



**HAL**  
open science

# Fuzzy Multi-Agent Simulation for Collective Energy Management of Autonomous Industrial Vehicle Fleets

Juliette Grosset, Ouzna Oukacha, Alain-Jérôme Fougères, Moïse Djoko-Kouam, Jean-Marie Bonnin

► **To cite this version:**

Juliette Grosset, Ouzna Oukacha, Alain-Jérôme Fougères, Moïse Djoko-Kouam, Jean-Marie Bonnin. Fuzzy Multi-Agent Simulation for Collective Energy Management of Autonomous Industrial Vehicle Fleets. *Algorithms*, 2024, 17 (11), pp.484. 10.3390/a17110484 . hal-04767307

**HAL Id: hal-04767307**

**<https://hal.science/hal-04767307v1>**

Submitted on 5 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Article

# Fuzzy Multi-Agent Simulation for Collective Energy Management of Autonomous Industrial Vehicle Fleets

Juliette Grosset <sup>1,\*</sup>, Ouzna Oukacha <sup>2,†</sup>, Alain-Jérôme Fougères <sup>2,†</sup>, Moïse Djoko-Kouam <sup>3</sup>  
and Jean-Marie Bonnin <sup>4</sup>

<sup>1</sup> ECAM Rennes and IMT Atlantique, IRISA, 35000 Rennes, France

<sup>2</sup> ECAM Rennes, 35000 Rennes, France; ouzna.oukacha@ecam-rennes.fr (O.O.); alain-jerome.fougeres@ecam-rennes.fr (A.-J.F.)

<sup>3</sup> ECAM Rennes and IETR, UMR CNRS 6164, CentraleSupélec, 35000 Rennes, France; moise.djoko-kouam@ecam-rennes.fr

<sup>4</sup> IMT Atlantique, IRISA, 35000 Rennes, France; jm.bonnin@imt-atlantique.fr

\* Correspondence: juliette.grosset@ecam-rennes.fr

† These authors contributed equally to this work.

**Abstract:** This paper presents a multi-agent simulation implemented in Python, using fuzzy logic to explore collective battery recharge management for autonomous industrial vehicles (AIVs) in an airport environment. This approach offers adaptability and resilience through a distributed system, taking into account variations in AIV battery capacity. Simulation scenarios were based on a proposed charging/discharging model for an AIV battery. The results highlight the effectiveness of adaptive fuzzy multi-agent models in optimizing charging strategies, improving operational efficiency, and reducing energy consumption. Dynamic factors such as workload variations and AIV-infrastructure communication are taken into account in the form of heuristics, underlining the importance of flexible and collaborative approaches in autonomous systems. In particular, an infrastructure capable of optimizing charging according to energy tariffs can significantly reduce consumption during peak hours, highlighting the importance of such strategies in dynamic environments. An optimal control model is established to improve the energy consumption of each AIV during its mission. The energy consumption depends on the speed, as demonstrated via numerical simulations using realistic data. The speed profile of each AIV is adjusted according to the various constraints within an airport. Overall, the study highlights the potential of incorporating adaptive fuzzy multi-agent models for AIV energy management to boost efficiency and sustainability in industrial operations.

**Keywords:** cooperative mobile robots; intelligent system architectures; fuzzy energy management; fuzzy multi-agent simulation; optimization control problem; airport 4.0



**Citation:** Grosset, J.; Oukacha, O.; Fougères, A.-J.; Djoko-Kouam, M.; Bonnin, J.-M. Fuzzy Multi-Agent Simulation for Collective Energy Management of Autonomous Industrial Vehicle Fleets. *Algorithms* **2024**, *17*, 484. <https://doi.org/10.3390/a17110484>

Academic Editor: Sergey Y. Yurish

Received: 12 August 2024

Revised: 21 October 2024

Accepted: 22 October 2024

Published: 28 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Industry 4.0 comes with a high degree of digitalization for industrial processes, as well as a significant increase in communication and cooperation between the machines that make it up. This is the case with Autonomous Industrial Vehicles (AIVs) and other cooperative mobile robots that are proliferating in factories or airports and whose intelligence and autonomy are increasing.

The deployment of AIV fleets raises several issues, all of which are related to their actual level of autonomy: acceptance by employees, vehicle localization, traffic flow, collision detection, and vehicle perception of changing environments. Simulation allows us to take into account the different constraints and requirements formulated by manufacturers and future users of these AIVs.

Before starting to test AIV traffic scenarios on a large scale in sometimes complex industrial or airport situations, it is essential to simulate these scenarios [1]. One significant

benefit of running simulations is that it provides usable results without the need to apply a scaling factor.

The main benefits of simulating AIV operations are extensively presented by Tsolakidis et al. [2]: simulation reduces the development time and cost of an AIV, minimizes the potential operational risks associated with the AIV, enables the feasibility of different AIVs scenarios to be assessed at a strategic or operational level, provides a rapid understanding of AIV operations (under conditions of limited data availability), and identifies improvements in facility layout configurations hosting AIVs.

Simulation also provides flexibility in terms of deployment and redeployment, and it enables us to study the sharing of responsibility between a central server and robots (local/global balance) for the various operational decisions. Another advantage of simulations is to introduce humans into scenarios in order to convince people, before the actual deployment of autonomous mobile robots, of the safe nature of the coexistence and possible interactions between these future mobile robots and human operators in industry [3].

Agent-based approaches are often proposed for the simulation of autonomous vehicles [4], including path planning in a large-scale contexts [5], or optimal task allocation with collision and obstacle avoidance [6].

In this article, we focus on collaborative strategies, as we believe the autonomy of each AIV can be significantly enhanced through collaboration with other AIVs and infrastructure. By sharing data, drivers can access information beyond their immediate perception, such as whether charging stations are in use, whether a task can be completed before going to recharge because there is a wait at the charging stations, etc. Collaboration and collective strategies can lead to more efficient and better coordinated behavior.

In order to achieve an efficient allocation of energy during each mission of an AIV, an algorithm based on distance is proposed, utilizing fundamental principles of acceleration and deceleration distances. This method systematically determines the optimal control strategy for minimizing the AIV energy consumption during its mission, based on the distance traveled. From this optimal control strategy, a speed profile is generated by connecting the maximum speeds at each node. Through the use of Pontryagin's Maximum Principle (PMP) [7], these maximum speed values are obtained.

Three strategies are assessed using this technique, each generating an energy consumption profile. The decision of which strategy to use during an AIV mission is made using a fuzzy logic model, which depends on two inputs: waiting baggage and traffic. This structured approach allows for more accurate and efficient path planning and control of the mobile robot, ensuring optimal performance while adhering to physical constraints.

In this article, we begin by presenting a state-of-the-art review of fuzzy agent-based simulations, energy control, and optimization models. In Section 3, we review the various elements of the fuzzy agent model before introducing our fuzzy decision model for battery recharging in Section 4. In Section 5, we present the case study and describe our simulation framework used to test different scenarios for the autonomous management of battery recharging for AIV. We conduct a comparative analysis between threshold-based approach and fuzzy logic models with three initial scenarios in Section 6. Section 7 presents the results of three heuristics simulated in three scenarios, incorporating more realistic constraints of an airport, such as the flow of baggage arrivals. To achieve a more realistic simulation framework, we further refine the AIV energy model. Therefore, in Section 8, we propose a methodology with an optimization model to calculate the energy cost for each AIV based on their speed throughout their mission in our case study. Section 9 describes in detail an algorithm for the energy consumed by each strategy and a fuzzy logic model to select the strategy adapted for each AIV. The different results are presented and discussed in Section 10. Finally, we conclude by highlighting the potential of incorporating adaptive fuzzy multi-agent models for AIV energy management to enhance efficiency and sustainability in industrial operations. We also present various perspectives for future work.

## 2. State of the Art

### 2.1. Fuzzy Agent-Based Simulation

To effectively enhance the autonomy of AIVs, it is important to explore a range of advanced decision-making techniques that help address the problem of energy management. These problems have been tackled using different techniques in the literature [8]. For example, ref. [9] provides a function for the energy minimization problem. They use meta-heuristic algorithms such as genetic algorithms and particle swarm optimization methods to create a more precise robot motion trajectory in their case, resulting in an energy-efficient robot configuration.

Our current research focuses on the use of fuzzy agents to manage the levels of imprecision and uncertainty involved in modeling the behavior of simulated vehicles [10]. Fuzzy set theory is well suited to the processing of uncertain or imprecise information that must lead to decision-making by autonomous agents [11]. The concept of the fuzzy agent can, therefore, be proposed as a partial implementation of this theory.

Most of the control tasks performed via autonomous mobile robots (perception, localization, mapping, path and task planning, navigation and motion control, obstacle avoidance, communication, and energy control [12]) have been the subject of performance improvement studies using fuzzy logic:

1. The navigation of mobile robots from conceptual, theoretical, or application points of view [13], the navigation of several mobile robots [14], the navigation and control of a mobile robot in an unknown environment in real time [15], and a comparison of the navigation performance of mobile robots obtained using fuzzy logic or neural networks [16];
2. Obstacle avoidance from conceptual and systemic points of view in an unknown dynamic environment [17];
3. Path planning strategies focusing on obstacle avoidance [18] or global navigation [19];
4. Motion planning [20];
5. The localization of mobile robots [21];
6. The intelligent management of energy consumption [22].

An agent-based system is fuzzy if its agents perform fuzzy behaviors or if the knowledge they use is fuzzy. This means that agents can have the following:

- Fuzzy knowledge (fuzzy decision rules, fuzzy linguistic variables, and fuzzy linguistic values);
- Fuzzy behaviors (the behaviors adopted by the agents as a result of fuzzy inferences);
- Fuzzy interactions, organizations, or roles [23].

Fuzzy agents can follow the evolution of fuzzy information that comes from their environment and from the agents [24]. By interpreting the fuzzy information they receive or perceive, fuzzy agents interact within a multi-agent system; they can also interact in a fuzzy manner. For example, a fuzzy agent can discriminate a fuzzy interaction value to evaluate its degree of affinity (or interest) with another fuzzy agent [25].

The different contributions of this article related to fuzzy logic are as follows:

- Propose a system designed from the literature, in particular [26], which aims to manage decisions regarding energy levels and tasks for a robot or autonomous agent.
- Conduct a comparative analysis between threshold-based approach and the fuzzy logic model adapted for our case-study scenario proposed in Section 6.
- Refine the AIV model for energy management to take into account more realistic constraints and the possibility of AIVs communicating with each other and with infrastructure elements such as charging stations.

### 2.2. Energy Control Models and Optimization Models

Reducing energy consumption has become a critical issue across all sectors. Optimization is an indispensable tool for saving energy, and the choice of the cost function is crucial. Various mathematical models are defined for mobile robots to address different objectives

of the problem. The power function is frequently used because it depends on simple parameters such as speed and motor force [27,28]. Both reducing and increasing speed have a significant impact on energy consumption, and variations in motor force also affect energy usage. In this paper, we are interested in an optimal control problem, and the speed profile generator is determined by the motor force, which depends on the acceleration (or deceleration) and the mass of mobile robots. Optimizing energy consumption involves finding an optimal control strategy modeled by the motor force.

Several techniques have been employed to tackle the optimization problem in mobile robots.

The authors [29] propose a genetic algorithm to reduce the energy consumption of a mobile robot. The cost function is the sum of the linear velocity squared and angular velocity squared. This algorithm relies on an optimal fuzzy logic controller, which enables the robot to track different paths (e.g., zigzags and sharp turns).

The fuzzy logic Mamdani method is employed in [30] to determine three outputs: curve speed, energy consumption, and time information. Through the use of optional curved path data (radius, center angle, load, and ground friction), three fuzzy rules are defined to identify these outputs. The total energy consumed is the sum of the power sensor, control, and motion. A trapezoidal speed profile is utilized to create an optimal profile based on fixed times or distances for straight paths, involving three segments: acceleration, constant speed, and deceleration. Alternatively, a triangular speed profile can be used, requiring only acceleration and deceleration phases. Minimum energy consumption is achieved from these speed profiles for straight paths. For curved paths, the speed profile is derived from fuzzy logic, incorporating the maximum and minimum values of the trapezoidal speed profile. The integration of these methods highlights the importance of optimizing speed and motor force to minimize energy consumption in mobile robots.

In the railway field, the most commonly used technique is based on braking distance to calculate a speed profile that respects the maximum speed across the path. In the paper by Tan et al. [31], the problem of train speed trajectory optimization is treated as an optimal control problem. The cost function represents the net electrical energy, which depends on the maximum electrical traction and braking forces, as well as the efficiency of electric motors during traction and braking operations. The authors propose a numerical algorithm based on PMP to find the optimal speed trajectory. An in-depth study using this principle identified conditions for the traction and braking phases. Based on the initial and final train speeds on a segment, and through the use of a numerical algorithm based on distance, the traction and braking values at each stage converge towards cruising speed. Various comparisons with other numerical optimization methods demonstrate the effectiveness of this approach.

To address an optimal control problem for a mobile robot operating in an airport environment, the goal is to solve an energy optimization problem specific to this setting. The cost function is given by the integral of the power delivered via the robot's motor, which provides the absolute work. This refinement of the cost function offers a more accurate reflection of the energy consumption during the AIV mission. Additionally, it enables the AIV to perform three types of movements simultaneously: acceleration, deceleration, and constant speed.

