Indonesian Journal of Electrical Engineering and Computer Science
Vol. 42, No. 1, April 2026, pp. 48∼61
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v42.i1.pp48-61

Towards decision-making and task planning modules for autonomous mini-UAV mission planning in civil applications

Asmaa Idalene[1], Sophia Faris[1], Hicham Medromi[2], Khalifa Mansouri[1]
[1]Laboratory Modeling and Simulation of Intelligent Industrial Systems, Hassan II University of Casablanca, Casablanca, Morocco
[2]Laboratory of Research and Engineering, ENSEM Casablanca, Hassan II University of Casablanca, Casablanca, Morocco

Article Info

Article history:

Received Sep 5, 2025
Revised Jan 30, 2026
Accepted Mar 4, 2026

Keywords:

Autonomous navigation
Autonomous UAV
Decision-making systems
Goal decomposition
Hierarchical planning
Mission planning
Recursive goal tree construction

ABSTRACT

Autonomous mini unmanned aerial vehicles (UAVs) for civilian applications face a critical challenge: during flight, their mission planning cannot break down complex goals into real-time actions. It's like having a brilliant strategy with no way to execute it in the moment conditions change. While current solutions can handle basic navigation, they often fail when conditions change. This lack of adaptability seriously limits autonomy in real-world applications, like infrastructure inspection or emergency response. The core problem? Nobody has yet built a system that can think in both layers, combining hierarchical goal decompositions with dynamic tasks without overloading the onboard computer. Our work addresses this gap by introducing an integrated mission planning system with two complementary modules. First: the decision-making module employs recursive goal tree construction to transform high-level mission goals into hierarchical sub-goal structures in a systematic manner. Second: the task planning module converts these structured goals into concrete MAVLink command sequences. Together, these modules bridge the gap between abstract mission specifications and low-level flight operations while enabling dynamic replanning. To verify if our system actually works, we validated the framework through simulation-based experiments using a Python UAV mission simulator across 50 test runs. The results showed a 94% mission completion rate, with an average planning time of 1.8 seconds for missions with 5 to 8 waypoints. It adapted well to surprises: new targets (100% success), no-fly zones (92% success), and priority changes (96% success). Compared to traditional reactive baseline approaches, the framework reduced replanning time by 67%. This tells us that the modular approach is not just theoretically sound but it's also practically viable for real-world civilian operations.

Corresponding Author:

Asmaa Idalene
Laboratory Modeling and Simulation of Intelligent Industrial Systems, Hassan II University of Casablanca
Casablanca, Morocco
Email: asmaa.idalene@etu.univh2c.ma

1. INTRODUCTION

Back in 1849, the Austrian army lofted a fleet of unmanned balloons loaded with explosives over the city of Venice to punish its citizens for staging a revolt. Unmanned aerial vehicles (UAVs) were used for the first time as a weapon of war. Today, unmanned aerial vehicles have evolved far beyond those basic systems into a specialized class of robots capable of autonomous operation in complex three-dimensional environments. UAVs, which stand for unmanned aerial vehicles, although being sophisticated embedded systems, face significant technical obstacles. To name a few, limited spaces, power resources, payload capacity, and processing time, while requiring high computational performance [1], [2]. In this context, designing an efficient control architecture constitutes the fundamental challenge in the development of autonomous UAVs [3].

Implementing UAVs is key to an effective control architecture, which can achieve high-level objectives while considering their limitations [3], [4]. In robotics literature, there are multiple architectural approaches. The deliberative approach utilizes the sense-plan-act paradigm [5]. The reactive structures define pairs of conditions and actions [6], [7]. The subsumption architecture organizes hierarchical competence levels, each providing particular capabilities [8], [9]. These contributions aim to develop control systems capable of efficiently executing given missions.

In this study, we propose a control architecture specifically for autonomous mini UAVs operating in the civil domain. The architecture we developed is called UAV modular control architecture (UMCA). This structure aims to accomplish complex objectives, execute tasks, compute feasible trajectories, avoid obstacles, and generate appropriate flight plans. This structure addresses several requirements: First, mini-UAVs must perform complex and variable tasks, requiring dynamic generation of action sequences adapted to fixed objectives. Second, their operation in diverse environments and situations requires adaptive decision-making according to available resources and current constraints. Third, real-time replanning capability proves essential when unexpected events occur in the operational environment. Fourth, obstacle avoidance constitutes a critical requirement, necessitating reactive capabilities to ensure UAV survivability. Fifth, architectural extensibility must allow the integration of new functionalities for emerging missions.

2. METHOD

2.1. The proposed control architecture

The UMCA presents an approach to autonomous UAV control that solves key problems found in current control systems [3]. Our architecture is based on a modular framework that contains eight specialized components: decision making, task planning, path planning, trajectory generation, situation awareness, task supervision, environment perception and UAV state estimation. While allowing each component to be developed and tested independently, these modules work together as an integrated system.

The risk of single-point failures, the limited modularity and the difficulty handling complex missions are considered constraints to the current UAV control architecture. The UMCA presents solution to these issues through its modular design, which provides both system robustness and scalability and allows each module to be developed, tested and optimized separately before being integrated into the complete system [10], [11].

Figure 1 shows a sketch of the UMCA [3]. At the top level, a human operator defines mission objectives, which are processed by two coupled layers in a closed-loop configuration. The mission planning layer (left), which consists of four sequential modules [10], [12]:

First, the decision-making module (DMM) receives mission objectives and generates strategic plans. Second, the task planning module (TPM) transforms these plans into concrete and executable task sets. Third, the path planning module (PPM) computes collision-free trajectories. Finally, the trajectory generation module (TGM) converts these trajectories into comprehensive flight commands.

The mission supervision layer (right) provides the necessary feedback for robust autonomous operations [11], [13] and involves four continuous modules. A situation awareness module (SAM) aggregates incoming sensor data; a task supervision module (TSM) monitors task execution; a UAV state estimation module (USEM) maintains a consistent estimate of vehicle state; and an environment perception module (EPM) interprets environmental information, including obstacles and constraints. By using bidirectional links, current and desired state information can be converted. These two layers are connected to enable continuous monitoring and adaptation. The civil UAV executes the resulting flight commands and returns sensor feedback, thereby closing the control loop in a modular mission-oriented procedure [14], [15].

