IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 April 2024, accepted 29 May 2024, date of publication 3 June 2024, date of current version 14 June 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3409051*

SURVEY

# Multi-Agent Systems: A Survey About Its Components, Framework and Workflow

**DIEGO MALDONADO[1], EDISON CRUZ[2], JACKELINE ABAD TORRES[1], PATRICIO J. CRUZ[1], (Member, IEEE), AND SILVANA DEL PILAR GAMBOA BENITEZ[1]**
[1]Departamento de Automatización y Control Industrial, Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito 170525, Ecuador
[2]Department of Electrical, Computer and Biomedical Engineering, Southern Illinois University, Carbondale, IL 62901, USA

Corresponding author: Diego Maldonado (diego.maldonadoa@epn.edu.ec)

This work was supported by the Escuela Politécnica Nacional under Grant PIS-21-01.

**ABSTRACT** With the rapid technological advancements and the ever-evolving complex systems, the identification and integration of the components and resources for the functioning of multi-agent systems (MAS) are crucial tasks. However, difficulties arise due to the complexity of not having reference frameworks that normalize their implementation. Therefore, in this survey, we propose the FC-MAS (Framework-Components in Multi-Agent System) model as a conceptual framework designed to simplify comprehension and standardization in incorporating the required functions and components for the deployment and operation of MAS in engineering applications. This model comprises five abstract layers, each of which serves a specific purpose and encompasses the details and resources required to operate MAS. Furthermore, we propose a structured workflow for centralized and distributed MAS schemes with a set of related activities that integrate the fundamental steps and stages for the successful implementation of MAS. Finally, this work discusses potential directions for future research, including a deeper exploration of essential components, the establishment of terminology standards across various domains, and the refinement of the proposed model to enhance its applicability and relevance across a broader spectrum of contexts.

**INDEX TERMS** Multi-agent system, complex system, components, framework, workflow.

## I. INTRODUCTION

An overarching trend observed across various industries, including robotics and power systems, is the increasing integration of their components. This integration has been facilitated by the advent of transformative technologies like the Internet of Things (IoT), artificial intelligence (AI), machine learning (ML), Industry 4.0, and networked systems. [1], [2], [3]. Thus, their components become a fundamental part of an interactive environment capable of putting together different processes and coordinating its activities to meet more complex objectives and global tasks. In addition, these interconnected systems are able to use the available resources (either local or global) without the constant interaction with a human operator or a great amount of information exchange [4].

Systems such as those described above are identified as Multi-agent Systems (MAS), which are generally defined as a set of elements, called agents, interacting with each other to achieve a common purpose [5], [6], [7], [8], [9]. The design, development, and coordination of these systems is still an open challenge due to the complexity that they entail. Therefore, several works have focused on studying them in more depth. For instance, MAS is a type of complex system [10], so the theory, terms, and concepts from this active research topic can be applied to them.

Applications involving MAS are very diverse and in constant growth. There are several works about their application in areas such as: emergency response operations involving drones and mobile robots [11], smart grids or micro-grids [12], [13], [14], industrial plant control with actuator saturation [15], multi-vehicle coordination [16], data traffic in computer science, transportation and communication systems for smart cities [17], [18], among others. Due to the different

The associate editor coordinating the review of this manuscript and approving it for publication was Chao-Yang Chen.

fields of MAS's applications, the names of the involved elements and processes differ according to the taxonomy terms typically used in their field of study.

Another important point to highlight from the recent use of MAS is the possibility of implementing novel techniques that seek to optimize the operation of specific procedures of the system. For this reason, many studies focus on collecting relevant information about these techniques and applying them in specific processes within the operation of MAS. For example, [19] presents general knowledge and advances of a MAS process known as consensus from a control perspective. Related works about this process, such as [20], present some consensus algorithm alternatives, considering delays and the agent's dynamics, *e.g.* single and double integrator dynamics. Another critical activity carried out in MAS is decision-making. For instance, [7] states that each agent is an essential part of the MAS whose performance defines the benefit of the whole system and optimal sequences of actions. Other processes in MAS are task allocation and coalition formation. In this regard, there are algorithms based on demands, resources and profit objectives such as the one proposed in [21], while [22] presents a combination of a greedy algorithm with a distributed many-objective evolutionary algorithm to find the best solution. An additional feature that becomes crucial in MAS is the development of fault-tolerant behavior during its operation. Thus, different related works have focused on the design and the operation principles of robust and resilient MAS [23], [24].

