MINIMUM TIME SEARCH AND RESCUE MISSIONS INVOLVING UAVS WITH A FRAMEWORK SUPPORTED BY DEVS, RHC AND PEA

Juan B. Bordón-Ruiz[a], Eva Besada-Portas[a], José L. Risco-Martín[a], and José A. López-Orozco [a]

[a]Department of Computer Architecture and Automation, Universidad Complutense de Madrid (UCM), Spain

ABSTRACT

Search and rescue missions of moving targets undeterministically placed can be formulated as probabilistic optimization problems where UAVs trajectories and sensor poses are determined according to multiple constraints and objective functions. To bring realism and flexibility to the involved simulations and optimizers, we developed a framework supported by Discrete Event System Specification (DEVS) and a Multi-Objective Evolutionary Algorithm (MOEA) working as a Receding Horizontal Controller (RHC). Its new version, presented here, is enhanced by the introduction of several MOEAs, distributable to cooperate in the Island Model (IM) of the Parallel Evolutionary Algorithm (PEA) paradigm, and by the transmission of multiple solutions between consecutive iterations of the RHC used to break the optimization into subproblems. The combination of this improvements outperforms the previous version of the framework by: 1) ensuring viable trajectories in scenarios where the original MOEA was prone to fail, and by 2) improving all optimization objectives.

Keywords: Search and Rescue, Island Model, Discrete Event Systems Specification, UAV Path Planning.

1 INTRODUCTION

Optimal Search Theory, whose origins trace back to World War II [1], addresses the search for targets placed at unknown locations within a given area [2]. Its current applicability to real-world scenarios is broad and varied [3], including civil domains (e.g. wildlife observation, or maritime search & rescue) and military missions (e.g. tactical reconnaissance, or persistent surveillance). Many of these tasks involve *dull, dirty, or dangerous* operations for humans, which can be effectively performed nowadays in large-scale environments by Unmanned Aerial Vehicles (UAVs), remotely or autonomously controlled without humans onboard, and equipped with sensors tailored to each type of missions [4, 5].

Assuming a directional search with targets not reacting to the movements of the searchers to avoid detection, the theory quickly expanded into the fields of *Operational Research* [6], *Optimal Control* [7], and *Information Fusion* [8]. The works within the third field (as this paper) exploit Bayes theory to: 1) represent the uncertainty associated to the target initial location and its possible movements; 2) update the uncertainty about the target location according to the UAVs trajectory and the likelihood of the observations from their onboard sensors; and 3) determine the success (for the mission) of the UAVs trajectories using probability-based utility functions such as the *Probability of Detection* (PD) or the *Expected Time of Detection* (ETD) [9, 8, 10, 11]. The complementary information provided by both functions is specially interesting for minimum time search & rescue missions, as maximizing PD increases the chances to detect the target during the mission, while minimizing a truncated version of ETD increments the chances to find it soon [12, 13].

As the problem is NP-Hard for a single UAV [14] and NEXP-Complete for multiple UAVs [9], real world instances of the problem are being tackled with Multi-Objective Evolutionary Algorithms (MOEAs), capable

Proc. of the 2025 Annual Modeling and Simulation Conference (ANNSIM'25)
May 26-29, 2025, Universidad Complutense de Madrid, Madrid, Spain
J.L. Risco-Martín, G. Rabadi, D. Cetinkaya, R. Cárdenas, S. Ferrero-Losada, and A. Bany Abdelnabi, eds.

of efficiently exploring the solution space of this optimization problem [9, 12, 13]. They are specially suited for minimum time S&R problems, since they can 1) simultaneous optimize PD and ETD; 2) incorporate also *mission-specific* constraints and objectives (e.g. UAVs collision and nonflying zones avoidance, or fuel consumption); and, therefore, 3) obtain effective UAV trajectories and sensor poses, according to the target detection possibilities and the real-world mission operation.

MOEAs manipulate a population of candidate solutions, which evolve iteratively through selection, crossover, and mutation, ensuring the survival of the solutions better-fitted to the utility functions. In this problem, the UAVs trajectories are usually encoded in the population as sequences of reference signals (e.g. UAV intended heading, speed and height, or sensor intended orientation). Hence, the UAV trajectories and sensor poses, required to compute PD and ETD, are often obtained via simulations of (usually deterministic) motion models. In particular, the probabilistic models of the target and sensor observation, the deterministic models of the UAV and sensor motion, and the utility functions have already been successfully defined and simulated in [15, 16], following the Discrete Event System Specification (DEVS) within xDEVS engine [17, 18]. This way of proceeding ensures a clear separation between the abstraction level (DEVS specification) and the technical implementation (of each target, UAV and sensor specific model), favors posterior improvements and refinements, and supports the interoperability among heterogeneous types of models (e.g. probabilistic/deterministic, periodic/aperiodic, and syncronous/asyncronous).

This target-search simulator has been integrated into a MOEA within a unified DEVS framework for Modeling, Simulating and Optimizing (M&S&O) target-search missions [19]. The framework simplifies the connection amongs its elements, since the DEVS coordinator transparently manages the optimization process and the target-search simulations. This framework also divides the optimization process in two levels. At the top, it operates as a Receding Horizon Controller (RHC), breaking the decision horizon into parts to optimize the sub-trajectories, running, in turn and at the bottom, a MOEA from the final UAVs locations, sensors poses and target probability distribution of the previous sub-trajectory. This division favors the use of different MOEAS, without altering the remaining structure and behavior of the framework. Finally, DEVS and MOEAs are inherently suitable for parallel and distributed computation, a property that the framework has already exploited to accelerate the optimization of the UAVs trajectories and sensor poses using x-DEVs capability to deploy and distribute simulations in the *Cloud*.