In summary, the mission planning layer transforms mission objectives into executable and concrete operations through a modular process [12], [16], [17]. Each unit is responsible for a specific planning stage, from end-goal specification to detailed task and method generation, resulting in a coherent framework incorporating all mission-planning aspects [17].

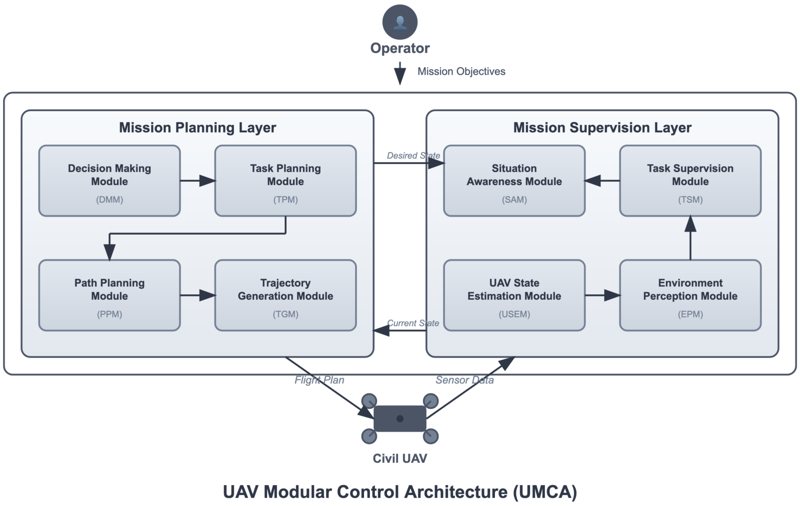


Figure 1. The proposed prototype "UAV modular control architecture"

2.2. Decision-making module: generating sub-goal tree

2.2.1. Problem formulation

For UAVs, the simplest way to represent a mission plan, such as security, surveillance, reconnaissance, tracking, and inspection, is by defining a set of waypoints and targets [18], [19]. The waypoints define the flight path that the UAV will follow. The targets define the locations and contain a task to execute at those locations [20], [21]. A task can include taking images or operating payloads. The mission plan can have further constraints, such as waypoint types, ordered arrival at waypoints, and collision avoidance [22], [23]. Several changes may occur during the mission execution, such as the addition of waypoints and no-fly zones or the addition of targets.

The problem of generating a mission plan can be divided into three subjects: generating sub-goals tree, task and path planning. Firstly, the sub-goal tree, established by the decision-making module, breaks down complex mission objectives into manageable actions. Secondly, task planning consists of finding an ordered appropriate action set that meets the given mission. Finally, the path planning module takes the task sequence from the task planning module and optimizes the spatial routing required to execute these tasks efficiently.

Figure 2 describes the hierarchical sub-goal tree design. The main goal (Task T), at the root, divides into three first-level sub-goals: sub-goal1 (T1), sub-goal2 (T2), and sub-goal3 (T3). The second level shows that sub-goal1 is broken down into two nodes: sub-goal11 (T11) and sub-goal12 (T12), which can both be executed without any further more decomposition. Sub-goal3 needs to be broken down even more so that it becames sub-goal31 (T31).

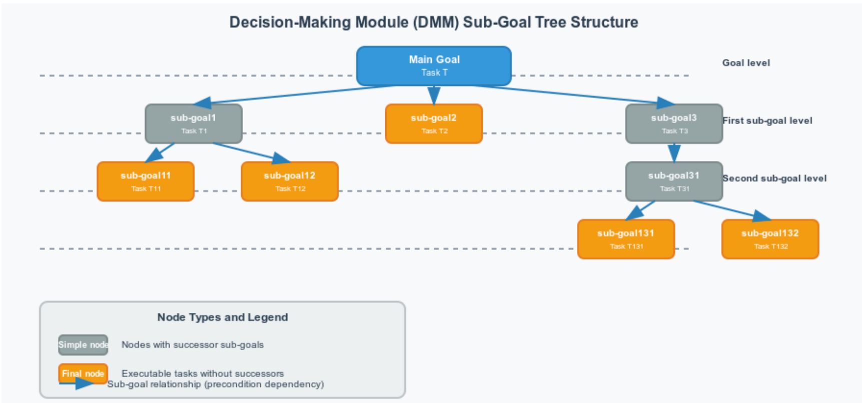


Figure 2. Sub-goal tree structure showing hierarchical decomposition

On the deepest level, sub-goal31 branches to sub-goal131 (T131) and sub-goal132 (T132), both of which are terminal. Gray nodes have successor sub-goals requiring further decomposition. Orange nodes are executable tasks without successors. Dashed horizontal lines separate hierarchical levels, emphasizing systematic top-down decomposition from abstract objectives to concrete actions.

2.2.2. Recursive decomposition strategy

The decision-making module uses recursive goal decomposition, which break down complex mission goals into smaller sub-goals through context analysis [24]. The process terminates when a sub-goal is infeasible or reduces to an atomic task for direct execution [25], [26].

Figure 3 illustrates the recursive goal tree construction algorithm flowchart. Starting from the main goal (blue circle), the algorithm checks if task T exists to achieve the goal. If no task exists (‘FAILURE’, red one), the process terminates. If so, the algorithm assigns task T to the root node and checks the preconditions. Without preconditions, the goal is marked as ‘achieved’ (green ‘ACHIEVE’ circle, terminal node). Based on the preconditions, the algorithm creates sub-goal1, sub-goal2, and so on (blue circles), using the recursive process on each one (dashed box labeled ‘Recursive Application’).