Having all this information, it is convenient to have a general structure of a MAS, which could serve as a basis for its analysis and implementation. This type of study helps to: identify its elements, know its capabilities, and then apply the appropriate methods and techniques for improving its performance. This is the main motivation behind this work, which seeks to provide a general understanding of MAS based on its most important components, characteristics and analyze case studies developed from a more realistic perspective. Further, we seek to provide a workflow that describes the deployment and operation of a MAS.

The goal of this study is to propose a schematic that describes any general MAS, which leads to formulating the following research questions(RQs):

- RQ1: What taxonomy relates to the elements of a MAS?
- RQ2: What are the most common components within a MAS?
- RQ3: Is there a framework that categorizes the components and elements of a MAS?
- RQ4: Is there a specific workflow for implementing a MAS and putting it into operation?
- RQ5: How can a MAS begin operating effectively following a workflow from its design?

To address these questions, in conjunction with an exhaustive examination of the pertinent literature, allows us to present our main contributions. First, we propose a conceptual model, FC-MAS, which serves as a reference for comprehending and standardizing the amalgamation of diverse resources and components within multi-agent systems. This model abstracts the functions and components in a hierarchical five-layer structure: the physical network, synchronization, network controller, assessment, and fault tolerance. Subsequently, we provide illustrative instances of how this proposed model can effectively delineate various engineering applications, irrespective of their specific objectives. Second, we propose a structured workflow that offers a systematic insight into the required steps and stages that must be followed from the initialization of the operation of a multi-agent system.

The general structure of this survey is as follows. Section II presents an overview of the MAS information, the taxonomy with the most commonly used terms, and some relevant applications in which they stand out. Then, in Section III, the conceptual framework model proposed is analyzed and its elements are described. Continuously, section IV presents how FC-MAS can be used to describe the implementation in complex systems, using Multi-robot system (MRS) and Power Systems (PS) as examples. Section V details the MAS deployment process with the stages and steps to achieve a main objective. Finally, in Section VI the conclusions and future directions are discussed.

## II. MULTI-AGENT SYSTEMS (MAS)

### A. GENERAL DESCRIPTION

In general, a multi-agent system (MAS) is a set of so-called *agents* interacting among them within a certain environment, exchanging information through some communication channel [5], [25] and acting cooperatively among them to solve a common problem with great skill [26]. Further, this group of autonomous entities (agents) perceives information about the environment through sensors and carries out their activities by using actuators [6], [27].

We begin defining the most basic element in a MAS: an agent. Many authors have oriented the definition of this term according to the objectives of the application on which their work focuses, even so, many of these definitions resemble each other. Starting with concepts that address MAS in a general way, [28] defines an agent as: *"an entity which is placed in an environment and senses different parameters that are used to make a decision based on the goal of the entity. The entity performs the necessary action on the environment based on this decision"*. In [25], the general definition of an agent is: *"a computer system that is capable of independent actions on behalf of its user or owner. The agent can figure out for itself what it needs to do in order to satisfy its design objectives"*. In works focused on robotics, such as in [7], the authors propose a general concept for an agent, which is *"an autonomous entity capable of performing actions on its environment and perceiving its environment, aiming to accomplish a goal"*.

In the field of AI, [29] proposes the following: *"agents are autonomous, computational entities that can be viewed* *as perceiving their environment through sensors and acting upon their environment through effectors"*. Therefore, they refer to agents as either software or hardware with a certain level of intelligence that depends on its own experience for achieving goals. Another field to take into account for defining agents is Autonomous agents. In this area, [30] defines an agent as: *"a computational system that is located in some environment and is capable of performing autonomous actions in order to achieve objectives. These agents take input from the environment towards the sensor and produce output actions"*.

Regarding the *goals* in MAS correspond to the objectives pursued by the agents within the system and can be either local or global. Global goals (or objectives) refer to the overall goal or main purpose to be achieved or accomplished, which is usually the fundamental reason for developing or implementing the system. Conversely, each agent pursues local goals according to their assigned tasks, and decisions are made based on their objectives and locally available information. Even though each agent may only focus in its individual goal, it impacts the system performance and supports to reach the global goal. For example, in a traffic management system, a vehicle's individual goal is to reach its destination as quickly as possible, while the system's global goal is to minimize traffic congestion and reduce overall travel times.