This paper presents the new version of this M&S&O framework for target-search missions, which exploits the parallelization and distribution capabilities of MOEAs. These capabilities have been exploited by Parallel Evolutionary Algorithms (PEAs, [20, 21, 22, 23]) and their foundational Island Model (IM, [24]), which divide the population into subpopulations that are independently optimized on separate processors and that periodically exchange a few individuals during a *migration* step. Beyond the computational speedup that can be obtained with these approaches, other benefits, such as faster convergence and improved solution quality, have been demonstrated. Although their applications to target-search problems involving UAVs is limited, they have been used in some works of route optimization for UAVs that address similar challenges [25, 26]. The new contributions of the framework that are introduced in this paper are:

- *Several MOEAs can be used at the bottom layer.* While the version in [19] was only supported by the 2^{nd} Non-dominated Sorted Genetic Algorithm (NSGA-II, [27]), the current one can also use the 2^{nd} Strength Pareto Evolutionary Algorithm (SPEA-II, [28]), the Non-dominated Sorting Particle Swarm Optimizer (NSPSO, [29]), and the Multi-Objective Differential Evolution (MODE, [30]).
- *A new internal migration mechanism for RHC-based optimization.* Instead of selecting the best solution of the last decision horizon to determine the start point of the current one, we pass the first Pareto front in order to enhance diversity and reduce the myopic effect of the RHC.
- *Island Model support to improve the quality (feasibility and optimality) of the solutions (UAVs trajectories and sensors poses).* After observing that different MOEAs were better suited for different

target-search scenarios, we include an IM with adjustable parameters (e.g. topology, MOEA for each node, and migration frequency and policy) in our framework to be able to exploit the capabilities of NSGA-II, SPEA-II, NSPSO and MODE into the same optimization process.

2 BACKGROUND

This section introduces the formulation of the target-search problem used within the framework in [19] and some background required to understand the novelties in the current version introduced in this paper.

2.1 Formulation of the Minimum Time S&R Problem

A group of U UAVs equipped each with K_u onboard sensors searches a single target, whose unknown location at time t within the search space Ω is represented by the random variable $\tau^t \in G_\Omega$, where G_Ω is the grid of $N_x \times N_y$ cells, each of size $w_x \times w_y$, in which Ω has been discretized. Each UAV trajectory is represented by $s_u^{0:T}$ and the sequence of poses of one of its sensors by $s_{u,k}^{0:T}$, both types of variables continuous.

The UAVs and sensors motions are modeled with deterministic differential equations and integrated using the fourth-order Runge-Kutta numerical method. The process for the UAV motion, summarized by Equation (1) of Table 1, calculates the new UAV state s_u^t based on its previous state $s_u^{t-T_u}$, the UAV reference input $a_u^{t-T_u}$, environmental conditions ε^{t-T_u}, and the sampling period T_u. Similarly, the process for the sensor motion, summarized by Equation (3), calculates the new sensor pose $s_{u,k}^t$ based on its previous pose $s_{u,k}^{t-T_{u,k}}$, the sensor reference input $a_{u,k}^{t-T_{u,k}}$, the UAV current state s_u^t, and the sampling period $T_{u,k}$. The UAV reference a_u^t contains, as Equation (2) states, the requested UAV heading θ_u^t, speed v_u^t, and altitude h_u^t. Similarly, the sensor reference a_u^t contains, according to Equation (4), the requested sensor azimuth $\alpha_{u,k}^t$ and elevation $\eta_{u,k}^t$. The UAVs and sensor references remain respectively unchanged during T_{a_u} and $T_{a_{u,k}}$, and constitute the decision variables that can be encoded and manipulated by the MOEAs.

The knowledge about the initial position of the target is modeled with the initial belief $b(\tau^0)$, which represents the probability of finding the target in each cell of G_Ω. The target motion is modelled with the Markovian probability function $p(\tau^t|\tau^{t-T_\tau})$, which represents the chances of reaching any $\tau^t \in G_\Omega$ from any given τ^{t-T_τ}. The prediction step of Recursive Bayesian Filter (RBF, [10]), stated in Equation (6), indicates how the belief of a given cell is distributed, after a T_τ lapse, among all the cells in the grid G_Ω.

The uncertainty on the sensor detections is represented by the sensor likelihood $p(D|\tau^t, s_u^t, s_{u,k}^t)$, which states the probability that sensor k of UAV u, both in their corresponding states $s_{u,k}^t$ and s_u^t, detects D the target at cell $\tau^t \in G_\Omega$. The assimilation step stated in Equation (4), updates $b(\tau^t)$ whenever the sensor should observe the target. Note that the lack of normalization factor in (4) allows to compute the objective functions PD and ETD up to t straightforward with Equations (7) and (8), and makes $b(\tau^t)$ no longer be a probability function.

We also consider three additional utility functions to determine the feasibility and optimality of the UAVs trajectories $s_{1:U}^{0:T}$ and sensor poses $s_{1:U,1:K}^{0:T}$ (which were obtained from $a_{1:U}^{0:T}$ and $a_{1:U,1:K}^{0:T}$). The additional objective, computed with Equation (7), accounts for the ability to reach the unobserved unnormalized belief in $b(\tau^t)$ from the last UAV location s_u^t and sensor pose $s^{t_u,k}$ within each sub-trajectory, in order to reduce the *myopic effect* of RHC. The constraints, computed with Equations (8) and (9), are the number of collisions among the UAVs and the number of overflights of restricted areas (Non Flying Zones, NFZ).

Finally, note that Equations (1)-(6) are computed cyclicallly (using the periods stated in the fourth column of Table 1), PD and ETD updated after the assimilation steps of each sensor, MYO computed for the final UAV location and sensor pose of each subtrajectory, and NFZs and COL obtained after simulating each trajectory.

Table 1: Target search equations.