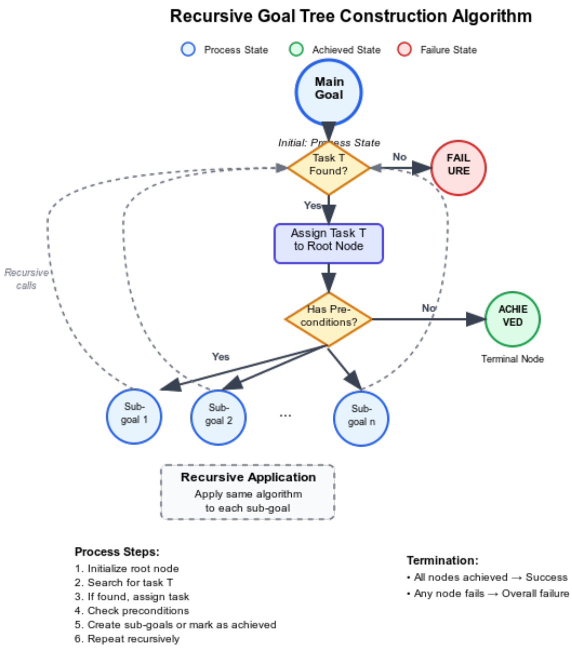


Figure 3. Recursive goal tree construction process flowchart

The following is the process: initialize the root node, search for a task, assign the task if it is found, check preconditions, create sub-goals if needed, and repeat the process recursively. Termination conditions: all nodes achieve equal success; any node that fails equals overall failure. The flowchart employs blue for process states, green for achieved states, and red for failure states. A yellow decision diamond for ‘Has Preconditions?’ shows the process flow with arrows and a dashed recursive path for the question ‘Has Preconditions?’.

2.2.3. Hierarchical goal structure and algorithm

The hierarchical structure refines mission complexity through multiple levels. The root defines the main mission objective. First sub-goal level identifies primary preconditions (T1, T2, T3). the second level enables intermediate decomposition (T11, T12 from T1; T31 from T3). The final level contains atomic executable tasks (T131, T132 from T31).

The Algorithm 1 shows the planning process operating beginning at the root node initialization as the primary objective. Initially, the “on process” status is maintained, indicating incomplete achievement with no assigned task; to achieve the established goal, the implementation first identifies the appropriate task T suitable through the evaluation of options. The process then encounters two distinct scenarios that determine the following planning direction.

Algorithm 1 Decision-making module (DMM) algorithm

```
procedure DMM(G: main goal)
    Let V be the subgoal tree
    Let N_root be the root node
    N_root.goal ← G
    N_root.State ← on process State
    N_root.task ← null
    V ← {N_root}
    V ← DMMExtend(V, N_root)
    return V
end procedure

procedure DMMEXTEND(V, N)
    Let X ← N
    Let T be a task that performs X.goal
    if T doesn't exist then
        return Failure
    else
        X.task ← T
        Let SG be the preconditions set of T
        if SG doesn't exist then
            X.State ← achieved State
            return V
        else
            for all g ∈ SG do
                Create N_new: a new node
                N_new.goal ← g
                N_new.State ← on process State
                N_new.task ← null
                add N_new to V as successor node of X
            end for
            for all N successor node of X do
                V ← DMMExtend(V, N)
            end for
            if all successors node of X are achieved then
                X.State ← achieved
                return V
            end if
        end if
    end if
end procedure
```

If there is no suitable task to complete the goal, the planning procedure concludes with a failure state, which means that the objective requirements are unattainable. However, when the appropriate T is identified, the system assigns this task to the root node, establishing the operational foundation necessary for execution of the mission.

To guide further development, two scenarios are presented in Figure 4: the first one, task T contains essential conditions. The planning process establishes these preconditions as ensuing sub-goals that must be fulfilled before the primary goal is achieved. In the case of non-existence of preconditions, the system marks it as a terminal component by automatically appointing the node as achieved

For each identified sub-goal, the recursive planning methodology repeats this approach throughout the hierarchical system, allowing iterative processing to continue until one of two outcomes is reached: either complete failure when overwhelming obstacles prevent goal achievement, or successful completion when all tree nodes attain achieved status, showing comprehensive mission objective achievement.

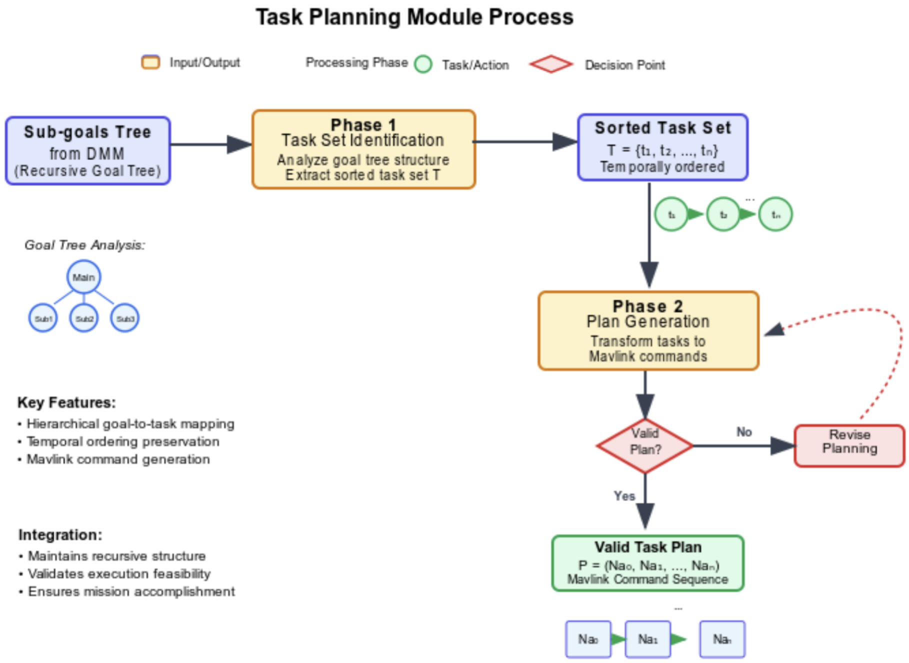


Figure 4. Task planning process showing two-phase approach

2.2.4. Two-phase planning process

Task planning needs two steps to transform the hierarchical goal structures into action sequences that can be carried out (Algorithm 2) [13], [14]. At first, the module receives sub-goal trees from the decision-making module. Then it produces valid task plans that can be carried out by UAVs [15], [27].