The different contributions of this article related to energy control and optimization models are as follows:

- Define an energy model for each AIV using an optimal control problem.
- Create a database for the optimal speed of an AIV between two nodes by solving the optimal control problem.
- Propose an offline/online algorithm for energy consumption that calculates the speed profile and energy consumed for each AIV mission.
- Present three strategies for speed profiles based on two parameters: waiting baggage and traffic circulation.

- Develop a fuzzy logic model to select the appropriate strategy for each AIV during its mission.

### 3. Fuzzy Agent-Based Model

An agent-based system is fuzzy if its agents have fuzzy behaviors or if the knowledge they use is fuzzy [23]. This means that agents can have the following:

1. Fuzzy knowledge (fuzzy decision rules, fuzzy linguistic variables, and fuzzy linguistic values);
2. Fuzzy behaviors (the behaviors adopted by agents because of fuzzy inferences);
3. Fuzzy interactions, organizations, or roles.

The different elements of the fuzzy agent model are as follows [10]:

- The agent-based fuzzy system (Section 3.1);
- The behavior of a fuzzy agent, inspired by perceive–decide–act feedback loops [32] (Section 3.2);
- The behavioral functions of a fuzzy agent (Section 3.3);
- The fuzzy interactions between two fuzzy agents (Section 3.4).

#### 3.1. Agent-Based Fuzzy System

$$\tilde{M}_\alpha = \langle \tilde{A}, \tilde{I}, \tilde{P}, \tilde{O} \rangle \tag{1}$$

In Equation (1), the following applies:  $A$  is a set of agents,  $A = \{\alpha_1, \dots, \alpha_n\}$ ;  $\tilde{A}$  is a set of fuzzy agents,  $\tilde{A} = \{\tilde{\alpha}_1, \dots, \tilde{\alpha}_m\}$  with  $\tilde{A} \subseteq A$ ;  $\tilde{I}$  is a set of fuzzy interactions between fuzzy agents;  $\tilde{P}$  is a set of fuzzy roles filled by fuzzy agents; and  $\tilde{O}$  is a set of fuzzy organizations defined for fuzzy agents (subsets of strongly related fuzzy agents).

#### 3.2. Behavior of a Fuzzy Agent

$$\tilde{\alpha}_i = \langle \Phi_{\Pi(\tilde{\alpha}_i)}, \Phi_{\Delta(\tilde{\alpha}_i)}, \Phi_{\Gamma(\tilde{\alpha}_i)}, K_{\tilde{\alpha}_i} \rangle \tag{2}$$

In Equation (2), for a fuzzy agent  $\tilde{\alpha}_i$ ,  $\Phi_{\Pi(\tilde{\alpha}_i)}$  is its observation function,  $\Phi_{\Delta(\tilde{\alpha}_i)}$  is its decision-making function,  $\Phi_{\Gamma(\tilde{\alpha}_i)}$  is its action function, and  $K_{\tilde{\alpha}_i}$  is its knowledge base.

#### 3.3. Behavioral Functions of a Fuzzy Agent

$$\Phi_{(I_{\tilde{\alpha}_i})} : (E_{\tilde{\alpha}_i} \cup I_{\tilde{\alpha}_i}) \times \Sigma_{\tilde{\alpha}_i} \rightarrow \Pi_{\tilde{\alpha}_i} \tag{3}$$

$$\Phi_{(\Delta_{\tilde{\alpha}_i})} : \Pi_{\tilde{\alpha}_i} \times \Sigma_{\tilde{\alpha}_i} \rightarrow \Delta_{\tilde{\alpha}_i} \tag{4}$$

$$\Phi_{\Gamma(\tilde{\alpha}_i)} : \Delta_{\tilde{\alpha}_i} \times \Sigma \rightarrow \Gamma_{\tilde{\alpha}_i} \tag{5}$$

Equations (3)–(5) define key functions for a fuzzy agent  $\tilde{\alpha}_i$  within a fuzzy multi-agent system  $\tilde{M}_\alpha$ , specifically as follows:

- Equation (3) represents the observation function, which maps the set of observed fuzzy events  $E_{\tilde{\alpha}_i}$  and interactions  $I_{\tilde{\alpha}_i}$ , along with the agent’s state  $\Sigma_{\tilde{\alpha}_i}$ , to its fuzzy perceptions  $\Pi_{\tilde{\alpha}_i}$ .
- Equation (4) defines the decision-making function, mapping fuzzy perceptions  $\Pi_{\tilde{\alpha}_i}$  and agent state  $\Sigma_{\tilde{\alpha}_i}$  to fuzzy decisions  $\Delta_{\tilde{\alpha}_i}$ .
- Equation (5) describes the action function, which maps fuzzy decisions  $\Delta_{\tilde{\alpha}_i}$  and the overall system state  $\Sigma$  to the agent’s fuzzy actions  $\Gamma_{\tilde{\alpha}_i}$ .

Here,  $E_{\tilde{\alpha}_i}$  denotes the set of fuzzy events observed by the agent,  $I_{\tilde{\alpha}_i}$  represents its fuzzy interactions,  $\Sigma_{\tilde{\alpha}_i}$  is the agent’s fuzzy state space,  $\Pi_{\tilde{\alpha}_i}$  represents the set of fuzzy perceptions,  $\Delta_{\tilde{\alpha}_i}$  denotes the fuzzy decisions made by the agent,  $\Gamma_{\tilde{\alpha}_i}$  represents the agent’s fuzzy actions, and  $\Sigma$  refers to the overall state of the fuzzy multi-agent system.

### 3.4. Fuzzy Interactions Between Two Fuzzy Agents

$$\tilde{l}_l = \langle \tilde{\alpha}_s, \tilde{\alpha}_r, \tilde{\gamma}_c \rangle \quad (6)$$

In Equation (6): for fuzzy interaction  $\tilde{l}_l$ ,  $\tilde{\alpha}_s$  is the fuzzy source agent,  $\tilde{\alpha}_r$  is the destination fuzzy agent, and  $\tilde{\gamma}_c$  is a fuzzy communication act (for instance, inform, diffuse, ask, or reply).

## 4. Fuzzy Decision Model for Battery Recharging

We propose a system in [33], designed from the literature, in particular [26], which aims to manage decisions regarding energy levels and tasks for a robot or autonomous agent. The primary goal is to decide whether the agent should recharge its energy or continue with its current task, based on its energy level and distances to the target and energy source.

The description of the model uses fuzzy logic as described in [10]. This approach relies on a formal knowledge representation using fuzzy sets, built on fuzzy elementary propositions of the form “ $V$  is  $A$ ”. These propositions are defined from a set,  $L(V, X, D_V)$ , consisting of the linguistic variable,  $V$ , the universe of possible values,  $X$ , and a set of descriptions,  $D_V$  for  $V$ , which are represented by fuzzy subsets of  $X$ .

Then, our proposed system uses three input variables and one output variable:

- **Energy Level**—input linguistic variable: *EnergyLevel*.
- **Distance to Target**—input linguistic variable: *DistanceToTarget*.
- **Distance to energy source**—input linguistic variable: *DistanceToEnergySource*.
- **Decision** (recharging or finishing the task)—output linguistic variable: *Decision*.

Each variable is defined over a specific range and described using linguistic terms, represented as fuzzy sets. We will describe each linguistic variable in the subsections below.

### 4.1. Input Linguistic Variables of the Fuzzy Decision Model

The fuzzy decision model relies on three primary input variables to make intelligent decisions regarding energy management and task execution. These input variables are defined as follows:

- **EnergyLevel**—this variable represents the energy level of the system, measured as a percentage from 0 to 100. Figure 1 illustrates the chosen membership functions for the input variable *EnergyLevel*. It is categorized into four linguistic terms:
  - *Critical*: [0, 20].
  - *Caution*: [10, 30].
  - *Operational*: [20, 75].
  - *Full*: [65, 100].
- **DistanceToTarget**—this variable measures the distance to the target in meters, ranging from 0 to 100. Figure 2 depicts the chosen membership functions for the input variable *DistanceToTarget*. It is described using three linguistic terms:
  - *Close*: [0, 25].
  - *Medium*: [10, 40].
  - *Far*: [25, 100].
- **DistanceToEnergySource**—this variable indicates the distance to the energy source in meters, also ranging from 0 to 100. Figure 3 shows the chosen membership functions for the input variable *DistanceToEnergySource*. It is classified into three linguistic terms:
  - *Close*: [0, 25].
  - *Medium*: [10, 40].
  - *Far*: [25, 100].

These variables form the foundation of the fuzzy decision-making process, allowing the system to interpret and respond to different scenarios based on the defined fuzzy rules.

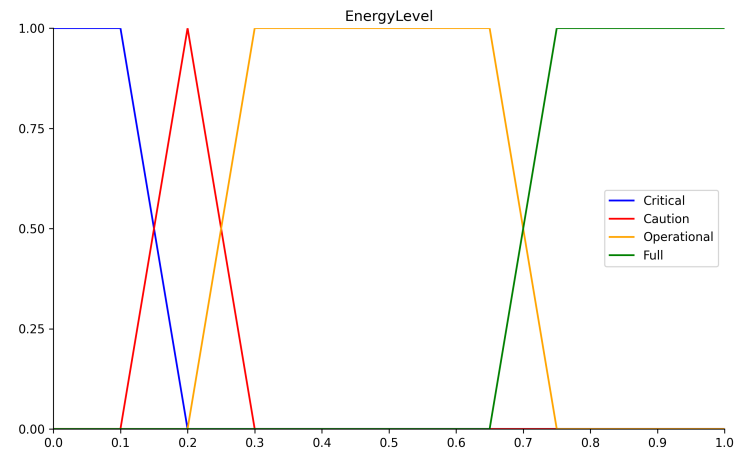


Figure 1. Energy level.

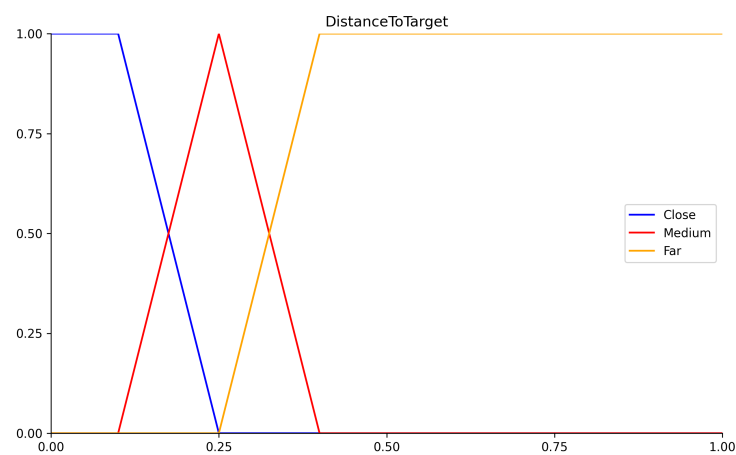


Figure 2. Distance to target.

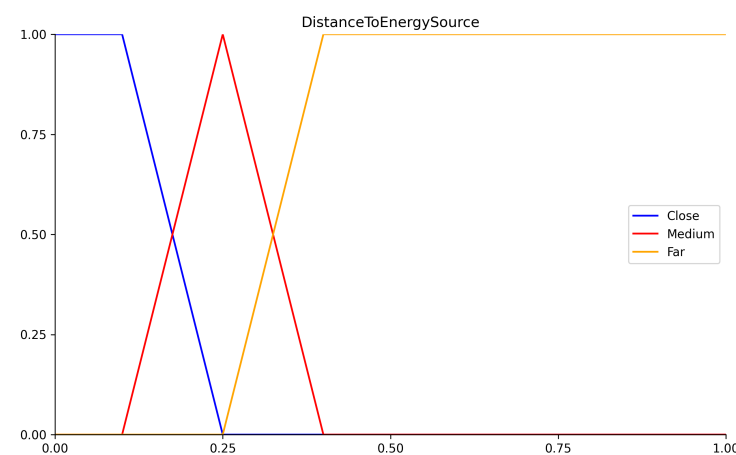


Figure 3. Distance to energy source.

#### 4.2. Output Linguistic Variable of the Fuzzy Decision Model

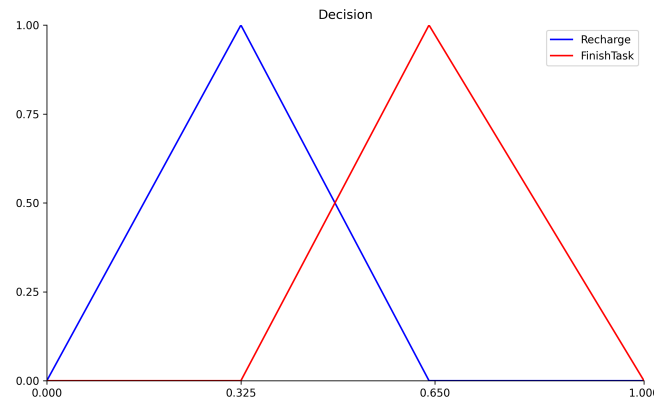
The fuzzy decision model produces one output variable that guides the system’s actions regarding energy management. This output variable is defined as follows:

- **Decision**—this variable ranges from 0 to 1 and determines whether the system should recharge or finish its current task. Figure 4 shows the chosen membership functions for the output variable *Decision*. It is described using two linguistic terms:
  - *Recharge*: [0, 0.65].