Other relevant element constantly mentioned in the definitions of MAS is the *environment* in which the agents interact with each other by executing their programming in pursuit of a common goal. Due to the interaction between the agents and the environment, the former may be able to alter the environment. In order to modify the environment, the agent must be able to completely or partially observe it by gathering information via sensors, data prediction or computing systems. Moreover, this environment control, which is either physical or logical action, is clearly limited by the application, *e.g*, changing the environment implies the reconfiguration of a transmission network in the electricity sector [31], or changing transportation routes in logistics.

On the other hand, the *interactions* among the agents refer to the way in which they share information with each other to achieve cooperative work within the environment. This can occur through communication channels or physical links that couple the agents, known as the Agent's Data Exchange (ADE). Figure 1 presents a general architecture of an agent that fits to the previous definitions and provides a clear idea of how this element works and interacts with the environment.

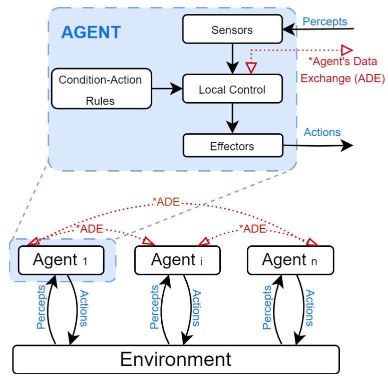


**FIGURE 1. General architecture of an agent interacting in an environment. *The Agent's Data Exchange (ADE) depends on the physical links, topology, and architecture used in the network.**

### B. TAXONOMY OF SOME COMMON TERMS IN MAS

Generally, MAS uses a large number of different terms corresponding to methods, features, or elements involved in certain areas of the system's operation. [32] provides an overview of agent terms and then delves into the key concepts and theories of MAS technology, touching upon related concepts like expert systems. Table 1 shows a compilation of the most used terminology in this type of work. The selected terms have been classified as a function of their main characteristics. Then sub-terms are associated with four major areas in engineering in which there are widely used: Control Systems, Power Systems, Telecommunication, and Robotics; here diverse MAS with their schemes, control architecture, and communication technology are applied. Through this relationship, it is possible to notice how MAS has been analyzed in various fields of research. Therefore, its application can be potentially generalized in other fields where they have not been studied yet.

#### 1) AGENT

The previously reviewed definitions of an agent share similar ideas about what it should be and do. An autonomous agent is capable of perceiving its environment, processing information, making decisions based on its objectives, taking actions without direct external control, and can have reactive, proactive and social behaviors [24], [25], [28], [33]. A *reactive agent* reacts immediately as a direct response to stimuli or changes in the environment without carrying out a deep analysis of the situation. A *proactive agent* takes the initiative in making decisions to achieve its objectives. It not only responds to environmental stimuli but also predicts future situations and can plan actions without direct intervention from other agents. A *social agent* interacts within the system to exchange data with its neighbors, seeking to cooperate, negotiate, or even compete. This type of agent recognizes that interaction with other agents is sometimes the best way to acquire knowledge. For instance, in a traffic management system, an autonomous agent can take the form of a self-driving vehicle equipped with sensors and advanced control algorithms. This allows the vehicle to interpret traffic signals and make decisions such as changing lanes or stopping at intersections without requiring direct communication with other vehicles. A reactive agent may be a traffic light control that adjusts signals based on predetermined timing patterns. It operates without making predictions about long-term traffic flow. In contrast, a proactive agent is a vehicle that predicts potential congestion based on historical data, weather forecasts, and special events. It can also suggest alternative routes to other vehicles. Lastly, social agents are vehicles that collectively exchange data from their position, speed, and intentions on the road. They can also announce traffic events, allowing other vehicles to plan their route. This optimizes traffic flow, mitigates the risk of collisions, and enhances safety in general.

#### 2) ORGANIZATION

Organization in a MAS refers to how agents interact based on their roles, behavioral expectations, and authority relationships. Some related terms include *centralized* [34], where agents send information to a control center that has the monopoly to make decisions, and *decentralized* [35], where many distributed controllers work locally. *Hierarchy* refers to decision-making authority [9], which can be distributed to raise fault tolerance. *Homogeneous systems* have agents with identical characteristics, while *heterogeneous systems* have agents with different characteristics that complement each other. *Holonic systems* are composed of holons, which are a set of agents grouped according to certain features and communicate with other agents in the same or different holons of the same level.