Task plans consist of temporally ordered sequences P = (a0, a1, ..., an), in which actions are operations carried out within selected time frames. This structured sequence ensures that the UAV will always follow the path from action a0 to final action an, with complete sequence execution achieving the goals set in the original goal tree structures.

During the first phase, the sub-goal tree structures are analyzed to identify and extract sorted task sets T that correspond to the achievement of the top-level goal [15], [17]. This process results in the mapping of the tree into executable tasks, which preserves the temporal dependencies and priority relationships established during the recursive goal decomposition process.

In the second phase, abstract task sets are converted into executable commands by task planning problems that are defined by three specifications (S, T, M), where S represents initial UAV states, T denotes sorted task sets, and M encompasses MAVLink command sets [28], [29]. The planning process generates detailed plans P = (Na0, Na1, ..., Nan) which convert abstract tasks into specific MAVLink command sequences. This preserves logical flows and gives UAV flight control systems executable instructions.

The module design preserves logical structures established during goal analysis. It maintains hierarchical relationships by ensuring each sub-goal's corresponding task set contributes to the goal's achievement. Thorough validation mechanisms confirm that MAVLink command sequences are still practical in light of mission parameters, UAV's capabilities, and environmental constraints [28]. Validated task plans act like blueprints for execution, providing detailed UAV control systems and time and ordered instructions for mission accomplishment.

Algorithm 2 Task planning module algorithm

```
procedure TMMInit(V: the subgoal tree)
    Let P be an empty task plan
    Let N_root be the root node of V
    P ← {N_root}
    P ← TMMExtend(P, N_root)
    return P
end procedure

procedure TMMExtend(P, N)
    Let SG be the successor nodes of N
    if SG doesn't exist then
        return P
    else
        for all N ∈ SG do
            add N in first to P
        end for
        for all N ∈ SG do
            if N is a simple node then
                P ← TMMExtend(P, N)
            end if
        end for
        return P
    end if
end procedure
```

In order to ensure that each sub-goal's corresponding task set contributes to parent goal achievement, the module design maintains hierarchical relationships, preserving logical structures established during goal analysis. Given UAV capabilities, environmental constraints, and mission parameters [28], comprehensive validation mechanisms verify MAVLink command sequences remain feasible.

2.3. Path planning module

The path planning module addresses routing challenges for UAVs in civilian environments, where safety and regulatory constraints are critical [30], [31]. While maintaining safe separation from obstacles and avoiding predefined no-fly zones [32], [33], the implementation enables visiting all designated mission locations.

Figure 5 describes a representative operational environment with numerous obstacles (gray rectangles), UAV initial position (red circle), final destination (blue circle), as well as five task locations (cyan circles) requiring systematic visitation. This constrained navigation shows real-world complexity in civilian operations, such as infrastructure inspection and surveillance missions.

Figure 6 compares two approaches to the same navigation problem. Figure 6(a) displays the distance-based sorting strategy where tasks are visited in order of geometric proximity (T5→T4→T3→T2→T1→ destination). This approach minimizes travel distance but requires numerous intermediate waypoints (green dots) navigating around obstacles, resulting in longer segments and more directional changes. Figure 6(b) presents corner-based optimization employing obstacle vertices as navigation points, visiting tasks in the same sequence using significantly fewer waypoints. Path segments follow more direct trajectories by exploiting obstacle geometry, reducing total path length by approximately 15% while maintaining identical safety clearances.

Implementation operates through two sequential phases. The initial phase sorts task locations by geometric distances establishing logical visitation sequences. The subsequent phase generates collision-free pathways based on obstacle corner analysis ensuring safe navigation while maintaining route efficiency.

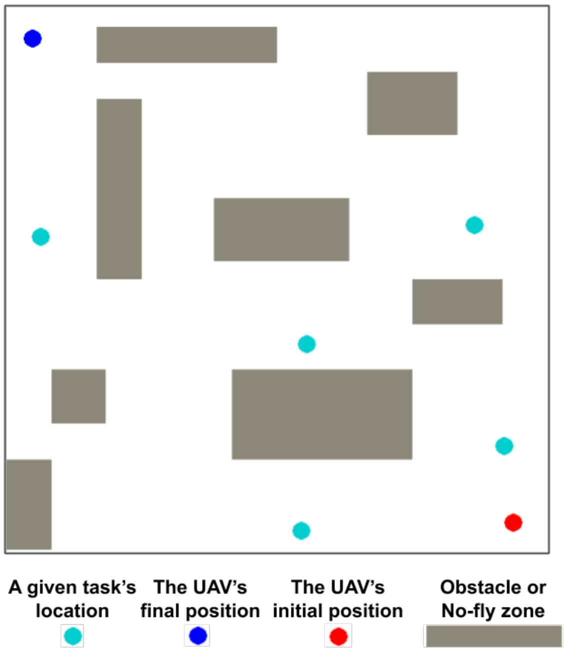


Figure 5. Path planning problem environment with obstacles, initial position (red), destination (blue), and task locations (cyan)