- *FinishTask*: [0.325, 1].

This output variable enables the system to make informed decisions based on the input variables and the fuzzy rules, ensuring optimal performance and energy efficiency.



**Figure 4.** Decision.

#### 4.3. Fuzzy Rules

The fuzzy rules establish the decision-making framework of the system. Each rule outlines conditions based on the input variables and determines the corresponding output decision. Our system adheres to the following rules:

1. If *EnergyLevel* is *Critical*, then *Decision* is *Recharge*.
2. If *EnergyLevel* is *Caution*, and *DistanceToTarget* is *Close*, and *DistanceToEnergySource* is *Far*, then *Decision* is *Recharge*.
3. If *EnergyLevel* is *Caution*, and *DistanceToTarget* is *Close*, and *Distance to EnergySource* is *Medium*, then *Decision* is *FinishTask*.
4. If *EnergyLevel* is *Caution*, and *DistanceToTarget* is *Close*, and *DistanceToEnergySource* is *Close*, then *Decision* is *FinishTask*.
5. If *EnergyLevel* is *Caution*, and *DistanceToTarget* is *Medium*, then *Decision* is *Recharge*.
6. If *EnergyLevel* is *Caution*, and *DistanceToTarget* is *Far*, then *Decision* is *Recharge*.
7. If *EnergyLevel* is *Operational*, then *Decision* is *FinishTask*.
8. If *EnergyLevel* is *Full*, then *Decision* is *FinishTask*.

To find the final decision, the first step in the fuzzy inference process is fuzzification, where the system evaluates each elementary condition in the rule premises. This involves taking the precise input values and mapping them to fuzzy values based on their membership in predefined fuzzy sets.

Once the input values are fuzzified, the system calculates the premises of each rule (“IF  $V$  is  $A$ ”). The result of this calculation is known as the activation value of the rule, which indicates how strongly the rule is triggered by the current inputs. Then, the implication function is applied to the conclusion of each rule, producing a fuzzy subset that represents the distribution of possible output values based on the activated rule.

These fuzzy subsets, corresponding to the same output across different rules, are then combined using an aggregation method—often the maximum function. This step merges the outputs of all relevant rules into a single fuzzy subset.

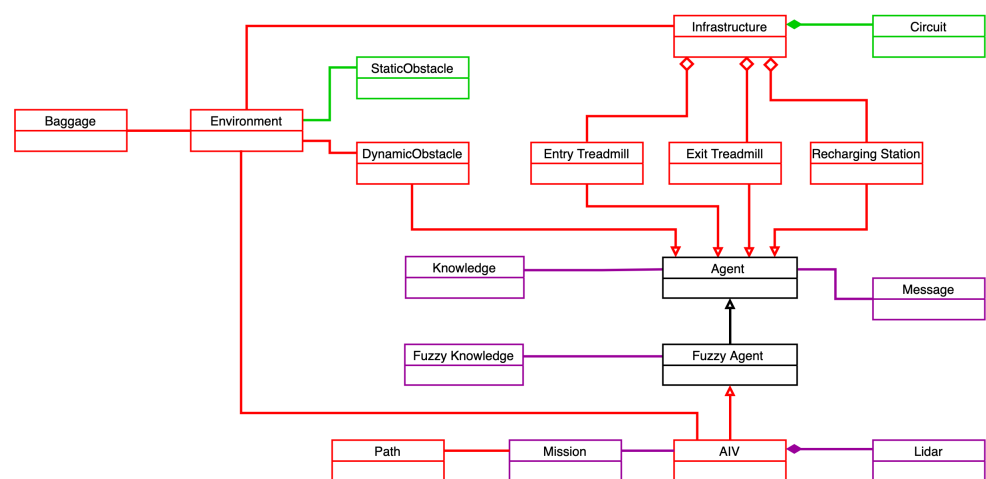
Finally, the aggregated fuzzy subset is converted into a precise output value through a process called defuzzification. Among various defuzzification methods, one of the most common is the centroid or barycenter method, which calculates the center of gravity of the area under the curve representing the fuzzy subset. We used this method in our simulations because it produces a single, clear output value that the system can use to make a final decision.

### 5. Case Study: Autonomous Management of Battery Recharging

We present an adaptable, fuzzy, multi-agent model in Figure 5 that addresses the challenges of energy management for AIVs [33]. The efficient management of AIVs requires a holistic approach that takes into account several factors, including operational availability, energy consumption [34], collaboration between AIVs and the dynamic infrastructure, and their adaptation to changing conditions. We aim to optimize recharging based on energy costs, as a low workload combined with frequent recharging can increase the overall energy consumption of the system. In addition, poor anticipation can limit system availability.

AIV missions do not follow a uniform distribution in terms of frequency, creating periods of intense activity and others that are quieter. It is, therefore, essential to link the energy consumption of AIVs to the amount of work carried out and their operational availability.

To avoid an overload of recharging requests due to too many simultaneous requests, the AIVs need to work together by communicating with each other or via the infrastructure. As for automatic recharging, although it solves the problem of the number of charges, it requires space and consumes energy. Even a 2 to 3% reduction in energy consumption is significant for certain warehouses and airports. For the introduction of fleets of autonomous vehicles in the industry of the future, it, therefore, seems necessary to fine-tune the number of recharging stations. This sizing can be improved by taking into account the possibilities for communication between the AIVs, which can collectively avoid critical (urgent) recharging.



**Figure 5.** Simulator agent architecture: dynamic elements in red, static in green, and not related to the environment in purple.

#### Description of the Simulation Framework

The interface of our “Airport Baggage Handling Simulation” application, in Figure 6, is designed to provide a comprehensive overview of the autonomous management and operation of AIVs for baggage handling. This intuitive and structured interface allows us to monitor and analyze the performance of AIVs. The variables used in our case study simulation, such as the number of baggage, AIV speed, circuit size, and the number of operational nodes, are summarized in Table 1. The circulation scenario is detailed with a distance-oriented graph presented in Figure 7. The interface is divided into several sections, each displaying critical information about the simulation status and AIV performance.

The different sections of the interface are as follows:

- **Energy-level display:** Each AIV’s energy level is represented by a horizontal bar, which visually indicates the remaining battery life. For instance, *AIV3*’s bar is red, signifying a critical battery level, while the other AIVs have yellow bars, indicating varying levels of charge.
- **Charge-level information:** Below the energy-level display, this section provides detailed statistics on the recharging activities:

- Recharge time: The total time spent on recharging.
- Waiting time: The total waiting time before recharging.
- Recharges per AIV: The number of recharge cycles completed by each AIV.
- **Simulation area:** The central area of the interface depicts the simulation environment. It visually represents the positions and movements of the AIVs as they handle baggage within the airport layout. Different black points represent the location of each node corresponding to the oriented graph (Figure 7).
- **Baggage-level information:** This section displays baggage handling metrics:
  - Total baggage: The total number of bags that must be dealt with from the start of the scenario.
  - Waiting baggage: The maximum number of bags waiting to be processed at a moment.
  - Baggage per AIV: The number of bags handled by each AIV.
- **Time-level information:** Below the baggage-level section, this displays time-related data:
  - Simulation time: The total elapsed time of the simulation.
  - Missions per AIV: The average duration of missions completed by each AIV.

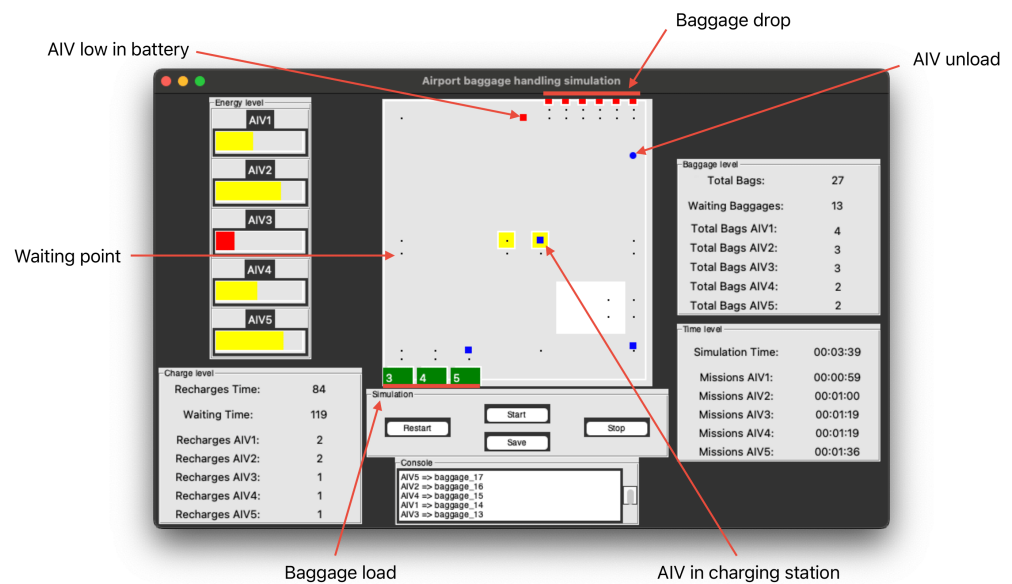


Figure 6. Airport baggage handling simulation HMI.

Table 1. Variables of the case study simulation.

Variable of Simulations	Values
Number of bags	1000
Speed of AIV (m/s)	[0, 7.5]
Circuit size (m × m)	275 × 275
Number of nodes	33
Number of recharging stations	2
Number of waiting nodes	2
Number of baggage drop nodes	6
Number of baggage loading nodes	3

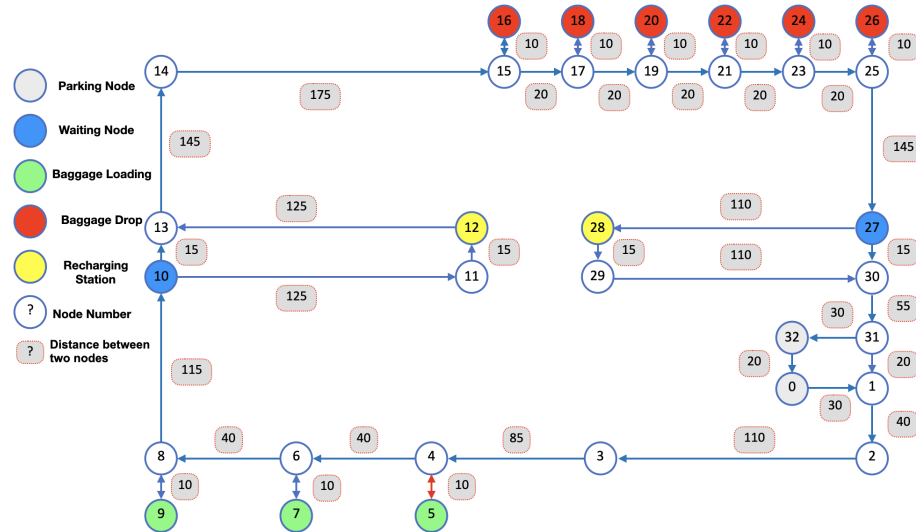


Figure 7. Oriented graph: distance in the environment in meters.

### 6. Comparisons Between Thresholds and Fuzzy Logic Models

To test different autonomous management strategies for solving the problem of AIVs recharging batteries, we defined an initial circulation environment (Figure 6). We proposed different scenarios and compared them with the following four parameters:

- *nbMissions*: number of missions carried out.
- *timeMission*: the average time taken to complete a mission in seconds.
- *nbRecharges*: the number of recharges performed.
- *wtRecharging.*: waiting times for recharging in seconds

We also varied the charge threshold at which an AIV must recharge its battery. We then introduced a fuzzy inference system to determine the recharge time. We also varied the values of the fuzzy model (fuzzy linguistic values).

In this section, we delve into a comparative analysis between different thresholds and fuzzy logic models. We propose three different scenarios:

- Scenario 1 (*sc1*): all AIVs have a uniform recharge threshold of 30%.
- Scenario 2 (*sc2*): each AIV has a different recharge threshold, maintaining the same context as *sc1*.
- Scenario 3 (*sc3*): AIVs use a fuzzy logic model for recharge.

We simulated these three scenarios for 1000 bags (a discussion regarding the scenario results is provided in the following three sections). The temporal results are shown in Table 2. We aim to discern the optimal threshold configurations that maximize mission throughput, minimize recharging frequency, and optimize resource utilization, thereby improving the overall efficiency of autonomous management strategies for recharging the AIV battery.

Table 2. Time results for 1000 bags for Sc1, Sc2, and Sc3.

Scenarios	Sc1	Sc2	Sc3
Number of bags	1000	1000	1000
Total recharge time (s)	4800	4619	4345
Total simulation time	03:42:54	03:42:42	03:38:34

#### 6.1. Basic Scenario

In the *basic scenario*, AIVs have a single threshold model set to 30% for recharging. This scenario makes it possible to compare performance in terms of the mission processing

time (overall and individual time), number of recharges, and waiting time for recharges (access to a free station). The AIVs results for *sc1* are shown in Table 3.

