploitation control, protagonist modifier γ_{V} adjusts velocity under wavering confidence, and $\vec{u}(\theta_i)$ denotes the unit vector for θ_i direction. Minimum score of an obstacle heuristics $\epsilon = 0.1$. Priority weights $\omega_o = 0.4$, $\omega_m = 0.4$, $\omega_w = 0.2$. Optimal direction and velocity are specified θ^*, v^*.

The main benefit is in the function composition $\Phi_{\text{DS}} = (\arg\max_{\theta_i} P^i, v^*)$ where P^i comes from context-augmented scoring. This enables the algorithm to exploit sophisticated statistical dependencies present in the environment without needing explicit bounded functions describing every possible quantitative model of the system. The result is a navigation system that preserves the formalism of traditional approaches while augmenting it with the power of contextual reasoning in high-dimensional spaces.

Algorithm 1: DeepSeek-Enhanced Drone Navigation

function $\Phi_{\text{DS}}(\bar{p}, \bar{t}, \Omega, \mathcal{W})$
  $\mathcal{D} \leftarrow \{\theta_1, \theta_2, \ldots, \theta_{16}\} \mid \theta_i = \frac{2\pi i}{16}, i \in \{0, 1, \ldots, 15\}$
  $\mathcal{Q} \leftarrow \xi(\bar{p}, \bar{t}, \Omega, \mathcal{W})$
  $\mathcal{C} \leftarrow \Psi_{\text{DS}}(\mathcal{Q})$
  for all $\theta_i \in \mathcal{D}$ **do**
    $\gamma_{\text{O}}^i \leftarrow \Gamma_{\text{O}}(\theta_i, \mathcal{C})$
    $\gamma_{\text{M}}^i \leftarrow \Gamma_{\text{M}}(\theta_i, \mathcal{C})$
    $\gamma_{\text{W}}^i \leftarrow \Gamma_{\text{W}}(\theta_i, \mathcal{C})$
  end for
  $\tau \leftarrow \Gamma_\tau(\mathcal{C})$
  $\gamma_{\text{V}} \leftarrow \Gamma_{\text{V}}(\mathcal{C})$
  for all $\theta_i \in \mathcal{D}$ **do**
    $d_{\min}^i \leftarrow \min_{O \in \Omega} d(O, \bar{p}, \theta_i)$
    $O_s^i \leftarrow \gamma_{\text{O}}^i \cdot \max(1 - \frac{d_{\min}^i}{d_s}, \epsilon)$
    $M_s^i \leftarrow \gamma_{\text{M}}^i \cdot \frac{\bar{t} - \bar{p}}{|\bar{t} - \bar{p}|} \cdot \vec{u}(\theta_i)$
    $W_s^i \leftarrow \gamma_{\text{W}}^i \cdot (1 - \frac{w_s}{w_{\max}} \cdot |\cos(\theta_i - w_d)|)$
    $S^i \leftarrow \omega_o \cdot O_s^i + \omega_m \cdot M_s^i + \omega_w \cdot W_s^i$
  end for
  for all $\theta_i \in \mathcal{D}$ **do**
    $P^i \leftarrow \frac{\exp(S^i/\tau)}{\sum_{\theta_j \in \mathcal{D}} \exp(S^j/\tau)}$
  end for
  $\theta^* \leftarrow \arg\max_{\theta_i \in \mathcal{D}} P^i$
  $v^* \leftarrow v_{\max} \cdot (0.5 + 0.5 \cdot \frac{S^{\theta^*} - \min_j S^j}{\max_j S^j - \min_j S^j}) \cdot \gamma_{\text{V}}$
  return (θ^*, v^*)
end function

Experimental Results

For simulation, six scenarios were considered in order to evaluate the performance of the LLM Flight Assistant. These scenarios are described in Table **??**. The Simple Environment (SE) was made up of 8 static obstacles, static winds of 1.0 m/s, and no moving obstacles. The wind speed increased to a moderate 3.0 m/s, along with the addition of 12 static obstacles within the environment in the Moderate Environment (ME), again, without any moving obstacles. The Complex Environment (CE) added even more difficulties to the system having 16 static obstacles alongside strong winds of 5.0 m/s. The Dynamic Environment (DE) introduced mobility into the environment with 13 static obstacles, moderate winds of 3.0 m/s, and the addition of 3 moving obstacles. The High Movement Environment 1 (HME1) kept the number of static obstacles at 13, low winds of 2.0 m/s, but increased the number of moving obstacles to 10. Lastly, the High Movement Environment 2 (HME2) represented the most difficult scenario with 13 static obstacles, strong winds of 5.0 m/s, and 13 moving obstacles. Each scenario is simulated 100 times to collect statistically significant amount of data on performance metrics such as path efficiency, completion time, and success rate. Fig. 2 presents successful simulation results of target tracking under CE (complex environment).