#### 3) ALGORITHMS IN MAS

Control algorithms or schemes are essential for managing the elements and ensuring the correct operation of a MAS. These algorithms can serve as the primary control for system operation or as a secondary algorithm for performing additional tasks that enhance overall performance. Some common algorithms used in MAS include *consensus algorithms* [36], which seek to reach a mutual agreement with all system elements through a continuous exchange of information. *Artificial Intelligence (AI)* [37], [38] is another critical area with broad applications, involving the ability of a system to interpret data correctly, learn from it, and use that knowledge to achieve specific goals. *Swarm intelligence algorithms* [39], inspired by group behavior in the animal kingdom, are often used in task assignments for MAS. *Game theory* [40], specifically in control theory and distributed control, is oriented towards designing games that evoke desirable emergent behavior of the collective subsystems. Finally, *Machine Learning algorithms* [41], refers to the capability of computer programs to identify intricate patterns by utilizing algorithms that analyze data. By constructing mathematical models of problems, these algorithms enable the prediction of future behaviors.

#### 4) COMMUNICATION

Effective communication is crucial for a MAS to function properly, as it involves information exchange and interactions among agents. Three key factors that impact communication in MAS are the agent communication language, transmission frequency, and delays [9], [42]. To ensure that agents with different hardware or software can interact and exchange information correctly, a common agreement is required in the choice of *communication language*. Additionally, the data *transmission frequency* must be synchronized to avoid information overflow and ensure that the receiver is available to process the information. However, factors such as the transmission medium, the amount of information, and the environment can cause *delays in communication*.

**TABLE 1. Compilation of terminology most used in MAS and how is it used in different research areas.**

| Term | Subterm | Control System | Power System | Tele-commu nications | Robotics |
|---|---|---|---|---|---|
| Agents | Autonomous | | ✓ | | |
| | Pro-active | | ✓ | | |
| | Reactive | ✓ | ✓ | | ✓ |
| | Social | | ✓ | | |
| Algorithms in MAS | Artificial intelligence | ✓ | ✓ | | ✓ |
| | Consensus | ✓ | ✓ | ✓ | ✓ |
| | Game theory | ✓ | ✓ | ✓ | ✓ |
| | Machine Learning | | | | ✓ |
| | Swarm intelligence | ✓ | ✓ | ✓ | ✓ |
| Communication | Agent communication language | | ✓ | ✓ | ✓ |
| | Delay | ✓ | ✓ | ✓ | ✓ |
| | Transmission Fequency | | ✓ | | |
| Organization | Centralized | ✓ | ✓ | ✓ | ✓ |
| | Descentralized | ✓ | ✓ | ✓ | ✓ |
| | Heterogeneity | ✓ | ✓ | ✓ | ✓ |
| | Hierarchical | ✓ | ✓ | ✓ | ✓ |
| | Holonic | ✓ | ✓ | ✓ | ✓ |
| | Homogeneity | | | | ✓ |
| Simulation Enviorments | Gama | | | ✓ | |
| | Matlab | ✓ | ✓ | ✓ | ✓ |
| | Ros | ✓ | | | ✓ |
| | Jade | | ✓ | | |

#### 5) SIMULATION ENVIRONMENTS

Simulation environments are essential for developing and testing MAS. They offer tools for modeling, building, and simulating agent behavior and interactions in a controlled virtual environment. These environments are valuable for comprehending, analyzing, and validating MAS designs before real-world implementation. For instance, *Java Agent DEvelopment (JADE)* [43], a platform for building multi-agent systems in Java provides a set of tools and libraries that enable the development of distributed and intelligent software based on the concept of agents. *MATLAB* is also a popular platform for developing engineering applications, which offers a variety of toolboxes and functions that are helpful for designing, implementing, and analyzing the behavior of individual agents and their interactions in a MAS. *Robot Operating System (ROS)*, an open-source framework widely used to control robotic systems with realistic environmental characteristics. *GIS and Agent-based Modelling Architecture (GAMA)* [44], a platform for building spatially explicit agent-based simulations with the capacity to manage geographical data and carry out simulations with hundreds of thousands of agents. These simulation environments allow researchers to test and refine MAS models, ultimately leading to more effective and efficient systems.