**Table 3.** AIV results for Scenario 1.

Indicators	AIV1	AIV2	AIV3	AIV4	AIV5	Global
<b>Thresholds</b>	30	30	30	30	30	
<i>nbMissions</i>	201	199	200	200	200	1000
<i>timeMission</i>	64	64	64	64	64	64
<i>nbRecharges</i>	80	80	80	80	80	400
<i>wtRecharging</i>	0	93	42	68	93	244

### 6.2. Different Threshold Values

*sc2* enables us to compare different threshold values for AIVs recharge. The results are depicted in Table 4. When we compare with thresholds varying between 15% and 30%, the overall mission processing time is slightly lower, and the number of recharges and overall recharge time are also lower (374 and 400, respectively). The performance of *AIV1* with the lowest threshold (15%) is obviously the best for the average time taken to complete a mission time. However, there is a greater risk of not being able to reach a station due to a lack of charge in the event of an incident!

**Table 4.** AIV results for Scenario 2.

Indicators	AIV1	AIV2	AIV3	AIV4	AIV5	Global
<b>Thresholds</b>	15	20	25	30	35	
<i>nbMissions</i>	202	201	199	199	202	1000
<i>timeMission</i>	63	64	64	64	64	63.8
<i>nbRecharges</i>	67	67	80	80	80	374
<i>wtRecharging</i>	180	140	49	77	51	497

### 6.3. Fuzzy Logic Model

In comparison with *sc1*, where AIVs have a threshold of 30%, in *sc3*, AIVs use a fuzzy basic model. The results are presented in Table 5 demonstrate an improvement in overall and individual AIV times (63 s on average instead of 64 s) and fewer recharges (335 recharges instead of 400).

**Table 5.** AIV results for Scenario 3.

Indicators	AIV1	AIV2	AIV3	AIV4	AIV5	Global
<b>FL model</b>	FL	FL	FL	FL	FL	
<i>nbMissions</i>	201	200	200	200	199	1000
<i>timeMission</i>	63	63	63	63	63	63
<i>nbRecharges</i>	67	67	67	67	67	335
<i>wtRecharging</i>	0	58	19	49	71	197

## 7. Increases in Fuzzy Logic Criteria

To improve the results of the previous simulations, we applied three types of adaptation (heuristics), taking into account more realistic constraints and the possibility of AIVs communicating with each other and with infrastructure elements such as charging stations:

1. Adaptation of recharging according to the needs of the AIVs and the availability of the charging points (centralized scenario with supervision and decentralized scenario with communication between the AIVs and the charging points);
2. Adaptation of recharging according to the rate of baggage arrival and the resulting variation in activity (the number of missions to be performed by the AIVs in a unit of time is no longer constant);
3. Adapting the speed of the AIVs according to the rate of baggage arrival (centralized scenario with supervision and decentralized scenario with communication between the AIVs and the charging points).

The objective of this section is to show that specific heuristics allow certain situations to be dealt with fairly finely and increase the collective/overall performance of AIVs. We simulated these three improved scenarios for 1000 baggages. The temporal results are shown in Table 6.

**Table 6.** Time results and configuration for 1000 bags for Sc4, Sc5, and Sc6.

Scenarios	Sc4	Sc5	Sc6
Number of bags	1000	1000	1000
Total recharge time (s)	4606	4285	935
Total simulation time	03:39:08	03:37:03	01:32:15
Maximum number of waiting bags	468	659	159
Average baggage waiting	234	327	99.62

### 7.1. Adapting Recharging to Demand and the Availability of Charging Points

The first heuristic, referred to as *sc4*, simulates the adaptation of charging behavior to both demand and the availability of charging points. The AIV results are presented in Table 7. The effectiveness of this heuristic is obvious, particularly for *AIV1*, which required 14 fewer recharges compared to *AIV5* and 12 fewer recharges than *AIV4*. Additionally, the total recharging time for *sc4* is shorter than for both *sc1* and *sc2*: 4606 s compared to 4800 s and 4619 s, respectively (Table 6 for *sc4* and Table 2 for *sc1* and *sc2*).

The input linguistic variables are as follows: *EnergyLevel*, *DistanceStation1*, and *DistanceStation2*. The output linguistic variable is *Decision*. The fuzzy rules for the distances between stations 1 and 2 are as follows:

1. If *EnergyLevel* is *Critical*, and *DistanceStation2* is *Far*, then *Decision* is *Station1*.
2. If *EnergyLevel* is *Critical*, and *DistanceStation1* is *Far*, then *Decision* is *Station2*.
3. If *EnergyLevel* is *Caution*, and *DistanceStation1* is *Medium*, then *DistanceStation2* is *Far*, and then *Decision* is *Station1*.
4. If *EnergyLevel* is *Caution*, and *DistanceStation1* is *Far*, then *DistanceStation2* is *Medium*, and then *Decision* is *Station2*.

**Table 7.** AIV results for Scenario 4.

Indicators	AIV1	AIV2	AIV3	AIV4	AIV5	Global
Thresholds	15/15	20/15	25/20	30/20	35/25	
<i>nbMissions</i>	203	201	199	198	199	1000
<i>timeMission</i>	62	62	63	63	63	62.6
<i>nbRecharges</i>	68	67	78	76	80	369
<i>wtRecharging</i>	143	178	5	10	70	406

### 7.2. Adaptation of Recharging According to the Baggage Arrival Rate

Heuristic 2, referred to as *sc5*, simulates recharging adaptations based on the baggage arrival rate and the corresponding variation in induced activity (the number of tasks to be

performed by the AIVs). As shown in Table 8, this heuristic enables the AIVs to complete their missions more quickly compared to sc4. Specifically, AIVs complete one mission in an average of 58 s under sc5, whereas it takes them 62.6 s under sc4 (see Table 7).

The input linguistic variables are *EnergyLevel*, *DistanceStation1*, *DistanceStation2*, *AvailabilityStation1*, and *AvailabilityStation2*. The output linguistic variable is *Decision*. The fuzzy rules for the distance and availability criteria for stations 1 and 2 are as follows:

1. If *EnergyLevel* is *Critical*, and *DistanceStation1* is *Near*, and *DistanceStation2* is *Far*, then *Decision* is *Station1*.
2. If *EnergyLevel* is *Critical*, and *DistanceStation1* is *Far*, and *DistanceStation2* is *Near*, then *Decision* is *Station2*.
3. If *EnergyLevel* is *Caution*, and *DistanceStation1* is *Medium*, and *DistanceStation2* is *Far*, and *AvailabilityStation1* is *High*, then *Decision* is *Station1*.
4. If *EnergyLevel* is *Caution*, and *DistanceStation1* is *Medium*, and *DistanceStation2* is *Far*, and *AvailabilityStation1* is *Weak*, and *AvailabilityStation2* is *High*, then *Decision* is *Station2*.
5. If *EnergyLevel* is *Caution*, and *DistanceStation1* is *Far*, and *DistanceStation2* is *Medium*, and *AvailabilityStation2* is *High*, then *Decision* is *Station2*.
6. If *EnergyLevel* is *Caution*, and *DistanceStation1* is *Far*, and *DistanceStation2* is *Medium*, and *AvailabilityStation1* is *High*, and *AvailabilityStation2* is *Weak*, then *Decision* is *Station1*.

**Table 8.** AIV results for Scenario 5.

Indicators	AIV1	AIV2	AIV3	AIV4	AIV5	Global
Thresholds	20	20	20	20	20	
<i>nbMissions</i>	200	199	201	200	200	1000
<i>timeMission</i>	58	58	58	58	58	58
<i>nbRecharges</i>	15	15	15	15	15	75
<i>wtRecharging</i>	66	0	25	57	94	242

### 7.3. Adapting the Speed of the AIVs to the Flow of Baggage Arrivals

The final heuristic, referred to as sc6, adapts the speed of the AIVs to the flow of baggage arrivals. Compared to sc5, the 30% speed threshold was adjusted, as the 20% threshold led to too many load faults due to increased energy consumption at higher speeds. This adjustment results in a significantly shorter overall simulation time, as shown in Table 6. Additionally, Table 9 demonstrates improved throughput control, with baggage waiting times reduced to 99.6 s in this scenario, compared to 327 s for sc5 (see Table 8).

The input linguistic variables are *EnergyLevel*, *Urgency*, and *ProximityAIV*. The output linguistic variable is *Speed*. The fuzzy rules for the speed criteria are as follows:

1. If *Urgency* is *Weak*, then *Speed* is *Normal*.
2. If *EnergyLevel* is *Critical*, and *Urgency* is *Weak*, then *Speed* is *Weak*.
3. If *EnergyLevel* is *Critical*, and *Urgency* is *Medium*, and *ProximityAIV* is *High*, then *Speed* is *Weak*.
4. If *EnergyLevel* is *Caution*, and *Urgency* is *Medium*, and *ProximityAIV* is *Medium*, then *Speed* is *Normal*.
5. If *EnergyLevel* is *Operational*, and *Urgency* is *Medium*, and *ProximityAIV* is *Weak*, then *Speed* is *High*.
6. If *EnergyLevel* is *Caution*, and *Urgency* is *High*, and *ProximityAIV* is *Medium*, then *Speed* is *Normal*.
7. If *Urgency* is *High*, then *Speed* is *High*.

**Table 9.** AIV results for Scenario 6.

Indicators	AIV1	AIV2	AIV3	AIV4	AIV5	Global
Thresholds	30	30	30	30	30	
<i>nbMissions</i>	201	199	200	199	201	1000
<i>timeMission</i>	25	25	25	25	25	25
<i>nbRecharges</i>	17	17	17	17	17	85
<i>wtRecharging</i>	0	19	76	20	54	169

## 8. Methodology Proposed for Energy Management with a Realistic Point of View

The variation in energy consumption is influenced by the acceleration and deceleration operations when using a stepping motor drive. As a general rule, the switching times for acceleration and deceleration have an impact on energy consumption. The choice of the cost function is crucial for optimizing problems in mobile robots. In some studies, the cost function is defined as the minimization of the energy supplied by the motor, represented by the integral of the squared motor force or the sum of the squared linear and angular velocities of the mobile robot. However, this function is optimal only for very small values, as it is more than a simple squared function. There are alternative cost functions for energy optimization, but they often include additional parameters, such as those related to the battery.

In this paper, we are interested in a cost function that accommodates three types of motion simultaneously: acceleration, deceleration, and constant speed. The variation in the system's energy is defined by a force that represents these three motions. The change in force during the target phase also affects the work done. Work, in this context, refers to the means by which forces transfer energy. Therefore, the cost function is defined as the absolute work performed by the motor force.

This cost function enables us to tackle an optimization problem with a basic dynamic system. Despite the simplification of the mathematical model, a numerical method based on time discretization is not well suited to the local environment of the application.

This section proposes a methodology to address this problem, operation by varying the maximum speed values under each node, using an optimal control problem. The biggest advantage of this methodology is that the numerical optimization methods are implemented to create the optimal maximum speed values. This technique reduces the execution time of this method in real time.

The proposed methodology calculates the motor force, which serves as the command for the optimization problem. It also involves determining the moments of acceleration and deceleration. To define these moments, an algorithm is implemented to compute the acceleration and deceleration distances. This algorithm operates based on distance, rather than time, making it suitable for implementation in the application.

Three strategies are generated by this methodology. With the help of two inputs: waiting baggage and traffic, a fuzzy logic model can select the most appropriate strategy for the AIV during its mission.

### 8.1. Optimal Control Problem

This part addresses an Optimal Control Problem (OCP) for the optimization model, focusing on a mobile robot. The problem (OCP) combines a dynamic system, which is given by a mobile robot model, where the mobile robot is moving in a straight line and a cost function that are described according to the absolute work of the motor force, which represents the amount of energy consumed by the AIV during his mission.



### Dynamic Model of Robot

The mobile robot model is derived from Newton’s second law:

$$\sum F = M a$$

where  $F$  is the sum of all forces acting on an AIV,  $a$  is the linear acceleration of the AIV, and  $M$  is the total mass (AIV + baggage).

Based on the specific route of the AIVs, the system operates under the following simplifying assumptions:

- **Assumption 1:** an AIV moves in a straight line and does not consider curvilinear motion or slopes.
- **Assumption 2:** air resistance is negligible.
- **Assumption 3:** the energy consumed by the motor is included in the cost function, without constraints on the dynamic system.

Based on these assumptions, an AIV dynamic system is described as a double integrator:

$$\begin{cases} \dot{x} = v \\ \dot{v} = \frac{F_T}{M} \end{cases} \tag{7}$$

where,  $x$ ,  $v$ , and  $\ddot{x}$  are the longitudinal position, the linear velocity, and the linear acceleration of the AIV, respectively. The ratio between force and mass,  $\frac{F_T}{M}$ , represents the linear acceleration,  $a$ . Then, the maximum force can be deduced from the high acceleration and the mass.

In the following, we assume that the forces acting on an AIV is composed of two forces:

$$F_T = F_m - F_r$$

where  $F_m$  is propelled by the motor force, and  $F_r$  is the resistance force.

### 8.2. Cost Function

To account for the varying cost function due to the changes in the robot’s movement (acceleration, deceleration, and stopping) from the starting position to the final position over a period from  $T_o$  to the final time,  $T$ , we need to refine our model. Specifically, we incorporate the power delivered via the robot’s motor, which depends on the force.