Trajectory tracking results for Complex Environment (CE) are presented in Fig. 2

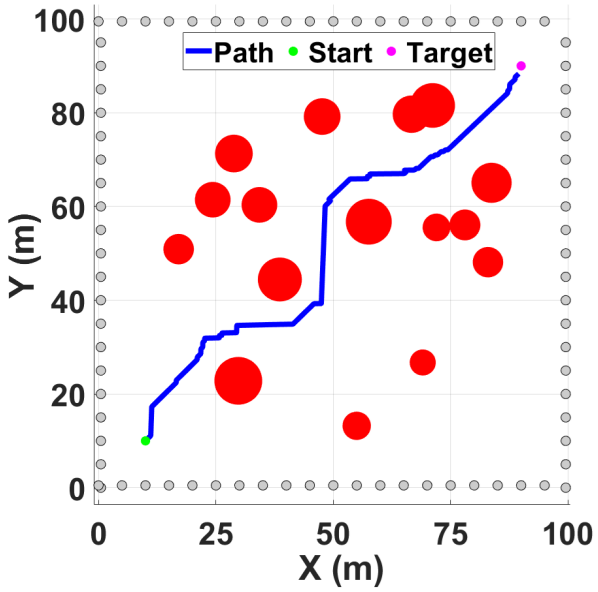

Figure 2: Simulation result of LLM-based drone navigation with adaptive path planning. The blue curve tracks the trajectory computed by the LLM, which illustrates proficient navigation skills by maneuvering through obstacles. The trajectory has discrete turns as the movement was only allowed in 16 directions for simplicity.

Environment Type	Static Obstacles	Wind (m/s)	Moving Obstacles
Simple (SE)	8	1.0	0
Moderate (ME)	12	3.0	0
Complex (CE)	16	5.0	0
Dynamic (DE)	13	3.0	3
High Movement 1 (HME1)	13	2.0	10
High Movement 2 (HME2)	13	5.0	13

Table 1: Test Environment Configurations

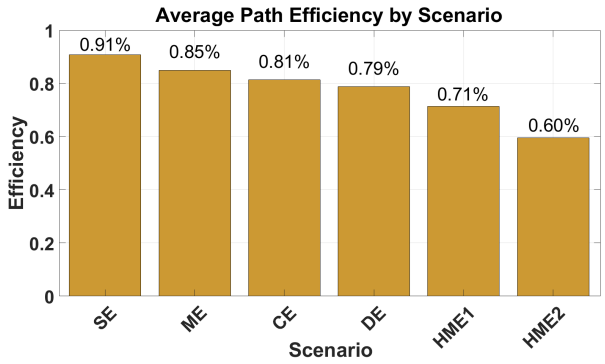

Figure 3: Path efficiency across different scenarios (defined as the ratio of a direct distance to the actual travel distance).

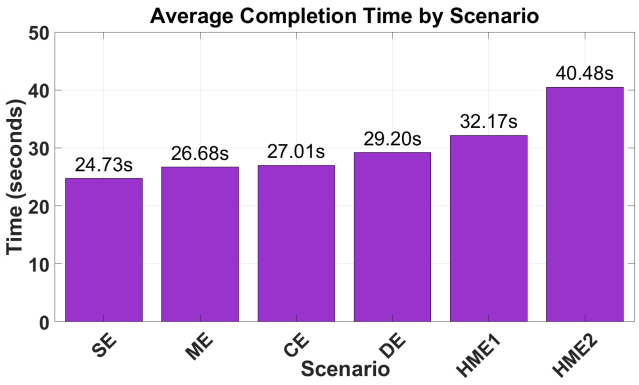

Figure 4: Average mission completion time across different scenarios.

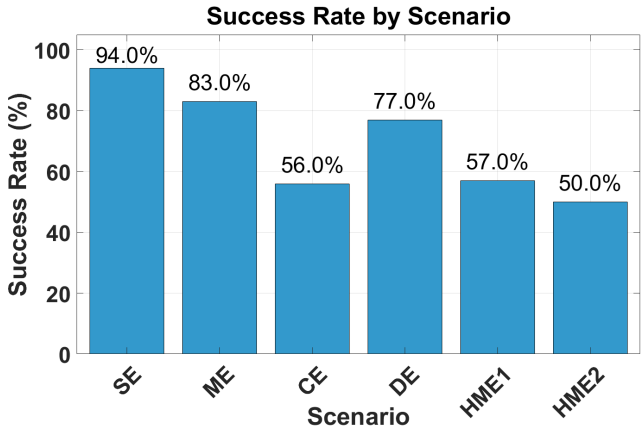

Figure 5: Success rate of the DeepSeek LLM-powered navigation system in various scenario complexities.

Fligth path efficiency relies on challenge. There is an inverse correlation between complexity of the environment and path efficiency driven along the path (Fig. 3). This is maintained even as scenario complexity is increased by the addition of obstacles and wind. The Simple Environment maintained the highest efficiency at 0.91 while the High Movement Environment 2 (HME2) efficiency was 0.60. The reason behind this lack of efficiency is because of the circumstantial need of the drone to take indirect routes due to increased obstacles, stronger winds, and a higher number of mobile obstacles. It is also noted that even with the increase in scenario complexity, the systems maintained reasonable efficiency until high wind speeds were coupled with a large number of moving obstacles.