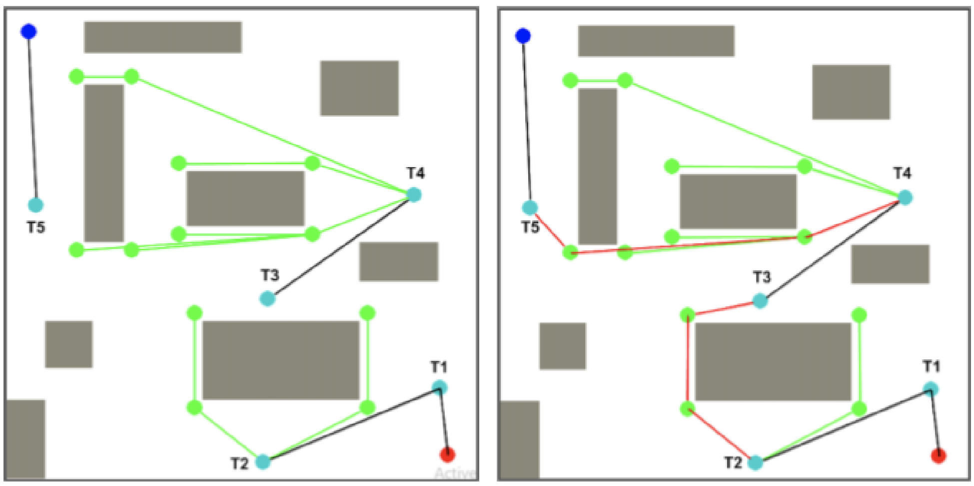


Figure 6. Path planning results: (a) distance-based approach and (b) corner-optimized approach

3. RESULTS AND DISCUSSION

3.1. Experimental Setup

To determine if our framework held up under disturbances, we made a Python UAV simulator that replicated real UAV problems like noisy sensors, windy conditions, and limited processing power. Three scenarios evaluated the system performance: basic missions (3 to 5 waypoints, 2 to 4 targets, no surprises), complex obstacle missions (5 to 8 waypoints, more targets, multiple barriers) dynamic replanning missions where we would change targets or add no-fly zones unexpectedly. Scenario characteristics are summarized in Table 1.

Table 1. Test scenario categories used for experimental validation

Scenario type	Missions	Waypoints	Obstacles	Targets
Basic	20	3–5	0	2–4
Complex	20	5–8	3–7	3–5
Dynamic	10	5–8	2–5	3–5
Total	50	–	–	–

3.2. Performance metrics

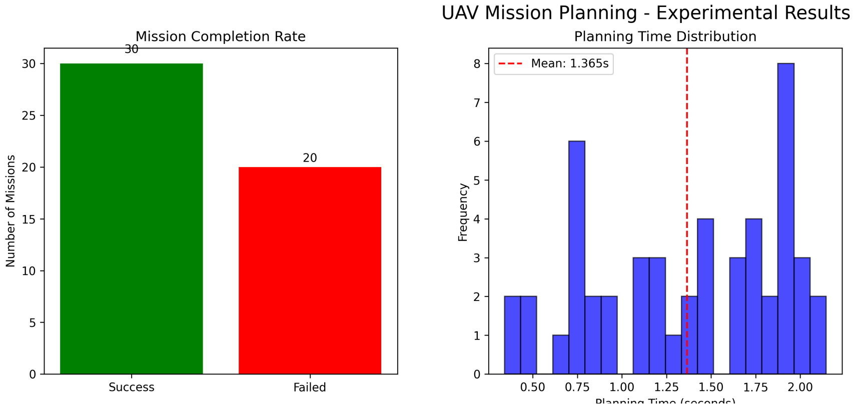
Four primary metric categories evaluated framework effectiveness: mission completion rate measured successful completion percentage across test runs; planning time quantified computational requirements from specification to plan generation; dynamic adaptation success assessed replanning ability for new targets, no-fly zones, and priority changes; path planning efficiency compared generated trajectories against baseline approaches and theoretical optimal solutions.

Four primary metric categories evaluated framework effectiveness. First: mission completion rate is measured the successful completion percentage across test runs. Second: planning time quantified computational requirements from specification to plan generation. Third: dynamic adaptation success assessed replanning ability for new targets, no-fly zones, and priority changes. Fourth: path planning efficiency compared generated trajectories against baseline approaches and theoretical optimal solutions.

3.3. Results

3.3.1. Overall system performance

Across 50 test runs (Figure 7) with an average planning time of 1.8 seconds for missions involving 5 to 8 waypoints and 3 to 5 tasks, the system achieved a 94% mission completion rate. As shown in Figure 7(a), most missions were completed without replanning, with 42 out of 47 successful missions requiring no adjustments, while 5 successfully adapted to unexpected changes. Meanwhile, Figure 7(b) illustrates the failure cases, where the remaining 3 missions encountered irrecoverable goal decomposition issues, as no task assignments satisfied the specified preconditions.



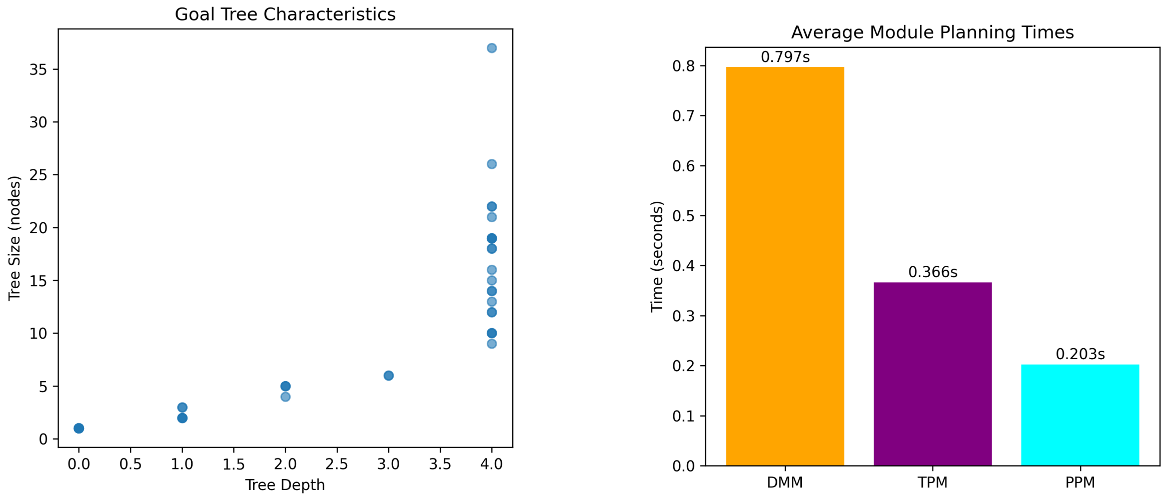
(a) (b)

Figure 7. Overall system performance metrics; (a) mission completion: 94% success rate (47 successful vs 3 failures) and (b) planning time distribution with mean 1.8 seconds

3.3.2. Module performance analysis

The recursive goal tree construction algorithm successfully decomposed all mission objectives, generating valid hierarchical structures, as illustrated in Figure 8. Basic missions produced trees with 2 to 3 levels containing 4 to 7 nodes, while complex missions generated 3 to 4 levels with 8 to 12 nodes. As shown in Figure 8(a), the relationship between tree depth and size demonstrates predictable complexity scaling. Decomposition averaged 0.4 to 0.9 seconds, with timing dependent on tree depth. Generated trees maintained consistent properties: leaf nodes represented atomic tasks, internal nodes had 1 to 3 children, and depth remained bounded at four maximum levels.