The power,  $P$ , delivered via each AIV motor at any time,  $t$ , is given by the following:

$$P = F_m v \tag{8}$$

The absolute work,  $W$ , is performed via the propulsion force over the period from  $T_0$  to  $T$ :

$$W = \int_{T_o}^T |P(t)| dt = \int_{T_o}^T |F_m v| dt \tag{9}$$

The optimization problem (OCP) admits an optimal solution, and the linear speed,  $v$ , is always non-negative ( $v \geq 0$ ) during the mission of the robot, as stated in [35].

Then, the cost function is given by the following:

$$\int_{T_o}^T |F_m| v dt \tag{10}$$

### 8.3. Optimal Control Problems (OCP)

The formulated problem is defined by the cost function and the AIV model presented above:

$$\left\{ \begin{array}{l} \text{Minimize} \quad W = \int_{T_0}^T |F_m| v \, dt \\ \text{Subject to} \\ \dot{x} = v \\ \dot{v} = \frac{F_m - Fr}{M} \\ x(T_0) = x_0, v(T_0) = v_0, v(T) = v_f, x(T) = x_f > 0 \\ T_0 \text{ and } T : \text{fixed} > 0 \\ F_{min} \leq F_m \leq F_{max} \end{array} \right. \quad (11)$$

where  $x_0$  and  $v_0$  are initial values of states,  $x_f$  and  $v_f$  are final values of states, and  $F_{min}$  and  $F_{max}$  are the lower and upper bounds of the motor force  $F_m$ .

### 8.4. Methods for Solving the Problem (OCP): Pontryagin’s Maximum Principle

This subsection is dedicated to showing the approach used. The primary tool used is PMP [7], which ensures a necessary optimality condition for the optimal control problem. This is achieved by introducing the adjoint vector  $P = (p_1, p_2)$  to the state vector  $X = (x, v)$ , and the Hamiltonian function  $H(X, P, F_m)$  is then defined as follows:

$$H(X, P, F_m) = -|F_m|v + p_1v + p_2 \frac{F_m - Fr}{M} \quad (12)$$

The pair  $(X, F_m)$  is called an admissible solution if  $X$  is the state trajectory corresponding to  $F_m$  and the conditions of the optimal control problem (11) are satisfied. An admissible pair that minimizes the cost function is called an optimal solution, denoted by  $(X^*, F_m^*)$ .

The adjoint vector  $P = (p_1, p_2)$  verifies these equations:

$$\left\{ \begin{array}{l} \dot{p}_1 = -\frac{\partial H(X, P, F_m)}{\partial x} = 0 \\ \dot{p}_2 = -\frac{\partial H(X, P, F_m)}{\partial v} = |F_m| - p_1 \end{array} \right. \quad (13)$$

The optimal control problem (11) is autonomous; then,  $H(X, P, F_m)$  is constant. The optimal control  $F_m^*$  maximizes the Hamiltonian almost everywhere:

$$\begin{aligned} H &= \max_{F_m} H(X, P, F_p) \\ &= p_1v - \frac{p_2}{M}Fr + \max_{F_m} \left\{ -|F_m|v + \frac{p_2}{M}F_m \right\} \end{aligned} \quad (14)$$

According to [35], the optimal strategy for the minimum time problem is given by the following:

$$F_m^* = \begin{cases} F_{max} & \text{for } 0 \leq t \leq \Delta t_0 \\ F_{min} & \text{for } \Delta t_0 < t \leq T_{min} \end{cases} \quad (15)$$

The expression of the minimum travel time,  $T_{min}$ , with  $\Delta t_0 = \frac{T_{min}}{2}$ :

$$T_{min} = - \frac{M(F_{min}v_0 - F_{max}v_f) + \sqrt{M(F_{max} - F_{min}) \left( 2F_{max}F_{min}(x_f - x_0) - M(F_{min}v_0^2 - F_{max}v_f^2) \right)}}{F_{max}F_{min}} \quad (16)$$

This strategy is reflected in the accelerator being at full travel until it reaches the maximum speed (high acceleration). Then, very high braking is applied until the motor stops, as there is not enough time.

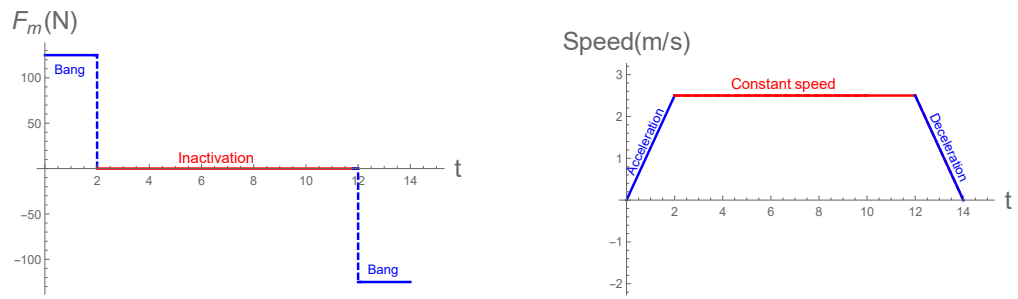
For  $T_{min} < T < T_{lim}$ ,  $T_{lim}$  is the time limit when the optimal strategy of the problem (11) is generated via the following:

$$F_m^* = \begin{cases} F_{max} & \text{for } 0 \leq t \leq \Delta t \\ 0 & \text{for } \Delta t < t \leq T - \Delta t \\ F_{min} & T - \Delta t < t \leq T \end{cases} \quad (17)$$

The terminal condition  $x_f$ , allows the explicit computation of  $\Delta t$ :

$$\Delta t = \frac{F_{max}T - \sqrt{F_{max}^2T^2 + 2M(F_{max} - F_{min})(x_0 - x_f + v_0T)}}{F_{max} - F_{min}} \quad (18)$$

The optimal strategy is illustrated in Figure 8 with its optimal speed profile, which was determined through the dynamic system. A solution is the *bang* type, if the control is equal to its maximum, or its minimum, which corresponds to the maximum acceleration or deceleration phase. An *inactivation* is defined when the control variable is null over a time interval, which corresponds to maintaining a constant speed.



**Figure 8.** Example: optimal command for  $T > T_{min}$  with  $T_{min} = 9.79$  and speed over a distance of 30 m.

### 8.5. Discrete Time Problem

Let us introduce a discrete time version of the problem (11). Given  $N$  and  $h = \frac{T-T_0}{N}$  defined for  $i = 0, \dots, N$ ,

$$\begin{cases} \text{Minimize} & W = \sum_{i=0}^N |F_m(i)| v(i) \\ \text{Subject to} & \\ & x(i+1) = x(i) + hv(i), \quad i = 0, \dots, N \\ & v(i+1) = v(i) + h \frac{F_m(i) - F_r(i)}{M}, \quad i = 0, \dots, N \\ & x(0) = x_0, v(0) = v_0, v(N-1) = v_f, x(N-1) = x_f > 0 \\ & F_{min} \leq F_m(i) \leq F_{max}, \quad i = 0, \dots, N \end{cases} \quad (19)$$

To solve this discrete time problem numerically, we can discretize the time horizon  $[0, T]$  into  $N$  intervals with a time step,  $h = \frac{T-T_0}{N}$ , for  $i = 0, \dots, N$ . Then, we can apply an optimization algorithm, such as a direct method (e.g., direct collocation).

## 9. Energy Control Strategies

We developed an energy control strategies algorithm based on an optimal speed profile that is defined from the optimal control problem. This algorithm is based on the distance traveled, using basic principles of acceleration and deceleration, and applying these principles to small segments of the path. This method is practical for short distances between nodes, and it provides a systematic way to control the robot's movement efficiently. A more practical and simple approach can be applied to determine the optimal control strategy for minimizing energy consumption.

By implementing three strategies based on the speed profile as a function of the robot's actual position, these speed profiles are generated via an algorithm capable of operating in real time.

#### *Proposed Algorithm*

We propose Algorithm 1, which connects two maximum speed values assigned to each node. This algorithm identifies two key distances—acceleration and deceleration distances—in order to link the two maximum speed values. The maximum speed values can be obtained using solutions generated via the PMP. However, these maximum speed values are optimal under certain constraints, and they need to be adapted to fit our specific application.

#### **Detailed steps of the algorithm:**

These instructions outline the steps for comparing speeds, calculating distances, making decisions, and setting commands based on those decisions. Adjustments can be made according to specific requirements or additional constraints.

- Input:
    - Path nodes with associated maximum speed values.
    - AIV initial position and speed.
  - Output:
    - Real-time energy consumed and speed profile for the AIV.
1. Initialization:
    - Identify the starting and ending nodes of the path.
    - Assign maximum speed values to each node using the PMP.
  2. Calculate acceleration and deceleration distances:
    - For each pair of nodes, calculate the distance required for the robot to accelerate from the current speed to the maximum speed of the next node.
    - Similarly, calculate the distance required for deceleration.
  3. Generate speed profile:
    - For each segment between nodes, generate a speed profile that smoothly transitions between the two maximum speed values.
    - Ensure the profile adheres to the calculated acceleration and deceleration distances.
  4. Real-time adjustment:
    - Continuously monitor the AIV position and adjust the speed profile in real time to account for any deviations or changes in the path.

In the application, step 1 “Offline Method: Table of Maximum Speed”, represents the three strategies, which are included as a table or database of the maximum speed for each node. The second step, “Online Method: Current Speed, Energy”, will be called up in real time under the fuzzy rule, which decides what type of strategy can be adapted to meet the constraints of waiting baggage and traffic flow. The energy constraint will be more advantageous for waiting baggage and traffic, which are defined as model inputs, with the three strategies as outputs of the fuzzy decision model in the next section. Finally, we define three strategies, *strategy1*, *strategy2*, and *strategy3*, which represent *LowEfficiency*, *MediumEfficiency*, and *StrongEfficiency*, respectively.

**Algorithm 1: Energy and current speed**


---

**Step1** → **Offline Method: Table of Speed Max**  
**Input:**  $M, a_{max}, F_r, x_0, x_f, v_0, v_f$   
**Output:** TableSpeedMax

1 **Function** [TableSpeedMax]=**Optimal-SpeedMax**( $M, a_{max}, x_0, x_f, v_0, v_f$ )

2      $F_{max} = M * a_{max}$   
3      $F_{min} = -M * a_{max}$   
4      $T_{min} = f(M, F_{max}, F_{min}, x_0, x_f, v_0, v_f)$  –cf. Equation (16)  
5     Fixed  $T > T_{min}$   
6      $\Delta t = g(T, M, F_{max}, F_{min}, x_0, x_f, v_0, v_f)$  –cf. Equation (18)  
7      $SpeedMax = \frac{F_{max}-F_r}{M} \Delta t + v_0$

**Step 2** → **Online Method: Current Speed, Energy**  
**Input:**  $M, h, a_{max}, NextNode, CurrentDistance, TableSpeedMax$   
**Output:** Energy, Current Speed

8 **Function** Energy=**MotorEnergy**(Speed, FinalCommand)  
9      $Energy = Energy + abs(FinalCommand) * Speed$

10 **Function** Speed=**CurrentSpeed**(Speed, FinalCommand,  $h, M, F_r$ )  
11      $Speed = Speed + h * \frac{FinalCommand - F_r}{M}$

12 **Function** FinalCommand = **Command**(CurrentPosition, NextNode, Command<sub>sup</sub>, Command<sub>inf</sub>, Distance1, Distance2)

13     **if** (CurrentPosition – Distance1) ≤ 0 **then**  
14          $FinalCommand = Command_{sup}$   
15     **else if** (NextNode – CurrentPosition – Distance2) ≤ 0 **then**  
16          $FinalCommand = Command_{inf}$   
17     **else**  
18          $FinalCommand = 0$   
19     **end if**

20 **Function** [Command<sub>sup</sub>, Command<sub>inf</sub>, Distance1, Distance2] =  
**Acceleration – Deceleration**(PreviousSpeedMax, NextSpeedMax, NextNextSpeedMax,  $M, a_{max}, h$ )

21      $F_{max} = M * a_{max}$   
22      $F_{min} = -M * a_{max}$

23     1. Compare the PreviousSpeedMax and NextSpeedMax values, as well as the NextSpeedMax and NextNextSpeedMax values, to then establish an acceleration or deceleration decision at the exit of each of the node.  
24     2. Two distances are calculated based on the speed comparisons:  
25          $Distance1 = |PreviousSpeedMax - NextSpeedMax| * N1$   
26          $Distance2 = |NextSpeedMax| * h$   
27         where N1 represents the number of points between PreviousSpeedMax and NextSpeedMax, depending on the step of discretization,  $h$ .  
28     3. Between each node, two decisions are defined. Commands are determined based on whether the decision is to accelerate, decelerate, or maintain a constant speed:

**if** Decision == Acceleration **then**  
   |      $Command_{sup} = F_{max}$ ;  
   **else if** Decision == Deceleration **then**  
   |      $Command_{sup} = F_{max}$ ;  
   **else**  
   |      $Command_{sup} = 0$   
   |      $Command_{inf} = 0$   
   **end if**

24 **Function** [PreviousSpeedMax, NextSpeedMax, NextNextSpeedMax]= **NodeSpeed** (TableSpeedMax, PreviousNode, NextNode, NextNextNode)

25     PreviousNode → PreviousSpeedMax  
26     NextNode → NextSpeedMax  
27     NextNextNode → NextNextSpeedMax

---

**9.1. Fuzzy Model for Energy Control**

We propose a fuzzy logic model that allows us to determine which strategy to implement. The system uses two input variables and one output variable:

1. Input linguistic variable: **WaitingBaggage**—*Wb*.
2. Input linguistic variable: **traffic**—*Traffic*.
3. Output linguistic variable: **strategy**—*Strategy*.