Element	Desc.	Expression	Period	Eq. #
UAV	Deterministic Model	$s_u^t = f(s_u^{t-T_u}, a_u^{t-T_u}, \varepsilon^{t-T_u}, T_u)$	T_u	(1)
	Control Reference	$a_u^t = \{\theta_u^t, v_u^t, h_u^t\}$	T_{a_u}	(2)
Sensor	Deterministic Model	$s_{u,k}^t = g(s_{u,k}^{t-T_{u,k}}, a_{u,k}^{t-T_{u,k}}, s_u^t, T_{u,k})$	$T_{u,k}$	(3)
	Control Reference	$a_{u,k}^t = \{\alpha_{u,t}^c, \eta_{u,k}^t\}$	$T_{a_{u,k}}$	(4)
	Assimilation step	$b(\tau^t) \leftarrow (1 - p(D\|\tau^t, s_u^t, s_{u,k}^t)) \cdot b(\tau^t)$	$T_{k,m}$	(5)
Target	Prediction step	$b(\tau^t) \leftarrow \sum_{\tau^{t-T} \in G_\Omega} b(\tau^t \| \tau^{t-T_\tau}) p(\tau^{t-T_\tau})$	T_τ	(6)
Utility Functions	Pareto Objectives	$\mathrm{PD}(t) = 1 - \sum_{\tau^t \in G_\Omega} b(\tau^t)$	-	(7)
		$\mathrm{ETD}(t) = \sum_{k=0}^{t} (1 - \mathrm{PD}(k))$	-	(8)
		$\mathrm{MYO}(t) = \sum_{\tau^t \in G_\Omega} \prod_{u=1,k=1}^{U,K_u} h(\tau^t, s_u^t, s_{u,k}^t) b(\tau^t)$	-	(7)
	Constraints	$\mathrm{NFZs} = \sum_{t=1}^{T} \sum_{u=1}^{U} WithinNFZ(s_u^t)$	-	(8)
		$\mathrm{COL} = \sum_{t=1}^{T} \sum_{i=1}^{U} \sum_{j=i+1}^{U} Collision(s_i^t, s_j^t, d_{COL})$	-	(9)

2.2 Optimization of the Minimum Time S&R Problem

The complexity and multiple utility functions of the problem are tackled in this paper with multiple RHCs supported by different MOEAs distributed within an IM. In the following, we will revise their key concepts.

The selected MOEAs employ the Two-Stage Dominance (TSD) definition that was used in [19]: a solution ρ dominates another ρ' when 1) the sum of the constraint outcomes for ρ are smaller to that of ρ', and 2) when these sums are equal, when at least one of the objective functions of ρ is better for that objective function in ρ', without worsening any of the other objective functions. This ensures 1) that feasible solutions are preferred to unfeasible ones, and 2) the identification of high-quality solutions regarding all objectives.

Each MOEA operation is governed by a RHC, which divides the total mission time $[t^0, t^{end}]$ into Q time intervals, each corresponding to a decision horizon T_{DH}. For each time interval q, the lists of references for the UAVs $a_{1:U}^{((q-1)\cdot T_{DH}):(q\cdot T_{DH})}$ and for the sensors $a_{1:U,1:K}^{((q-1)\cdot T_{DH}):(q\cdot T_{DH})}$ are optimized until a stop criteria is met. The final UAVs locations, sensors poses and unnormalized belief of interval q, become the corresponding starting elements of the next. The selected starting elements of $q+1$ in the previous version of the framework [19] correspond to a *unique solution* selected within the first Pareto front of the last MOEA iteration for interval q. This selection relies on a function that prioritizes the Pareto objectives in the best order (which the optimization of different scenarios indicate that is MYO, ETD, PD) to reduce the myopic effects of the RHC for this problem. This strategy, also used previously in related works [12, 13], is straightforward to implement but eliminates the diversity of solutions coming from interval q into interval $q+1$.

The benefits of the IM rely on a *migration* process where each optimization nodes (commonly known as islands) sends a subset Ψ of solutions from its current population Φ to its neighbors, in order to increase the genetic diversity and prevent the premature convergence to local optima within each island. The following attributes are relevant to set it up. First, the IM *topology*, usually defined as a graph, shapes the migration flow of solutions among the islands. Key factors that can affect the IM performance are the number of islands and of communication channels, the connectivity degree, and the average lenght of the shortest path between any two islands. Second, the *migration frequency* μ determines how often the migrations from an island occur. It can be fixed (e.g. defined by the number of generations required) or variable (e.g. triggered when stagnation in island diversity is detected). It impacts the IM performance: if μ is too high, the islands may fail to converge between migration events; if μ is too low, the MOEA on each island will work as if it was isolated. Third, the *migration mode* refers to how migrations among the nodes take place: synchronously (all islands migrate at the same time) or asynchronously (each island performs migration independently). The

mode impacts the process's speed and scalability: asynchronous migration is more scalable and efficient; while synchronous migration allows precise control over migration events and is easier to analyze. Fourth, the *migration policy* defines the criteria for *selecting* the subset Ψ of emigrants from the current population Φ on the source island and for *replacing* solutions in its neighbors after having received Ψ.

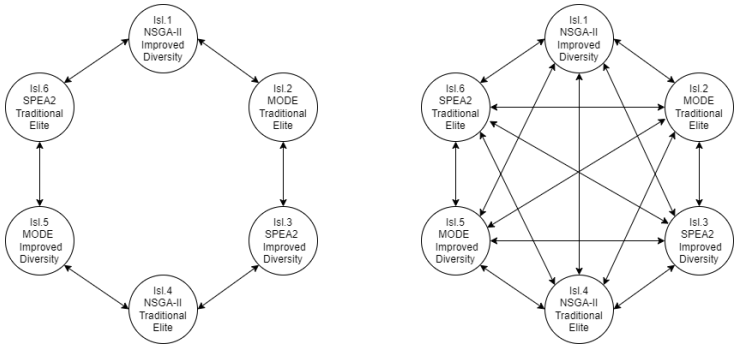
3 MINIMUM TIME SEARCH AND RESCUE FRAMEWORK

This section details the framework improvements described in this paper and their specification within DEVS. The notation used for the latter, not included in the paper for lack of space, is the usual of DEVS.

3.1 Improvements Description

The framework currently supports the following MOEAs: NSGA-II [27], SPEA-II [28], NSPSO [29] and MODE [30]. They have been selected for its good performance in other UAV trajectory planning problems and from their intrinsic differences. In short, NSGA-II and SPEA-II share the operators that create new solutions (parent selection, crossover and mutation), but differ in the function that decides which are the survival solutions from the old and new operation. NSPSO creates new solutions using information of the local and global best of each particle (current solution). Finally, MODE generates them from the crossover of each possible solution with a new one obtained from the values of three solutions of the population.

The RHC is enhanced to increment the diversity of the population across sequences. In particular, the new population Φ_0 transitioning from sequence q to $q+1$ is built using the entire Pareto front of the last iteration of sequence q instead of a unique solution within it. To implement it, the MOEA has been adapted to handle multiple problem definitions during the optimization of any sequence (associated to each of the final states of the best Pareto front of last sequence), instead of the single best solution. These definitions will differ, among other aspects, in the initial unnormalized target belief ${}^{q}b(\tau)^0$, the initial states of the UAVs ${}^{q}s_u^0$ and the sensors ${}^{q}s_{u,k}^0$, and the lists of actions ${}^{q}a_u^{1:J}$ and ${}^{q}a_{u,k}^{1:J}$ used in the previous sequences to reach them.