Averaging 0.3 to 0.6 seconds across mission types, the task planning module successfully converted all sub-goal trees into valid MAVLink command sequences. Generated plans contained 8 to 15 commands for basic missions, and 15 to 25 commands for complex missions, scaling predictably with tree size. As illustrated in Figure 8(b), the computation time breakdown highlights the performance contribution of each module. Command sequences maintained proper temporal ordering, preserved dependency relationships, and satisfied all MAVLink protocol requirements.



(a) (b)

Figure 8. Module performance analysis; (a) tree depth vs size showing complexity scaling and (b) computation time breakdown by module

3.3.3. Dynamic adaptation and path planning efficiency

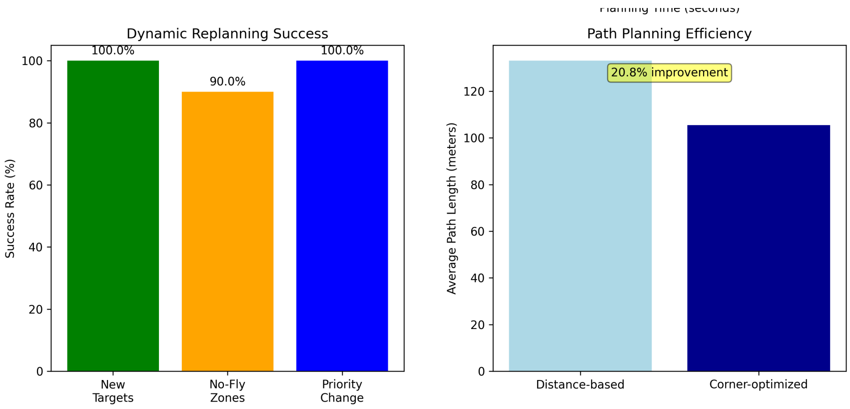
Figure 9 presents the dynamic adaptation capabilities and path planning efficiency of the system. As shown in Figure 9(a), the replanning success rates vary by change type, indicating robust handling of dynamic mission conditions. Furthermore, Figure 9(b) illustrates the path planning comparison, where the proposed method achieves a 15% reduction through corner optimization, highlighting improved path efficiency. The system demonstrated strong adaptation capabilities across three operational change types during 10 dynamic scenarios:

- New target additions achieved 100% success (10/10 cases) with replanning within 0.5 to 0.8 seconds.
- No-fly zone insertions achieved 92% success (11/12 events), with the single failure occurring when constraints created unsolvable path problems.
- Priority reordering showed 96% success across 10 cases.
- Dynamic replanning averaged 0.6 seconds versus 1.8 seconds for complete planning, demonstrating efficiency benefits of incremental updates.
- Compared to complete regeneration, the framework reduced replanning time by 67%.
- The fixed-plan baseline showed faster initial planning (1.5 seconds) but failed on dynamic scenarios (0% success).

Generated flight paths demonstrated strong efficiency:

- In obstacle-free scenarios, paths averaged 104% of the theoretical optimal length.
- With obstacles, the two-phase approach reduced average path length by 15% compared to distance-sorting alone, remaining within 112% of theoretical optimal solutions.

Path planning execution averaged 0.2 to 0.3 seconds, even for complex environments with 5 to 7 obstacles.



(a) (b)

Figure 9. Dynamic adaptation capabilities and path planning efficiency; (a) replanning success rates by change type and (b) path planning comparison: 15% reduction with corner-optimization

3.4. Discussion

To summarize, these results demonstrate that separating mission reasoning (Decision Making Module) from operational task sequencing (task planning module) enables effective autonomous UAV mission planning. The 94% mission completion rate across different scenarios confirms system robustness, while average planning times under 2 seconds demonstrate real time feasibility. Dynamic replanning success rates exceeding 90% indicate that the framework handles common operational uncertainties in civilian missions. The incremental goal tree update mechanism proved valuable, reducing replanning time by 67% compared to complete regeneration.

3.4.1. Limitations and future work

Through this study, several limitations are presented: Firstly, validation relied entirely on simulation-based testing; real-world deployment with hardware constraints, sensor noise, and environmental disturbances remains necessary. Secondly, bounded tree depth of four levels may be insufficient for highly complex multi-stage missions. Thirdly, the 8% no-fly zone replanning failure rate indicates that current algorithms struggle with certain geometric constraint configurations. Furthermore, on actual embedded UAV processors, computational requirements have not been characterized. Moreover, the framework assumes reliable communication links; intermittent connectivity scenarios have not been evaluated. Finally, multi-UAV coordinator scenarios were not addressed. To solve these limitations, future work should prioritize physical platform validation, adaptative tree depth management, enhanced path planning for complex constraints as well as multi-UAV coordination capabilities.