#### 9.1.1. Input Linguistic Variables of the Fuzzy Decision Model to Determine the Strategy

The fuzzy decision model relies on two primary input variables to make intelligent decisions regarding the strategy to use. These input variables are defined as follows:

- **Waiting baggage**—this variable represents the waiting baggage level of the system, measured as a number of bags from 0 to 1000. It is categorized into three linguistic terms:
  - *Low*: [0, 5].
  - *Medium*: [2, 5, 8].
  - *Strong*: [5, 1000].
- **Traffic**—this variable represents the traffic level of the system, measured as a number of AIVs from 0 to 5 (in our case study simulation), *Traffic*. It is categorized into three linguistic terms:
  - *Low*: [0, 2].
  - *Medium*: [1, 2, 3].
  - *Strong*: [2, 5].

#### 9.1.2. Output Linguistic Variable of the Fuzzy Decision Model to Determine the Strategy

The fuzzy decision model produces one output variable that guides the system's actions regarding energy management. This output variable is defined as follows:

- **Strategy**—this variable ranges from 0 to 100 percent of energy used, and it determines which strategy to follow. It is described using three linguistic terms:
  - *LowEfficiency*: [0, 20].
  - *MediumEfficiency*: [0, 30, 50].
  - *StrongEfficiency*: [40, 100].

This output variable enables the system to make informed decisions based on the input variables and the fuzzy rules, ensuring optimal performance and energy efficiency.

#### 9.1.3. Fuzzy Rules

The fuzzy rules establish the decision-making framework of the system. Each rule outlines conditions based on the input variables and determines the corresponding output decision. Our system adheres to the following rules:

1. If *Wb* is *Low*, and *Traffic* is *Low*, then *strategy* is *LowEfficiency*.
2. If *Wb* is *Low*, and *Traffic* is *Strong*, then *Strategy* is *MediumEfficiency*.
3. If *Wb* is *Medium*, and *Traffic* is *Medium*, then *Strategy* is *MediumEfficiency*.
4. If *Wb* is *Medium*, and *Traffic* is *Strong*, then *Strategy* is *StrongEfficiency*.
5. If *Wb* is *Strong*, and *Traffic* is *Low*, then *Strategy* is *MediumEfficiency*.
6. If *Wb* is *Strong*, and *Traffic* is *Strong*, then *Strategy* is *StrongEfficiency*.

## 10. Numerical Simulation in Matlab

The simulation data are adapted from the application environment, from which a step of discretization is  $h = 2$  as a minimum movement of the 5 m. In addition, it is suitable for an airport environment using data provided by Alstef Group for Bagxone; for example the mass max is 100 kg, the maximum speed is 7.5 m/s, the average speed is 5 m/s, the maximum acceleration is 1.5 m/s<sup>2</sup>, and the maximum deceleration is =2 m/s<sup>2</sup>. In simulation for the three strategies, we defined the same values, except that the maximum acceleration and maximum deceleration was fixed to =1.25 m/s<sup>2</sup> for the application environment to include the concept of a minimum speed that is equal to 2.5 m/s.

Based on this knowledge, three strategies were generated, thanks to these three speed values: minimum, average, and maximum. Three strategies were developed to fit the

different input linguistic variables of the fuzzy decision model, as noted in the previous section. *Strategy1* represents the first input, “Waiting Baggages”, when the level is low. When the level is strong and the second input, “traffic”, is low, it is *Strategy2* that satisfies these conditions. *Strategy3* describes strong traffic. Figure 9 shows the path traveled for each of the strategies.

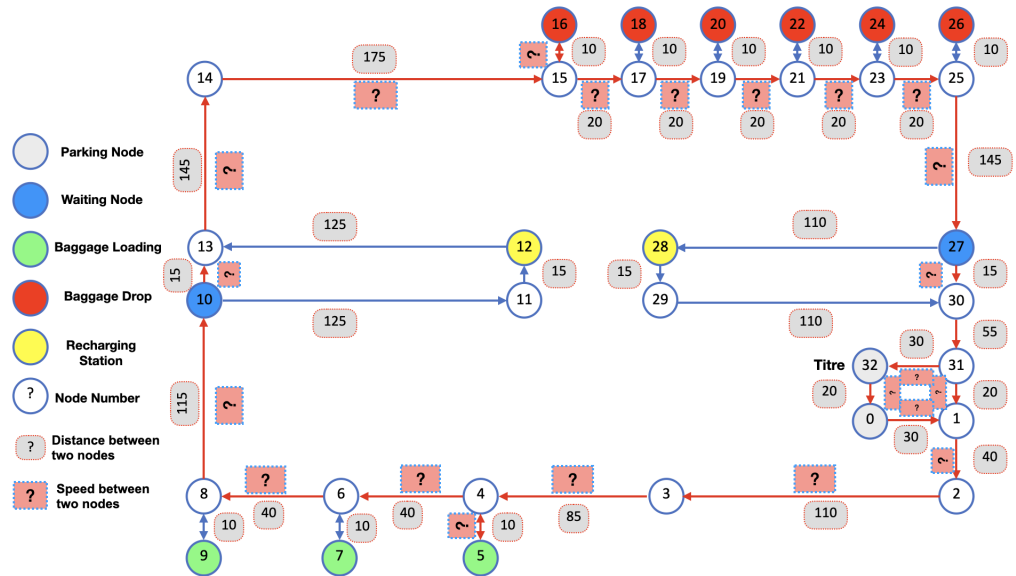


Figure 9. Map of strategies in red color: oriented graph nodes.

10.1. Strategy 1

Let the set *NodeDistance* represent the distance between two successive nodes. From this distance, the maximum speed values are defined for *Strategy1*. Figure 8 shows how we can obtain the value of the maximum speed, which is 2.5 m/s, for the path from *Node0* to *Node1* with distance of 30 m (cf.—Figure 10). *Strategy1* is defined according to these speed values:

$$\begin{cases} 5 \text{ m/s} & \text{if } \text{NodeDistance} \geq 100 \text{ m} \\ 2.5 \text{ m/s} & \text{Otherwise} \end{cases}$$

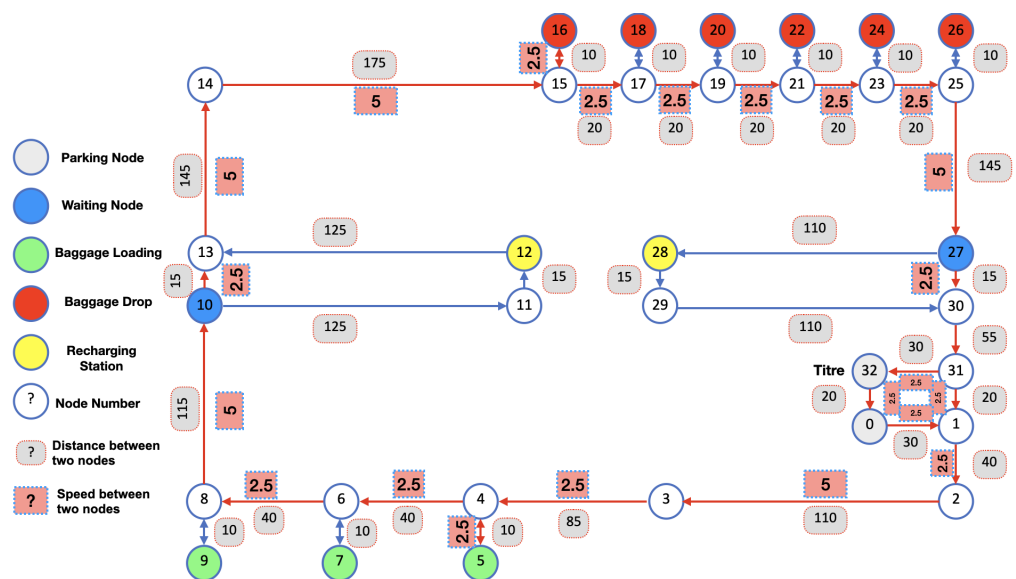


Figure 10. Map Strategy 1: oriented graph nodes.

### 10.2. Strategy 2

Unlike *Strategy1*, where speed values are fixed in relation to the distance between two nodes, *Strategy2* is arbitrarily defined with its maximum speed achieved. In *Strategy2*, high speed is the main speed used throughout the entire circuit (cf.—Figure 11). On the other hand, in *Strategy1*, the speed is limited to the average speed.

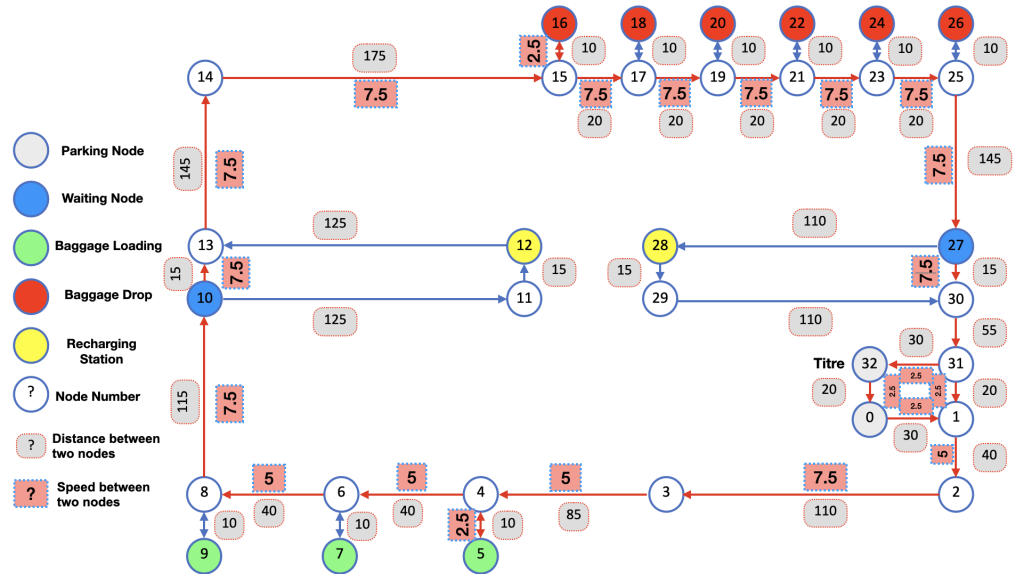


Figure 11. Map Strategy 2: oriented graph nodes.

### 10.3. Strategy 3

The *Strategy3* is approximately a mix of *Strategy1* and *Strategy2* (cf.—Figure 12).

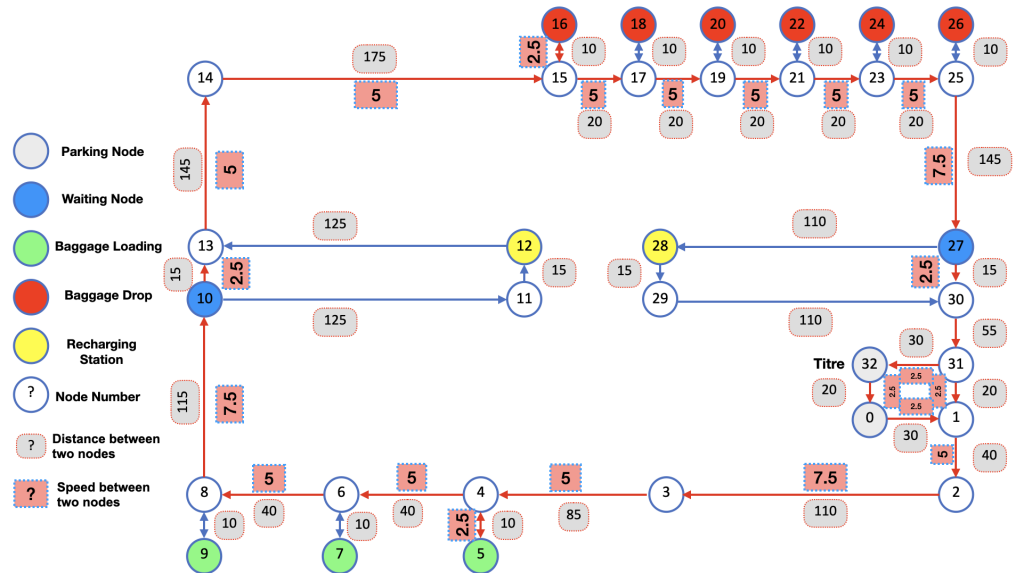


Figure 12. Map Strategy 3: oriented graph nodes.

### 10.4. Discussion

The path taken for the three strategies is indicated in red color, as shown in Figures 10–12 with the maximum speed value for each node.

A position–speed phase diagram is used to visualize the relationship between speed and position (distance traveled) for each strategy, as shown in Figures 13–15. Three dots with different colors (red, orange, green, and blue) are depicted in each phase diagram.



These dots will help illustrate a specific situation or task along the path. The red and orange dots indicate the nodes where the direction changes (i.e., when the movement changes from horizontal “Node-Axis-X” to vertical “Node-Axis-Y” or vice versa). Three green dots normally correspond to a node stop, first to load baggage, second to the dropping of the baggage, and the last to node “0” in parking. Two waiting nodes, noted as blue points, represent the two points waiting to enter battery charging. However, the energy consumed for each strategy is illustrated in Figures 16–18.