(a) Ring layout. (b) Full-connection layout.

Figure 1: IM experiments configurations.

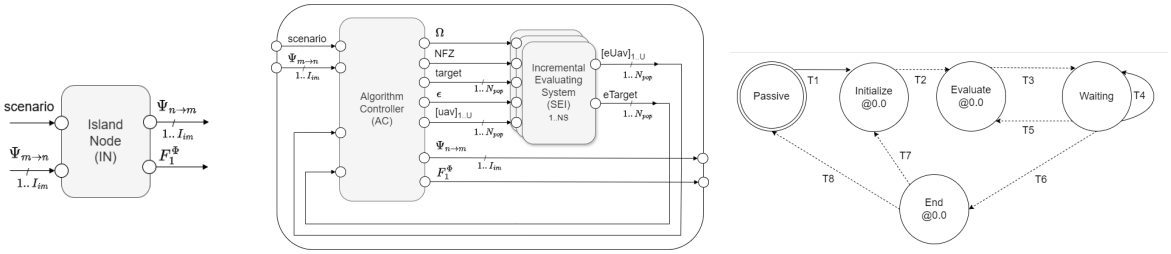
The IM is integrated to enhance the diversity and improve the quality of the solutions. Two different *topologies* are supported: *ring* and *full-connection*. While the ring, exemplified in Figure 1a, is the easiest to implement at the expense of limiting the flow of information through the nodes; the full-connection, exemplified in Figure 1b, allows a highest degree of information exchange at the expenses of higher communication requirements and the risk of a rapid convergence of all islands towards a local optimum. The migration *frequency* is fixed and equal in all the islands, making the migration *mode* synchronous. Regarding the *migration policy*, the *selection* of emigrants can be *elitist* (making the emigrant set equal to the first Pareto front) or *diverse* (selecting the N_{ex} solutions with biggest crowding distance); while *replacement* with

immigrants is also *elitist* (the solutions Ψ from the source island are inserted into the population Φ of the receiving island, and the combined population is reduced to N_{pop} solutions, using TSD and SPEA-II survival criteria for SPEA-II, and the NSGA-II one for NSGA-II, NSPSO and MODE).

3.2 DEVS Model Specification

The complete specification of the different models of the previous version of the framework are presented in [19]. The current version includes a novel coupled `Island Node(IN)`, which extends the `Optimizer(OP)` model presented in [19] by the inclusion of the required functionality to implement the improvements presented in Section 3.1.

IN specification and structure are illustrated in Figure 2a and Figure 2b. The specification introduces the subset $\Psi_{n\to m}$ (emigrant individuals) as an external output and $\Psi_{m\to n}$ (immigrant individuals) as an external input. The number of inputs and outputs depends on the island topology (being the number of nodes minus 1 in the full connection, and 1 input and 1 output in the ring). Additionally, the `scenario` input is modified to include the migration period μ and the chosen migration selection policy (elitist or diverse). Internally, these external inputs and outputs are integrated with the atomic model `Algorithm Controller (AC)`, responsible for the optimization, including the RHC and MOEA iterations.



(a) IN Specification. (b) IN Internal Structure. (c) AC model states.

Figure 2: Island Node model.

The behavior of the AC model, illustrated in Figure 2c (where dotted arrows represent δ_{int} while continuous ones represent δ_{ext}), proceeds as follows. In the passive state, the model remains inactive until it receives the `scenario` input, configuring the MOEA with the optimizer data and transitioning to the initialization state via external transition T1. In this state, the initial sequence q^0 is prepared by creating the population Φ of N_{pop} candidate solutions of the reference decision variables required to generate the UAVs trajectories and sensor poses. Afterwards, the model transitions to the evaluation state through the internal transition T2. Here, the MOEA iterates, generating solutions, and T3 is programmed to output via λ each solution in Φ to its Integrated Evaluation System (SEI) coupled model, where it will be simulated and evaluated. In the waiting state, the model pauses until all the solutions are received (transition T4), encapsulating results into their respective solutions. After receiving all the results, MOEA stopping criterion is checked. If unmet, T5 loops back to the evaluation state for another MOEA iteration. If met, T6 moves the model to the final state, where it determines whether to initiate a new RHC sequence $q+1$ or conclude (when $q=Q$). If further sequences are required, T7 returns the model to the initialization state; otherwise, T8 returns the model to the passive state and outputs the first Pareto front F_1^{Φ} via the λ function. The IN model introduces specific adjustments to this behavior: 1) in the evaluation state, during T3, the model checks whether the current MOEA iteration is a multiple of the inverse of the migration frequency μ. If so, the subset $\Psi_{n\to m}$ is generated according to the migration selection policy and outputted via the λ function to neighboring islands; and 2) in the waiting state, if a subset $\Psi_{m\to n}$ is received from a neighboring island via T4, it is integrated into the current population Φ with the replacement policy.

Our framework has been implemented using xDEVS engine [18], where system simulation is governed by a single supervisory element from the `Coordinator` class, resulting in a hard synchronization among all optimization nodes. This happens because all candidate solutions $\rho \in \Phi$ generated by the nodes share the same mission start and end times. As the SEI system aligns simulation time with mission time to replicate the real-world sequence of events, the DEVS event queuing mechanism prevents nodes to progress to the next iteration until all of them have completed their current evaluations. Although this imposes a significant speedup limitation in parallel or distributed architectures, it simplifies the deployment of the framework in these environments, as the coordinator transparently handles synchronization issues.

4 RESULTS

This section details the scenarios and optimization setup; and analyzes the performance of the different MOEAs, of the enhanced RHC migration, and of the IM.