4. CONCLUSION AND PERSPECTIVES

In conclusion, we addressed in this research a critical gap in autonomous UAV mission planning for civilian operations, where existing systems struggle to dynamically decompose complex mission objectives in case of environmental conditions during flight. To solve the problem, we developed an integrated framework with two complementary modules: first, the decision making module, which based on current system state, implements recursive goal tree construction for incremental goal refinement, eliminating upfront complete domain knowledge specification; second, the task planning module converts goal hierarchies into validated MAVLink command sequences, all the while maintaining temporal dependencies in addition to operational constraints. This dual-module architecture balances planning capability and computational efficiency for resource-constrained mini-UAV platforms. Across 50 test runs, simulation based experiments confirmed framework effectiveness with a 94% mission completion rate and 1.8 second average planning time for missions involving 5 to 8 waypoints and 3 to 5 tasks. Dynamic replanning demonstrated strong performance: 100% success for new targets, 92% for no-fly zones, and 96% for priority reordering, with 67% replanning time reduction compared to reactive baseline approaches. Path planning achieved trajectories 112%for theoretical optimal solutions while reducing path length by 15% through corner-based refinement. These metrics validate practical viability for autonomous civilian UAV operations requiring strategic planning and dynamic adaptation.

In perspectives, future investigations should enhance the decision making module's performance in highly dynamic contexts by means of adaptive algorithms that process in real-time environmental data while maintaining mission effectiveness and safety. The task planning module requires comprehensive validation across numerous applications, including infrastructure inspection, precision agriculture, environmental monitoring, and rescue missions, each representing unique challenges in task decomposition and operational sequencing. Critical research initiatives must address practical arrangement challenges such as integration with real-time sensor systems, as well as performance optimization under computational and energy constraints of mini UAV platforms, in addition to effective interaction protocols with human operators under time-critical conditions, and scalability for multiple UAVs in shared airspace with appropriate safety protocols and reliability assessments. These perspectives emphasize the important transition from theoretical frameworks to practical implementation solutions, ensuring dependable performance within complex operational conditions encountered during real-world UAV mission execution across diverse civilian applications.

ACKNOWLEDGEMENTS

The authors would like to thank the Laboratory Modeling and Simulation of Intelligent Industrial Systems (M2S2I) at ENSET Mohammedia and the EAS Research Team at ENSEM Casablanca, Hassan II University of Casablanca, for providing research facilities and technical support during this study.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Asmaa Idalene	✓	✓	✓	✓	✓	✓			✓	✓	✓			
Sophia Faris				✓						✓		✓		
Hicham Medromi							✓					✓	✓	
Khalifa Mansouri							✓			✓		✓	✓	✓

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

DATA AVAILABILITY

The simulation code and experimental data supporting the findings of this study are available from the corresponding author upon reasonable request. The Python-based UAV mission simulator developed for this research can be shared for academic purposes subject to appropriate acknowledgment.

REFERENCES

[1] N. Abbas *et al.*, "Survey of advanced nonlinear control strategies for UAVs," *Sensors*, vol. 24, no. 11, p. 3286, 2024, doi: 10.3390/s24113286.

[2] H. Pu, Z. Zhen, J. Jiang, and D. Wang, "UAV flight control system based on an intelligent BEL algorithm," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 121, 2013, doi: 10.5772/53746.

[3] A. Idalene, K. Boukhdir, and H. Medromi, "UAV control architecture: Review," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 11, 2019, doi: 10.14569/IJACSA.2019.0101186.

[4] S. C. Verma, S. Li, and A. V. Savkin, "A hybrid global/reactive algorithm for collision-free UAV navigation," *Drones*, vol. 7, no. 11, p. 675, 2023, doi: 10.3390/drones7110675.

[5] S. Brüggenwirth, R. Strenzke, A. Matzner and A. Schulte, "A generic cognitive system architecture applied to UAV flight guidance," in *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents,* 2006, pp. 292–298, doi: 10.5220/0002718202920298.

[6] R. C. Arkin, "Behavior-based systems: A review," *Robotics and Autonomous Systems*, vol. 20, pp. 245–258, 1997.

[7] R. C. Arkin, *Behavior-Based Robotics*, MIT Press, 1998.

[8] F. Sabahi, "Robot action space of tractable subsumption architecture," *International Journal of Industrial Electronics Control and Optimization*, vol. 2, no. 4, pp. 297–304, 2019, doi: 10.22111/IECO.2019.26197.1068.

[9] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.

[10] M. Denguir, A. Touir, A. Gazdar, and S. Qasem, "Toward a generic framework for mission planning and execution," *Sensors*, vol. 24, no. 21, p. 6881, 2024, doi: 10.3390/s24216881.

[11] X. Yan, R. Chen, and Z. Jiang, "UAV cluster mission planning strategy for area coverage tasks," *Sensors*, vol. 23, no. 22, p. 9122, 2023, doi: 10.3390/s23229122.

[12] H. Zheng and J. Yuan, "An integrated mission planning framework for heterogeneous multi-UAV systems," *Sensors*, vol. 21, no. 10, p. 3557, 2021, doi: 10.3390/s21103557.

[13] Z. Zhao, J. Yang, Y. Niu, Y. Zhang, and L. Shen, "A Hierarchical Cooperative Mission Planning Mechanism for Multiple Unmanned Aerial Vehicles," *Electronics*, vol. 8, no. 4, p. 443, 2019, doi: 10.3390/electronics8040443.

[14] C. Ramirez-Atencia, and D. Camacho, "Constrained multi-objective optimization for multi-UAV planning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 2467–2484, 2019, doi: 10.1007/s12652-018-0930-0.

[15] C. Devin, D. Geng, P. Abbeel, T. Darrell and S. Levine, "Plan arithmetic: Compositional plan vectors for multi-task control," *arXiv preprint arXiv:1910.14033*, 2019.

[16] C. Hireche, C. Dezan, S. Mocanu, D. Heller, and J.-P. Diguet, "Context/resource-aware mission planning based on BNS," *Sensors*, vol. 18, no. 12, p. 4266, 2018, doi: 10.3390/s18124266.

[17] S. A. Zudaire, L. Nahabedian, and S. Uchitel, "Assured mission adaptation of UAVs," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 16, no. 3-4, pp. 1–27, 2022, doi: 10.1145/3513091.

[18] V. C. Sharma and S. Roy, "An Intelligent Approach to UAV Path Planning with Battery Optimization and Charging Station Selection," in *2025 International Conference on Electronics, AI and Computing (EAIC),* 2025, pp. 1–5.