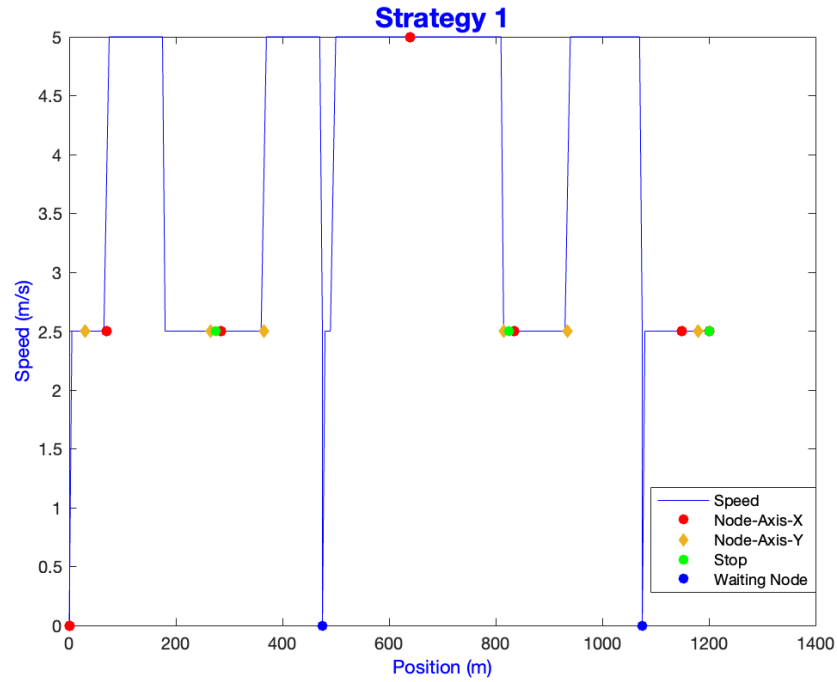


Figure 13. Position–speed phase diagram: Strategy1.

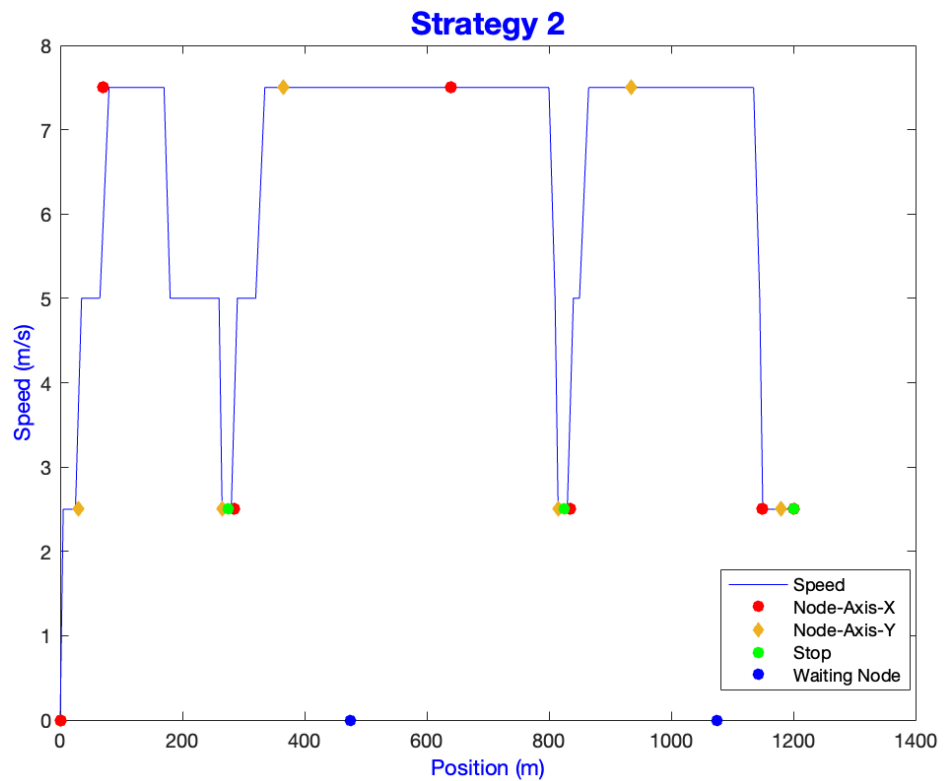


Figure 14. Position–speed phase diagram: Strategy2.

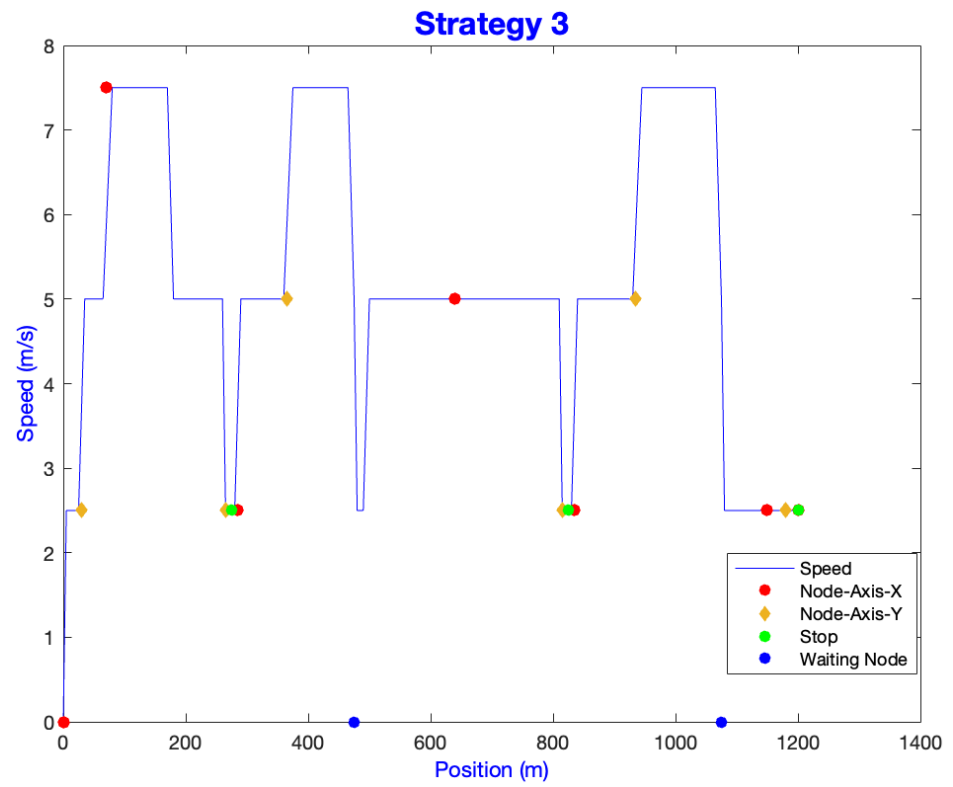


Figure 15. Position–speed phase diagram: Strategy3.

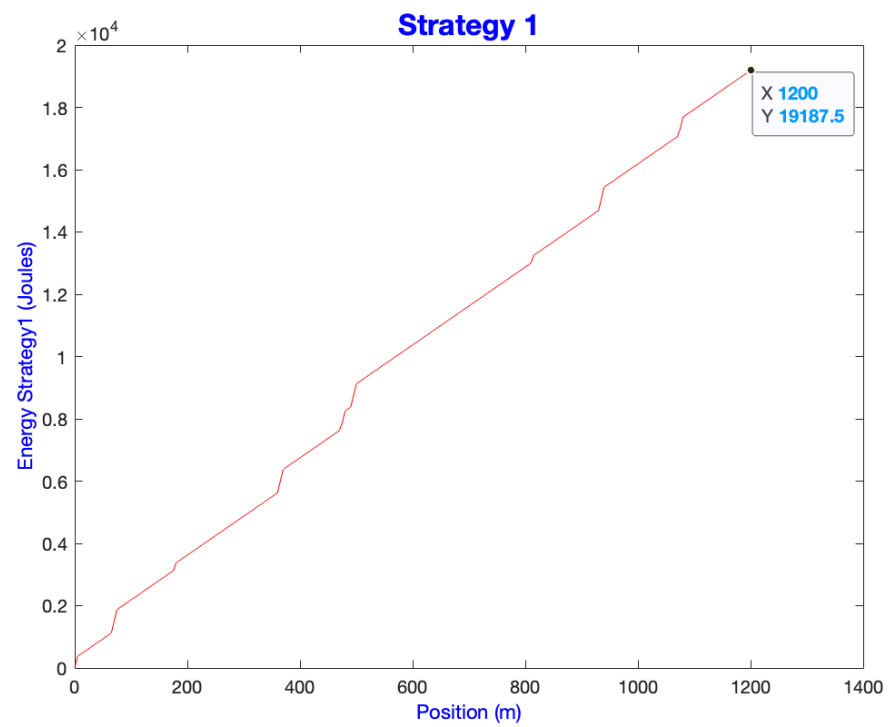


Figure 16. Energy consumed in Strategy1.

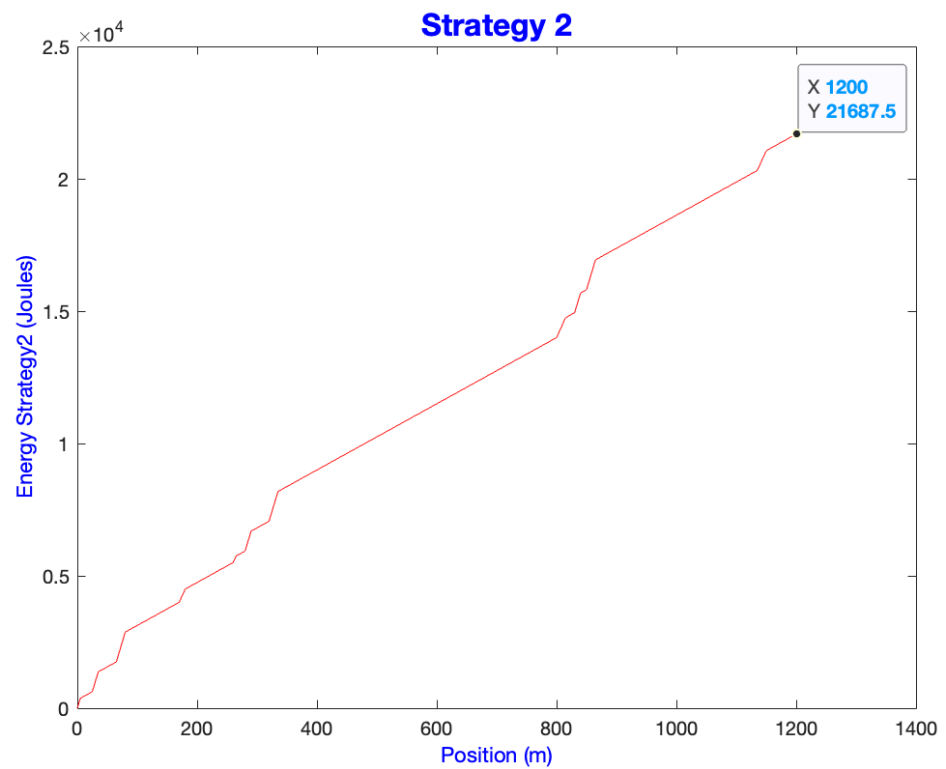


Figure 17. Energy consumed in Strategy2.

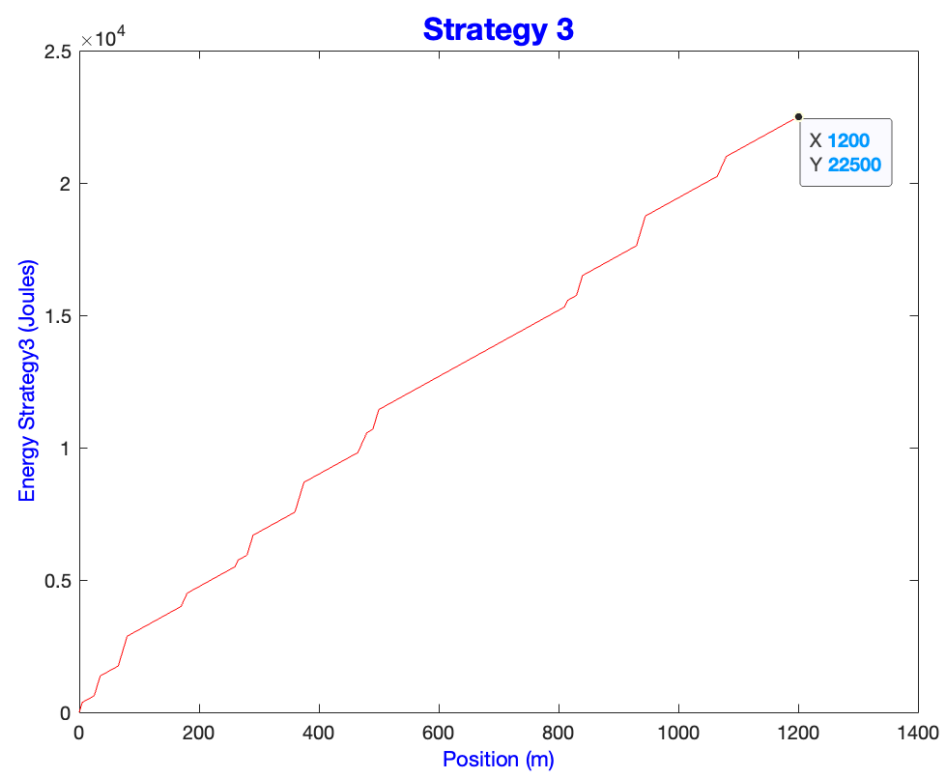


Figure 18. Energy consumed in Strategy3.