4.1 Scenarios and Optimization Setup

The results are analyzed over three maritime Scenarios (S#), where a 10m-long drifting boat has to be found. The operations are conducted using three types of fixed-wing UAVs, whose properties and sensors are detailed in Table 2. Since the only decision variables for the UAVs are their orientations θ_u^t , their speed and altitude are fixed. The radar is static. The camera azimuth $\alpha_{a_{u,k}}$ or elevation $\eta_{a_{u,k}}^t$ is fixed when not stated. The main characteristics of each scenario are summarized in Table 3 and their initial beliefs $b(\tau^0)$ illustrated as the colored-map surface in Figure 3. The following aspects should also be noted: 1) S1 and S2 include the NFZ shown in black; 2) the target motion in S1 and S3 simulates the effects of wind and tides; and 3) in the illustrations, the initial UAV positions relative to $b(\tau^0)$ are shown at a fixed height (which is not representative of the UAV altitude h_u^t or arrival time at Ω) on the z-axis.

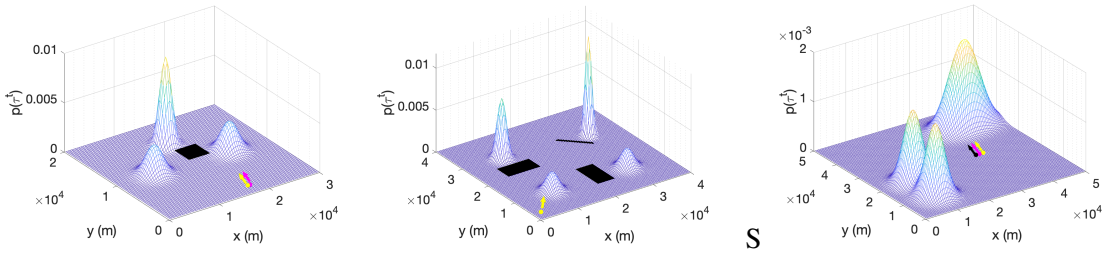
Table 2: UAVs & onboard sensors configuration

UAV				Sensors				
Type	Motion			Onboard	Motion			Detection
	T_u	Decision Variable	T_{a_u}		$T_{u,k}$	Decision Variable	$T_{a_{u,k}}$	$T_{k,m}$
1	1s	$\theta_u^t \in [-180°, 180°]$	10s	1 Radar	N/A			3s
2	0.5s	$\theta_u^t \in [-180°, 180°]$	20s	1 Camera	1s	$\alpha_{a_{u,k}} \in [-70°, 70°]$	6s	3s
				1 Radar	N/A			1.5s
3	1s	$\theta_u^t \in [-180°, 180°]$	20s	1 Camera	4s	$\eta_{a_{u,k}}^t = [20°, 40°]$	30s	4s
				1 Camera	4s	$\alpha_{a_{u,k}} = [-90°, 90°]$	6s	

Table 3: Search scenarios characteristics

Scenario	Search area			Target		UAVs		
	Ω (km^2)	$G_\Omega = N_x \times N_y$	$w_x \times w_y$ (m^2)	Moving	T_τ	Number & Type	t_u^0	t_u^{end}
S1	30x20	80x80	750x750	✔	75s	2 x Type 1	0s	900s
S2	40x40	100x100	400x400		N/A	1 x Type 2	0s	1500s
S3	50x50	100x100	500x500	✔	144s	2 x Type 3	0s	1800s
						1 x Type 2	700s	1800s

For optimization purposes, the total available flight time is divided in $Q = 4$ intervals of $T_{DH} = 225$s for S1, $Q = 6$ of $T_{DH} = 250$s for S2, and $Q = 8$ of $T_{DH} = 225$s for S3. Each interval is optimized by each MOEA manipulating 50 solutions during 100 generations. The specific parameters of each MOEA are the following. NSGA-II and SPEA2 use binary tournament selection, single-point crossover with crossing probability $p_c = 0.8$, and a two level additive gaussian mutation with $\sigma_h = 0.1$ and $\sigma_l = 0.01$. NSPSO acceleration coefficients $c_1 = 1.5$ and $c_2 = 1.5$, inertia coefficient $w = 0.7$, and constriction coefficient to



(a) Scenario S1. (b) Scenario S2. (c) Scenario S3.

Figure 3: Scenarios illustration. NFZs are shown in black.

$\chi = 2$. For MODE, the base vector is uniformly randomly selected, the mutation values are $\sigma_l = 0.1$ and $\sigma_h = 0.01$, and the recombination is performed with a probability of $p_r = 0.8$ with single-point crossover.

For the IM, we use the two topologies specified in Figure 1, which combine two search strategies: one group of nodes prioritizes population diversity, integrating an *enhanced* RHC internal migration with a *diverse* island emigration, while the other focuses on solution quality, combining a *traditional* RHC internal migration with an island *elitist* emigration. In all the cases, we also use a migration frequency $\mu = 1/25$ (this makes 3 IM migrations within each RHC sequence), and the selection policy can be either elitist and diverse.

4.2 Independent MOEAs Analysis

Given the stochastic nature of the MOEAs, their results are statistically characterized. For this reason, the MOEAs configuration under analysis are executed 20 times (Number of Runs, $NR = 20$). With their results, we analyze their ability to generate feasible trajectories, recording the number of runs that fail in Table 4. Then, we assess the performance of each MOEA independently using two types of graphs, considering in both cases only the best solution ρ^{best} (according to the ordering MYO, ETD and PD) of the first Pareto front F_1^{Φ} for each generation. The *evolution graphs*, shown in the first and second row of Figure 4, illustrate the progress, through the generations, of the mean (line) and standard deviation (envelope) of the $NR = 20$ values of ETD and of PD for $\rho^{best} \in F_1^{\Phi}$. The objective MYO is excluded since it is designed to mitigate myopic effects and inherently influences both ETD and PD. The *dominance graphs*, in the third row of Figure 4, depict (again through the generations) when the mean of the $NR = 20$ values of MYO, of ETD and of PD values of $\rho^{best} \in F_1^{\Phi}$ of each MOEA dominate the mean of the $NR = 20$ values of MYO, of ETD and of PD for $\rho^{best} \in F_1^{\Phi}$ of the others. In these graphics, which help to determine which MOEA is better suited for each scenario, green identifies the MOEA dominating all the others, grey MOEAS that are both dominated and dominant, and red MOEAs dominated by all the others. Last, the fourth row of graphics show a representative solution of each scenario, representing with lines of different colors the trajectory of the different UAVs involved in the mission and with the colored-map surface the updated unnormalized belief. Besides, on the top of these last graphics, we show the ETD and PD values achieved at the end of the trajectories for the corresponding simulation.