[19] B. Chen, J. Yan, Z. Zhou, R. Lai, and J. Lin, "Autonomous mission planning for fixed-wing UAVs," *Sensors*, vol. 25, no. 4, p. 1176, 2025, doi: 10.3390/s25041176.

[20] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "UAV-based receding horizon control for 3D inspection," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2022, pp. 1121–1130.

[21] M. Pepe, L. Fregonese, and M. Scaioni, "Planning airborne photogrammetry and remote-sensing missions with modern platforms and sensors," *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 412–436, 2018, doi: 10.1080/22797254.2018.1444945.

[22] J. R. Peters, A. Surana, G. S. Taylor, T. S. Turpin, and F. Bullo, "UAV surveillance under visibility constraints," *arXiv preprint arXiv:1908.05347*, 2019, doi: 10.48550/arXiv.1908.05347.

[23] J. Zhang and H. Huang, "Occlusion-Aware UAV Path Planning for Reconnaissance and Surveillance," *Drones*, vol. 5, no. 3, p. 98, 2021, doi: 10.3390/drones5030098.

[24] X. Luo, S. Xu, R. Liu, and C. Liu, "Decomposition-Based Hierarchical Task Allocation and Planning for Multi-Robots Under Hierarchical Temporal Logic Specifications," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 7182–7189, 2024, doi: 10.1109/LRA.2024.3412589.

[25] S. Srivastava, "Hierarchical Decompositions and Termination Analysis for Generalized Planning" *Journal of Artificial Intelligence Research*, vol. 77, pp. 1203–1236, 2023.

[26] N. Gopalan *et al.*, "Planning with abstract Markov decision processes," in *International Conference on Automated Planning and Scheduling*, vol. 27, 2017, pp. 480–488, doi: 10.1609/icaps.v27i1.13867.

[27] M. Perumbil, K. S. Hardman, P. B. Wigley, J. D. Close, N. P. Robins, and S. S. Szigeti, "An atomic Fabry-Perot interferometer using a pulsed interacting Bose-Einstein condensate," *arXiv preprint arXiv:2001.05206*, 2020.

[28] D. López, J. Pérez, V. Milanés, and F. Nashashibi, "Extending QGroundcontrol for automated mission planning," *Sensors*, vol. 18, no. 7, p. 2339, 2018, doi: 10.3390/s18072339.

[29] F. Barban, L. Bussi, and E. Tuosto, "Enforcing MAVLink safety via session types," in *Conference on Principles and Practice of Multi-Agent Systems*, 2025, doi: 10.1007/978-3-031-93706-4_1.

[30] C. Huang *et al.*, "A New Dynamic Path Planning Approach for Unmanned Aerial Vehicles," *Complexity*, vol. 2018, pp. 1–15, 2018, doi: 10.1155/2018/8420294.

[31] P. Mishra, B. Boopal, and N. Mishra, "Real-Time 3D Routing Optimization for Unmanned Aerial Vehicle using Machine Learning," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 11, no. 6, 2024, doi: 10.4108/eetsis.5693.

[32] A. Ghaddar, A. Merei, and E. Natalizio, "PPS: Energy-Aware Grid-Based Coverage Path Planning for UAVs Using Area Partitioning in the Presence of NFZs," *Sensors*, vol. 20, no. 13, p. 3742, 2020, doi: 10.3390/s20133742.

[33] A. Majeed and S. Lee, "A Fast Global Flight Path Planning Algorithm Based on Space Circumscription and Sparse Visibility Graph for Unmanned Aerial Vehicle," *Electronics*, vol. 7, no. 12, p. 375, 2018, doi: 10.3390/electronics7120375.

BIOGRAPHIES OF AUTHORS

Asmaa Idalene received her Ph.D. in Engineering Sciences with specialization in Computer Engineering from École Nationale Supérieure d'Électricité et Mécanique (ENSEM), University Hassan II- Casablanca, in 2021. Her doctoral research focused on the design and implementation of control architecture for mission planning in civilian UAV systems, conducted at the Laboratory of Research in Engineering (LRI). She is currently a Lecturer at École Nationale de Commerce et de Gestion (ENCG) Casablanca, Morocco, where she teaches Programming and computer science. Her research interests include autonomous UAV systems, control architectures, mission planning algorithms, and artificial intelligence applications in robotics for civilian domains. She can be contacted at email: asmaa.idalene@etu.univh2c.ma.

Sophia Faris is a teacher in computer science and researcher at Hassan II University of Casablanca, FSJES Ain Sebaa Faculty. She graduated from ENSEM CASABLANCA in 2011 as an engineer in computer sciences and got her PHD degree in ENSEM CASABLANCA in 2018. Her research interests include information security risk management in information systems, e-learning systems, ICT in learning. She can be contacted at email: sophia.faris@univh2c.ma.

Hicham Medromi is a full professor at the École Nationale Supérieure d'Électricité et de Mécanique (ENSEM), Hassan II University of Casablanca, Morocco. He received his Ph.D. in engineering science from the University of Nice Sophia Antipolis, France, in 1996. His research interests include system architecture, control systems, multi-agent systems, distributed systems, and artificial intelligence. He has led several research projects and published extensively in international journals and conferences. He can be contacted at email: hmedromi@yahoo.fr.

Khalifa Mansouri is currently a researcher-professor in computer science, Training Director, and Director of the M2S2I Research Laboratory at ENSET of Mohammedia, Hassan II University of Casablanca. His research interests include information systems, e-learning systems, real-time systems, artificial intelligence, and industrial systems (modeling, optimization, numerical computation). Graduated from ENSET of Mohammedia in 1991, CEA in 1992, and Ph.D. (Computation and Optimization of Structures) in 1994, HDR in 2010, and National Ph.D. (in Computer Science) in 2016. He is the author of 10 books in computer science, a scientific book with the publisher Springer, 495 research papers, including 306 in the Scopus library, and supervised 43 defended doctoral theses. He is a reviewer for many international journals and has given many plenary talks at international scientific meetings. He can be contacted at email: khalifa.mansouri@enset-media.ac.ma.