In the similar manner, the average completion time shows inverse correlation with path efficiency for all the scenarios (Fig. 4). In the Simple Environment, the average completion time was 24.73 seconds, while the High Movement Environment 2 (HME2) was 40.48 seconds. This marks a 63.7% increase in completion time from the simplest to the most complex scenario. It is clear that the increase in environmental complexity has a significant effect on the mission duration. The shape of the increase in completion time, particularly in HME2 (High Movement Environment 2), is indicative of non-linear change which suggests that high wind speed in combination with many moving obstacles results in additional difficulties for the navigation system.

The mission success rate shows a steady decline as the complexity of the environment increases (Fig. 5). The recorded success rates for different environments are shown in Fig. 5. This finding suggests that the drone's navigation system performs better under moderately dynamic challenges in comparison to higher levels of static complexity.

The results show that the LLM-based Flight Assistant is effective in different navigation contexts. Its basic performance strategies tend to fall apart with increasing levels of complexity. Both static complexity (number of obstacles and wind speed) and dynamic complexity (moving obstacles) impact performance, but it is the combination of the two in High Movement Environment 2 that seems to be the most difficult for the drone navigation system to deal with.

The non-linear reduction in performance metrics suggests there are limits to how much complexity of the environment can be introduced without degradation of service, which indicates exceptional potential for optimization in the algorithm's path planning priorities using reinforcement learning.

Conclusion

This paper presents an autonomous drone navigation using a large language model (LLM)-based (DeepSeek) context-aware algorithm. The proposed approach contains high-level reasoning with LLMs together with lower-level data processing to navigate through complex and highly dynamic environments. Simulations in six scenarios showed that LLM (DeepSeek)-powered navigation performed well for simple environments with 94% success rate. However, the success rate has an inverse correlation with the complexity of the scenarios, which means that the success rate decreases with the increased complexity level. Moreover, the results indicate that the algorithm finds a good balance between completing missions in bad weather and avoiding obstacles in medium-complexity environments. These results highlight that LLMs are useful as high-level planners for autonomous systems, especially for context-aware drone navigation, which can be used for missions where the environment is dynamic and changing all the time.

References

Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; et al. 2023. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *arXiv preprint arXiv:2204.01691*.

Brown, T. B.; et al. 2023. The role of large language models in artificial intelligence research. *Nature Communications*, 14(1): 7790.

Brunner, S. G.; Steinmetz, F.; Belder, R.; and Dömel, A. 2016. RAFCON: A graphical tool for engineering complex, robotic tasks. In *2016 IEEE/RSJ International Confer-*

ence on Intelligent Robots and Systems (IROS), 3283–3290. IEEE.

Codevilla, F.; Miiller, M.; López, A.; Koltun, V.; and Dosovitskiy, A. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–9. IEEE.

Duan, Y.; Jia, J.; and Chen, K. 2022. A survey of embodied AI: From simulators to research tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9103–9121.

Furrer, F.; Burri, M.; Achtelik, M.; and Siegwart, R. 2016. RotorS—A Modular Gazebo MAV Simulator Framework. In *Robot Operating System (ROS): The Complete Reference (Volume 1)*, 595–625. Springer International Publishing.

Garcez, A. d.; and Lamb, L. C. 2020. Neurosymbolic AI: The 3rd wave. *arXiv preprint arXiv:2012.05876*.

Han, D.; Zhang, A.; Chen, R.; Feng, C.; and Guo, S. 2025. Agent in the Sky: Intelligent Multi-Agent Framework for Autonomous HAPS Coordination and Real-World Event Adaptation. In *AAAI 2025 Workshop on Artificial Intelligence for Wireless Communications and Networking (AI4WCN)*.

Huang, W.; Abbeel, P.; Pathak, D.; and Mordatch, I. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, 9118–9147. PMLR.

Hwangbo, J.; Sa, I.; Siegwart, R.; and Hutter, M. 2017. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4): 2096–2103.

Kendoul, F. 2012. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2): 315–378.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1): 90–98.

LaValle, S. M. 1998. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University.

Liu, Q.; Mu, J.; ChenZhang, D.; Liu, Y.; and Hong, T. 2024. LLM enhanced reconfigurable intelligent surface for energy-efficient and reliable 6G IoV. *IEEE Transactions on Vehicular Technology*.

Wu, Z.; Jia, J.; Fu, M.; Ni, Y.; et al. 2023. Large language models for robotics: A survey. *arXiv preprint arXiv:2307.04190*.

Yang, M.; Han, Y.; Ren, Y.; and Li, W. 2024. Large Language Model Guided Reinforcement Learning Based 6 Degree-of-Freedom Flight Control. *IEEE Access*.