Analyzing trajectory feasibility, we found that all MOEAs, whether using traditional or enhanced internal migration among RHC intervals, successfully satisfy the COL constraint in all three scenarios (COL is omitted for S2 in Table 4 since it is inherently met with a single UAV). However, scenarios with NFZs (S1 and S2) are more challenging, occasionally resulting in non-feasible trajectories with the traditional internal migration. This happens because the MYO function eventually guides a UAV too close to an NFZ by the end of sequence q, prioritizing the shortest path to areas with high $b(\tau^t)$ accumulation. As a result, selecting a single solution $\rho^{best} \in {}^{q}F_1^{\Phi}$ (following MYO, ETD, and PD) to generate the solutions for sequence $q+1$ may

lead to NFZ runs and non-feasible trajectories. The enhanced internal migration eliminates this problem by leveraging the entire set ${}^{q}F_1^{\Phi}$, and allowing to start from other solutions that can avoid NFZs.

Table 4: Number of runs with non-feasible trajectories.

MOEA	Internal Migration	S1		S2	S3
		COL	NFZ	NFZ	COL
NSGA-II	Traditional	0	1	4	0
	Enhanced	0	0	0	0
SPEA2	Traditional	0	0	1	0
	Enhanced	0	0	0	0
NPSO	Traditional	0	5	1	0
	Enhanced	0	0	0	0
MODE	Traditional	0	2	0	0
	Enhanced	0	0	0	0

The ETD and PD *evolution graphs* summarize the ability of each MOEA (with traditional internal migration) to improve the objective functions. Their stepped behavior is due to the RHC approach, since it occurs whenever a new segment is being optimized, letting the sensors onboard the UAV increase the chances to detect (and therefore to reduce the expected time for detecting) the target. Readers can observe that the four algorithms perform similarly for S2. However, for S1 and S3, MODE (in purple) outperforms the others, while NSPSO (in yellow) consistently yields the worst results. This trend is confirmed by the *dominance graphs*, where MODE appears in green for S1 and S3 in most generations, NSPSO is predominantly red for the same scenarios, and NSGA-II (in blue) and SPEA-II (in red), exhibit stable and similar performance for all the scenarios (becoming the best or intermediate choice for S2).

The previous results are complemented with the information provided by Figure 5, which presents pair to pair dominance graphs comparing each MOEA with the traditional and enhanced internal migration (respectively labeled as A and B). These graphs show that there is not a consistent advantage in any of internal migrations regarding the objective function. Therefore, while the enhanced internal migration proposed in this paper favors the feasibility of the solutions, it may not be the best option to optimize a specific objective order (in particular MYO, ETD and PD). Additionally, as the number of objectives increases, may ultimately degrade the prioritization of this order across RHC sequences compared to the traditional approach.

Finally, the representative solutions show how the UAVs flights let their onboard sensors collect a good percentage of the initial belief. It is worth noting that in S1 and S2, the 'z' scale, related to the remaining unnormalized belief, is different for the graphics in Figure 3 and the fourth row of Figure 4. Finally, the complexity of the UAVs trajectory is incremented from S1 to S2 and from S2 to S3, not only for the difficulty of each scenario, but also for the increment of the mission time.

4.3 IM Performance Analysis

For the IM analysis, we focus on scenarios S1 and S2, as they exhibit myopic effects not affecting S3. Note that NSPSO was excluded from the topology due to its poor performance in the previous analysis, while the remaining MOEAs, with both internal migration mechanisms, are distributed across the nodes of the IM.

The dominance graphs in Figure 6 and the information in Table 5 (where the canonical reference corresponds to the best-performing independent MOEA in each scenario) shows that, at the end of the optimization (i.e. when the last sequence of the RHC has been optimized) for each scenario, both IM configuration outperforms its corresponding canonical MOEA. While the IM initially performs similarly to the canonical MOEAs, it steadily improves solution quality once the migration process begins, maintaining this improvement to the end of the optimization. Consequently, the IM is not usually dominated by any canonical