When the AIVs moves with a limited speed to the average and when the energy consumed is low, as Strategy1 (cf.—Figures 13 and 16), despite making two stops at this level waiting node. An abrupt variation in speed between the maximum and average involves high levels of energy consumption; this situation is represented by Strategy3 (cf.—Figures 15 and 18).

The average energy consumption is determined by maintaining the maximum speed along almost the entire length of the circuit, as shown in *Strategy2* (cf.—Figures 14 and 17).

During an AIV mission, two constraints are imposed, waiting baggage and traffic, which are considered inputs into our fuzzy logic model for strategy selection.

Three strategies generated via our methodology effectively reflect these constraints. The goal is to find an optimal compromise between energy consumption, baggage waiting times, and traffic conditions.

As shown using numerical simulations, *Strategy1* focuses on optimal energy consumption. *Strategy2* allows for a slight increase in energy usage to address waiting baggage. *Strategy3* prioritizes alleviating energy constraints related to traffic while addressing waiting baggage.

## 11. Conclusions

In this paper, we have proposed a multi-agent simulation, including fuzzy logic, to test various scenarios of battery recharging management. This approach offers a flexible adaptation to the various aspects of AIV management, and it facilitates any adjustments required for deployment at an industrial site. The use of a distributed system provides temporary autonomy in the event of a failure in central infrastructure, taking into account the individual differences in the battery capacity of the AIVs.

The simulation results demonstrate that incorporating adaptive fuzzy multi-agent models for AIV energy management can significantly optimize recharging strategies, improve operational efficiency, and mitigate energy consumption, particularly by considering dynamic factors such as workload variation, the communication between AIVs, and infrastructure elements. In fact, an infrastructure capable of optimizing recharging according to energy tariffs is advantageous, particularly with the ability to cut consumption over an hour. These findings will underscore the importance of flexible, collaborative approaches in enhancing the performance of autonomous systems in dynamic environments.

An optimal control problem was defined to introduce an accurate energy consumption model. PMP was used to calculate the maximum speed values for each node. Subsequently, an algorithm was implemented to plot a speed profile based on the distance traveled and the corresponding energy profile, which considers two distances: acceleration and deceleration. Using this algorithm, three strategies were generated. The strategy selected for each AIV was determined using a fuzzy logic model.

The first perspective for this part was to integrate the stopping phase into the algorithm for loading or unloading baggage. We can also consider stops for the AIV for other reasons, such as parking or during its journey. Then, we implemented the energy control in the “Airport baggage handling simulation”.

In this study, a constant resistance force was considered. From a second perspective, we will examine a force that depends on linear speed or another parameter adapted to the application’s environment. Additionally, we will develop strategies that better respond to other constraints, such as prioritizing fully loaded AIVs over empty ones and considering the battery level of an AIV. The final perspective is to refine the mathematical model, particularly the cost function, by considering more detailed models, for example, including motor efficiency to better reflect the AIV performance.

Finally, another important future work could involve conducting field trials in actual airport settings, which will be essential to validate the performance of the adaptive fuzzy multi-agent models. These trials will provide insights into how well the algorithms handle real-time operational challenges and battery management. Moreover, simulating a variety of real-world scenarios, such as peak travel periods, maintenance disruptions, and emergency situations, to test the robustness of the energy management strategies. This will help in understanding how the models perform under different operational stresses.

**Author Contributions:** Conceptualization, J.G., O.O. and A.-J.F.; methodology, J.G. and A.-J.F.; software, J.G. and A.-J.F.; validation, M.D.-K. and J.-M.B.; formal analysis, J.G., O.O. and A.-J.F.; investigation, J.G., O.O. and A.-J.F.; writing—original draft preparation, J.G. and A.-J.F.; writing—review and editing, J.G., O.O. and A.-J.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Brittany region for funding the VIASIC and ALPHA projects, respectively as part of the ARED-2021-2024 call for projects entitled “The economy at the service of industry for intelligent production” and the PME 2022 call for projects entitled “Accelerate time to market of digital technological innovations from SMEs in the Greater West”.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

AIV	Autonomous industrial vehicle
OCP	Optimal control problem
PMP	Pontryagin’s maximum principle

## References

- Hu, X.; Zeigler, B.P. A Simulation-Based Virtual Environment to Study Cooperative Robotic Systems. *Integr.-Comput.-Aided Eng.* **2005**, *12*, 353–367. [[CrossRef](#)]
- Tsolakis, N.; Bechtsis, D.; Srai, J.S. Intelligent Autonomous Vehicles in Digital Supply Chains: From Conceptualisation, to Simulation Modelling, to Real-World Operations. *Bus. Process Manag. J.* **2018**, *25*, 414–437. [[CrossRef](#)]
- Hentout, A.; Aouache, M.; Maoudj, A.; Akli, I. Human–Robot Interaction in Industrial Collaborative Robotics: A Literature Review of the Decade 2008–2017. *Adv. Robot.* **2019**, *33*, 764–799. [[CrossRef](#)]
- Jing, P.; Hu, H.; Zhan, F.; Chen, Y.; Shi, Y. Agent-Based Simulation of Autonomous Vehicles: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 79089–79103. [[CrossRef](#)]
- Kou, N.M.; Peng, C.; Yan, X.; Yang, Z.; Liu, H.; Zhou, K.; Zhao, H.; Zhu, L.; Xu, Y. Multi-Agent Path Planning with Non-constant Velocity Motion. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, Montreal, Canada, 13–17 May 2019.
- Grosset, J.; Ndao, A.; Fougères, A.J.; Djoko-Kouam, M.; Couturier, C.; Bonnin, J.M. A Cooperative Approach to Avoiding Obstacles and Collisions between Autonomous Industrial Vehicles in a Simulation Platform. *Integr.-Comput.-Aided Eng.* **2023**, *30*, 19–40. [[CrossRef](#)]
- Kopp, R.E. *Pontryagin Maximum Principle*; Elsevier: Amsterdam, The Netherlands, 1962; Volume 5, pp. 255–279.
- Gurguze, G.; Turkoglu, I. Energy Management Techniques in Mobile Robots. *Int. J. Energy Power Eng.* **2017**, *11*, 1085–1093.
- Nonoyama, K.; Liu, Z.; Fujiwara, T.; Alam, M.M.; Nishi, T. Energy-Efficient Robot Configuration and Motion Planning Using Genetic Algorithm and Particle Swarm Optimization. *Energies* **2022**, *15*, 2074. [[CrossRef](#)]
- Fougères, A.J. A Modelling Approach Based on Fuzzy Agents. *arXiv* **2013**, arXiv:1302.6442.
- Fougères, A.J.; Ostrosi, E. Fuzzy Agent-Based Approach for Consensual Design Synthesis in Product Configuration. *Integr. Comput.-Aided Eng.* **2013**, *20*, 259–274. [[CrossRef](#)]
- De Ryck, M.; Versteyhe, M.; Shariatmadar, K. Resource Management in Decentralized Industrial Automated Guided Vehicle Systems. *J. Manuf. Syst.* **2020**, *54*, 204–214. [[CrossRef](#)]
- Hong, T.; Nakhaeinia, D.; Karasfi, B. *Application of Fuzzy Logic in Mobile Robot Navigation*; Intech: Rijeka, Croatia, 2012. . [[CrossRef](#)]
- Pradhan, S.; Parhi, D.; Panda, A. Fuzzy Logic Techniques for Navigation of Several Mobile Robots. *Appl. Soft Comput.* **2009**, *9*, 290–304. [[CrossRef](#)]
- Yerubandi, V.; Reddy, Y.; Vinod, M.; Vinod Kumar Reddy, M. Navigation System for an Autonomous Robot Using Fuzzy Logic. *Int. J. Sci. Res. Publ. (IJSRP)* **2015**, *5*, 1–4.
- Yudha, H.; Dewi, T.; Hasana, N.; Risma, P.; Oktarina, Y.; Kartini, S. *Performance Comparison of Fuzzy Logic and Neural Network Design for Mobile Robot Navigation*; IEEE: Piscataway, NJ, USA, 2019; p. 84. [[CrossRef](#)]
- Meylani, A.; Handayani, A.; Ciksadan, Carlos, R.S.; Husni, N.L.; Nurmaini, S.; Yani, I. *Different Types of Fuzzy Logic in Obstacles Avoidance of Mobile Robot*; IEEE: Piscataway, NJ, USA, 2018; p. 100. [[CrossRef](#)]
- Shitsukane, A.; Cheriuyot, W.; Otieno, C.; Mgala, M. A Survey on Obstacles Avoidance Mobile Robot in Static Unknown Environment. *Int. J. Comput. (IJC)* **2018**, *28*, 160–173.
- Patle, B.; Loganathan, G.; Pandey, D.A.; Parhi, D.; Anne, J. A Review: On Path Planning Strategies for Navigation of Mobile Robot. *Def. Technol.* **2019**, *15*, 582–606. [[CrossRef](#)]

20. Nasrinahar, A.; Chuah, J.H. Intelligent Motion Planning of a Mobile Robot with Dynamic Obstacle Avoidance. *J. Veh. Routing Algorithms* **2018**, *1*, 89–104. [[CrossRef](#)]
21. Alakhras, M.; Oussalah, M.; Hussein, M. A Survey of Fuzzy Logic in Wireless Localization. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 89. [[CrossRef](#)]
22. Lee, M.F.R.; Nugroho, A. Intelligent Energy Management System for Mobile Robot. *Sustainability* **2022**, *14*, 10056. [[CrossRef](#)]
23. Ostrosi, E.; Fougères, A.J.; Ferney, M. Fuzzy Agents for Product Configuration in Collaborative and Distributed Design Process. *Appl. Soft Comput.* **2012**, *12*, 2091–2105. [[CrossRef](#)]
24. Ören, T.I.; Ghasem-Aghaee, N. Personality Representation Processable in Fuzzy Logic for Human Behavior Simulation. In Proceedings of the SCSC, Phoenix, AZ, USA, 1 August–31 July 2003; pp. 3–10.
25. Ostrosi, E.; Fougères, A.J.; Ferney, M.; Klein, D. A Fuzzy Configuration Multi-Agent Approach for Product Family Modelling in Conceptual Design. *J. Intell. Manuf.* **2012**, *23*, 2565–2586. [[CrossRef](#)]
26. de Lucca Siqueira, F.; Della Mea Plentz, P.; De Pieri, E.R. A Fuzzy Approach to the Autonomous Recharging Problem for Mobile Robots. In Proceedings of the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 13–15 August 2016; pp. 1065–1070. [[CrossRef](#)]
27. Morales, M.F.J.; Mendoza, J.B.G. Mixed energy model for a differential guide mobile robot. In Proceedings of the 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland, 27–30 August 2018; pp. 114–119.
28. Zhang, X.; Huang, Y.; Rong, Y.; Li, G.; Wang, H.; Liu, C. Optimal trajectory planning for wheeled mobile robots under localization uncertainty and energy efficiency constraints. *Sensors* **2021**, *21*, 335. [[CrossRef](#)]
29. Štefek, A.; Pham, V.T.; Krivanek, V.; Pham, K.L. Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot. *Appl. Sci.* **2021**, *11*, 6023. [[CrossRef](#)]
30. Gürgöze, G.; Türkoğlu, İ. A Novel, Energy-Efficient Smart Speed Adaptation Based on the Gini Coefficient in Autonomous Mobile Robots. *Electronics* **2022**, *11*, 2982. [[CrossRef](#)]
31. Tan, Z.; Lu, S.; Bao, K.; Zhang, S.; Wu, C.; Yang, J.; Xue, F. Adaptive partial train speed trajectory optimization. *Energies* **2018**, *11*, 3302. [[CrossRef](#)]
32. Brun, Y.; Di Marzo Serugendo, G.; Gacek, C.; Giese, H.; Kienle, H.; Litoiu, M.; Müller, H.; Pezzè, M.; Shaw, M. Engineering Self-Adaptive Systems through Feedback Loops. In *Software Engineering for Self-Adaptive Systems*; Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 48–70. [[CrossRef](#)]
33. Grosset, J.; Fougères, A.J.; Djoko-Kouam, M.; Bonnin, J.M. Fuzzy Agent-Based Simulation for Managing Battery Recharging for a Fleet of Autonomous Industrial Vehicles. In Proceedings of the ASPAI 2024: 6th International Conference on Advances in Signal Processing and Artificial Intelligence, Funchal (Madeira), Portugal, 8–10 April 2024.
34. Lasi, H.; Fettke, P.; Kemper, H.G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [[CrossRef](#)]
35. Oukacha, O.; Boizot, N. Consumption minimization for an academic model of a vehicle. *Optim. Control Appl. Methods* **2020**, *41*, 1288–1320. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.