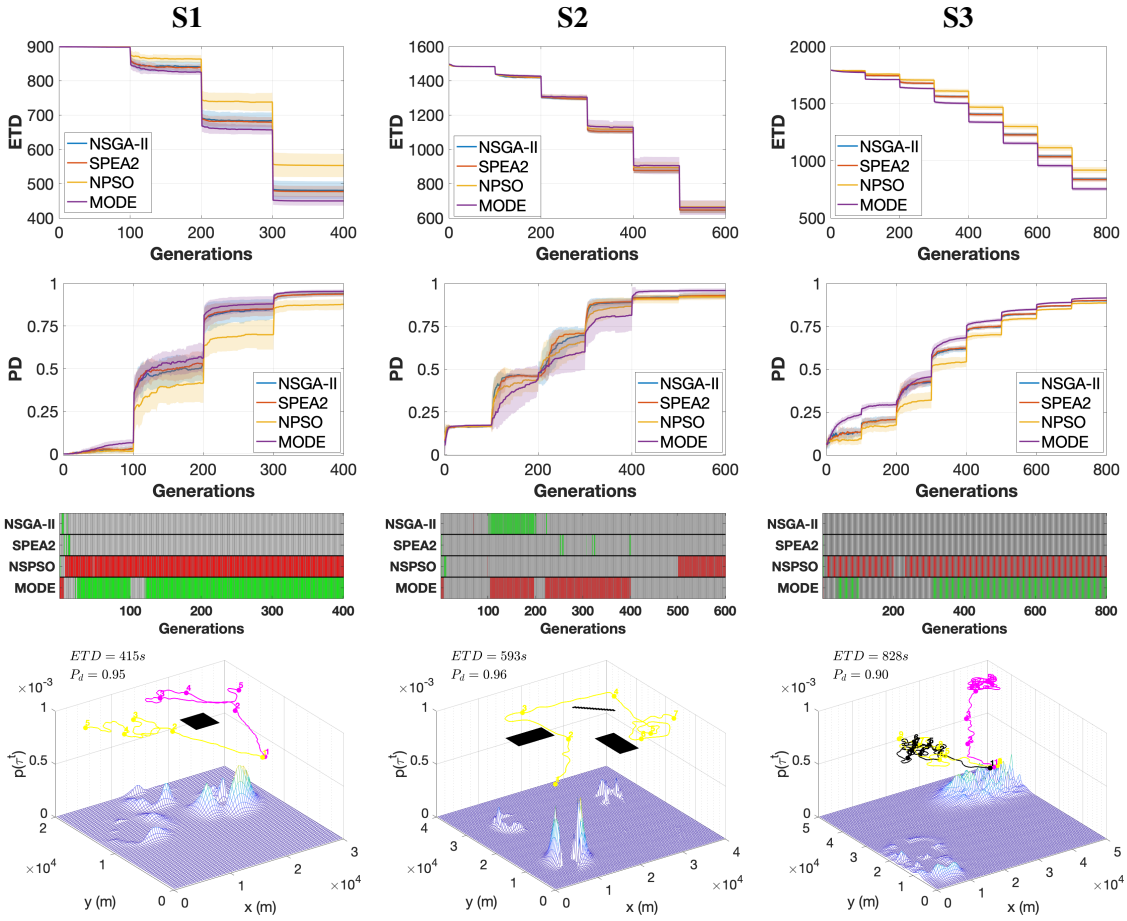


Figure 4: Objective and dominance graphs plus representative solutions for MOEAs running independent.

execution during any generation, and in most cases (particularly in the final sequences), it dominates every other algorithm. Finally, the combination of the two search strategies in the optimization nodes prevents the IM from producing non-feasible runs.

Table 5: IM Numerical results after $NR = 20$ executions.

Scenario	Objective	Canonical	Ring	Full-connection
S1	ETD	439s	427s	424s
	PD	0,95	0,956	0,956
S2	ETD	644s	599s	597s
	PD	0,966	0,97	0,968

4.4 Computational Cost Analysis

The improvements achieved by the IM come at a computational cost associated with the inter-island solution migration and the computations requirements of the involved execution nodes. To analyze the computations requirements of the different approaches, Table 6 presents the average WCT (Wall-Clock Time) of the canonical and of the two IM topologies. The times are obtained in a Intel Core i9 (10th generation model 10-900x) with 10 cores each at 4.5 GHz, 20 MB cache, and 48 GiB of DDR4 RAM at 2400 MHz. The table also quantifies the Slowdown of the IM compared to the canonical (single-node) execution, and the overhead obtained dividing the slowdown by $I_{im} = 6$, and associated with the migration operations and node

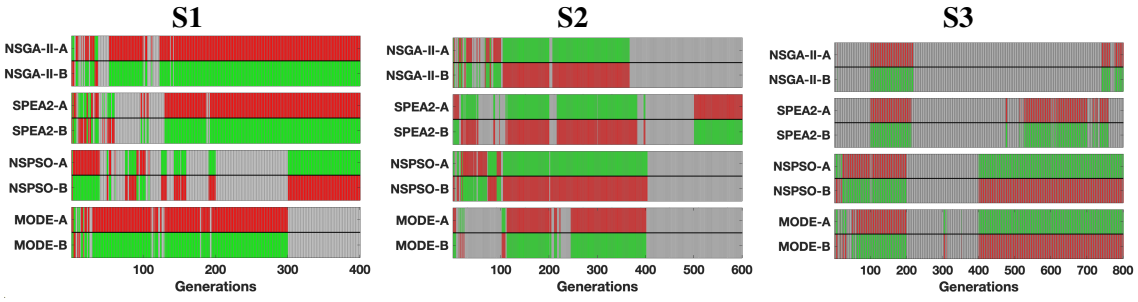


Figure 5: Pairwise dominance graphs comparing RHC internal migration approaches.

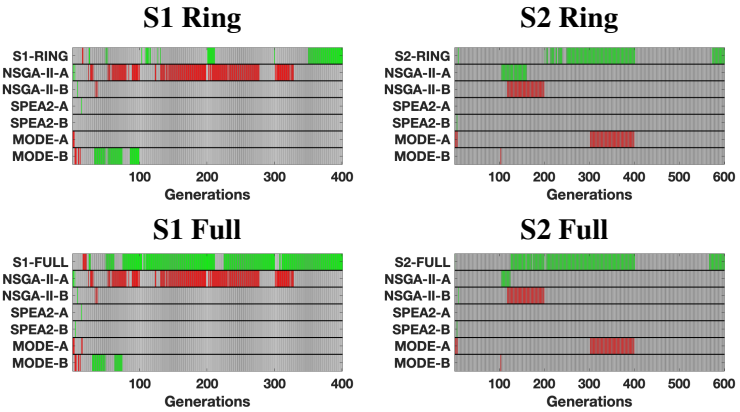


Figure 6: Dominance graphs comparing IM against indepent runs of MOEAs in the inland.

communication. The results indicate that the IM is between 6.92 and 7.27 times slower than the canonical node, when the complete framework is run in a single computer. However, when considering the workload distribution across $I_{im} = 6$ optimization nodes, the additional overhead due to migration operations only ranges between 1.15 and 1.21 times.

Table 6: Computational Cost Analysis.

Scenario	Parameter	Canonical	Ring	Full-connection
S1	WCT	265s	1834s	1926s
	Slowdown	1.00	6.92	7.27
	Overhead	1.00	1.15	1.21
S2	WCT	507s	3542s	3541s
	Slowdown	1.00	6.99	6.98
	Overhead	1.00	1.16	1.16

5 CONCLUSIONS AND FUTURE WORK

This paper presents a new version of our DEVS-based UAV path planner framework to optimize target-search problem, which incorporates multiple MOEAs, an enhanced internal migration between RHC sequences, and a PEA implemented as an Island Model (IM). The new algorithms outperform the previous NSGA-II only-support in certain scenarios. The possibility of passing the complete first Pareto front between sequences of the RHC brings diversity to the solutions and favors the final feasibility of the solutions.

The IM enables the simultaneous execution of different MOEAS that exchange solutions through migration operations. The results confirm the benefits of the different enhancements, showing that as a whole, the

combination of nodes focused on preserving population diversity with those enhancing solution quality, consistently outperforming the independent execution of the single MOEA. In short, this framework mitigates the myopic effect of non-feasible trajectories and improves the results across all optimization objectives.

Given the WCT required to execute the whole IM in a single computer, we are planning to leverage xDEVS Cloud computing capabilities to accelerate its execution, as we did for the previous version of the framework in [19]. We also intend to introduce in the IM a variable migration frequency controlled by a metric that measures population diversity, and to support the optimization of trajectories with acyclic reference signals within the framework.

ACKNOWLEDGMENTS

This work has been supported by the Research Projects IA-GESBLOOM-CM (Y2020/TCS-6420) funded by the Synergic program of the Comunidad Autónoma de Madrid (CAM), INSERTION (PID2021-127648OB-C33) by the Knowledge Generation program of the Spanish Ministry of Science and Innovation, and SMART-BLOOMS (TED2021-130123B-I00) by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

REFERENCES

[1] B. O. Koopman, "Search and screening, operations evaluation group report 56," *Center for Naval Analysis, Alexandria, Virginia*, 1946.

[2] L. D. Stone, *Theory of optimal search*. Elsevier, 1976.

[3] L. D. Stone, J. O. Royset, A. R. Washburn *et al.*, "Optimal search for moving targets," 2016.

[4] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote Sensing*, vol. 4, 2012.

[5] M. Raap, M. Preuß, and S. Meyer-Nieberg, "Moving target search optimization – a literature review," *Computers & Operations Research*, vol. 105, 2019.

[6] J. N. Eagle, "The optimal search for a moving target when the search path is constrained," *Operations Research*, vol. 32, 1984.

[7] A. A. Feldbaum, "Dual control theory. i," *Avtomatika i Telemekhanika*, vol. 21, 1960.

[8] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, "Information-theoretic coordinated control of multiple sensor platforms," in *IEEE Int. Conf. on Robotics and Automation*, vol. 1, 2003.

[9] Y. Yang, A. Minai, and M. Polycarpou, "Decentralized cooperative search in UAV's using opportunistic learning," in *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, 2002.

[10] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a bayesian world," *Field and Service Robotics: Recent Advances in Reserch and Applications*, 2006.

[11] P. Lanillos, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. De la Cruz, "Minimum time search in uncertain dynamic domains with complex sensorial platforms," *Sensors*, vol. 14, 2014.

[12] S. Perez-Carabaza, J. Bermudez-Ortega, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "A multi-uav minimum time search planner based on aco r," in *Genetic and Evolutionary Computation Conf.*, 2017.

[13] S. Pérez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and G. Pajares, "Minimum time search in real-world scenarios using multiple UAVs with onboard orientable cameras," *Journal of Sensors*, vol. 2019, 2019.

[14] K. Trummel and J. Weisinger, "The complexity of the optimal searcher path problem," *Operations Research*, vol. 34, 1986.

[15] J. B. Bordón-Ruiz, E. Besada-Portas, J. A. López-Orozco, and J. L. Risco-Martín, "DEVS-based simulation for search and rescue missions involving multiple uavs," in *Annual Modeling and Simulation Conf.*, 2021.

[16] J. B. Bordón-Ruiz, E. Besada-Portas, J. L. Risco-Martín, and J. A. López-Orozco, "DEVS-based evaluation of uavs-based target-search strategies in realistically-modeled missions," in *ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation*, 2021.

[17] B. P. Zeigler, A. Muzy, and E. Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018.

[18] J. L. Risco-Martín, S. Mittal, K. Henares, R. Cardenas, and P. Arroba, "xDEVS: A toolkit for interoperable modeling and simulation of formal discrete event systems," *Software: Practice and Experience*, 2022.

[19] J. Bordón-Ruiz, E. Besada-Portas, and J. A. López-Orozco, "Cloud DEVS-based computation of uavs trajectories for search and rescue missions," *Journal of Simulation*, vol. 16, 2022.

[20] R. Tanese, "Parallel genetic algorithm for a hypercube," in *Int. Conf. on Genetic Algorithms*, 1987.

[21] T. C. Belding, "The distributed genetic algorithm revisited," in *Int. Conf. on Genetic Algorithms*, 1995.

[22] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 6, 2002.

[23] T. Harada and E. Alba, "Parallel genetic algorithms: A useful survey," *ACM Comput. Surv.*, vol. 53, 2020.

[24] D. Whitley, S. Rana, and R. B. Heckendorn, "The island model genetic algorithm: On separability, population size and convergence," *Journal of Computing and Information Technology*, vol. 7, 1999.

[25] R. Jarray, M. Al-Dhaifallah, H. Rezk, and S. Bouallègue, "Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles," *Sensors*, vol. 22, 2022.

[26] T. Ma, Y. Wang, and X. Li, "Convex combination multiple populations competitive swarm optimization for moving target search using uavs," *Information Sciences*, vol. 641, 2023.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, 2002.

[28] E. Zitzler, "SPEA2: Improving the strength pareto evolutionary algorithm," in *European Genetic Conf.*, 2001.

[29] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Genetic and Evolutionary Computation Conf.*, 2003.

[30] H. A. Abbass and R. Sarker, "The pareto differential evolution algorithm," *Int. Journal on Artificial Intelligence Tools*, vol. 11, 2002.

AUTHOR BIOGRAPHIES

JUAN B. BORDÓN-RUIZ is a Ph.D. student from UCM, working for a private firm of the aerospace sector. His research interest include modeling and simulation, control of autonomous vehicles and optimization of target-search problems. His email address is jubordon@ucm.es.

EVA BESADA-PORTAS received her Ph.D. from UCM, where she is an Associate Professor in the Department of Computer Architecture and Automation. Her research interests include uncertainty modeling and simulation, and optimal planning of unmanned vehicles. Her email address is ebesada@ucm.es.

JOSE LUIS RISCO-MARTIN received his Ph.D. from UCM, where he is currently Full Professor in the Department of Computer Architecture and Automation. His research interests include systems modeling, simulation, and optimization. His email address is jlrisco@ucm.es.

JOSÉ A. LOPEZ-OROZCO is a Full Professor in the University Complutense of Madrid. He holds a Ph.D. in Physics from the same University. His research interests include multisensor data fusion, control and planning of unmanned vehicles, and robotics. His email address is jalo@ucm.es.