### C. APPLICATIONS

As mentioned in Section I, many applications have included the MAS concept to model complex systems and facilitate a solution by dividing the global problem or objective into a series of individual tasks that can be performed more easily. Below, we mention some applications that use MAS models in the resolution and optimization of real problems. Among the applications, we have not included PS or multi-robot systems (MRS) because they will be analyzed in Section IV.

#### 1) SMART TRANSPORT SYSTEM

The current development of some urban areas have increased the expectations about services, infrastructure, and mobility among other aspects in the cities. Talking specifically of mobility and daily transportation of people, some research works have used the application of MAS in order to give innovative solutions to this problem.

For instance, [45] shows the adaptation of a MAS to an intelligent transport system for better management of traffic in the streets using swarm algorithms. The intention is to increase the quality of the entire transport network, avoiding congestion due to many cars on the same routes, organizing new routes and itineraries considering real traffic data, optimizing in this way travel time. With the same purpose, [46] developed another application, which uses the data obtained from a network sensor installed in vehicles and roads to get accurate data, create the traffic scenario in specific places, and then use this information in the MAS for traffic light management.

#### 2) SENSOR NETWORKS

They include a series of sensors and actuators that are responsible for collecting data for further processing and providing useful information for a monitoring application. In recent years and with current technological advances, the implementations of these sensor networks are more affordable. Thus, favoring the implementation of applications where a network of multiple elements is required [47]. These sensor networks are widely used in different applications such as: monitoring and security for highly important sites, caring for animals to detect patterns of their behavior, controlling efficient consumption in industrial and residential facilities, monitoring the environmental behavior of volcanoes, ecological reserves and the sea, among others [48].

#### 3) EMERGENCY RESPONSE ACTIVITIES

Natural disasters are phenomena that seriously affect people's lives every time they occur. The devastating effects after their appearance such as fires, floods, collapse of buildings, among others, cause that any rescue operation to be high risk and sometimes it is not efficient enough to help all the people involved in the accidents. In this type of situation, the use of MAS, applied to sets of robots, becomes a solution that allows improving search and rescue activities in emergency situations to help people as quickly as possible, without putting in risk to third parties. For example, [49] simulates multi-agent terrestrial and aerial robotic systems for earthquake rescues in ROS environment. It applies additional extensions for processes related to the selection of trajectories and also using specific algorithms for internal process like task allocation. Another example is the protection action against PS failure events, where a coordinated set of remote units disconnect sections of the transmission network, isolating the failure from the rest of the elements of interest [14]

#### 4) EPIDEMIOLOGY

Health care is an area in which the application of MAS becomes useful. In specific studies such as epidemiology, which among other objectives is responsible for the analysis of the spread of contagious diseases, the inclusion of MAS has made it possible to improve the analysis processes, create models through which health personnel can analyze different scenes of disease contagions and thus provide effective solutions. Taking current situations such as the COVID-19 pandemic, [50] mentions some situations in the health area in which MAS can be useful, like the correct handling of the data related to the health status of the patients, which would be used for a correct allocation of resources. In [51] considers decision-making according to the patient's condition and data analysis of remote patient monitoring. Other works, such as [52] focus more on the analysis of the model that represents the spread of the virus, and thus study its behavior with different variables involved.

## III. FC-MAS MODEL

To show a general structure in which different MAS can be described, we propose a framework based on five layers containing main components that could be both essential and complementary elements, functions, and/or properties (Table. 2). Some authors have proposed similar schemes to analyze MAS from their subject of study perspective; for instance, [29] presents agents, interactions and the environment as principal attributes of MAS, and [33] proposes a terminology of subjects of study for MAS, but oriented to the study of PS.

In this case, as one of the main contributions of this paper, we present the FC-MAS (Framework-Components in Multi-Agent System) model as a framework that will allow us to understand the integration of components of the MAS used in the applications according to their characteristics. The layers are hierarchically organized, with this purpose, the physical network and the fault tolerance are defined as the first and last layer respectively.

It is worth clarifying that the proposed model tries to cover all the possible applications and configurations of the MAS, so the order and use of the components will depend on the application itself or the problem to be solved. Fig. 2 shows each proposed layer with its components in detail. Since some basic applications only require fundamental components, while other complex applications integrate high-level components for greater self-regulation capacity, they (marked with an asterisk) are not fundamental and their

**TABLE 2.** Description of Layers and components of the FC-MAS model. Items with an asterisk are optional and their use will depend on their application.

| Layer | Components |
|---|---|
| **Physical network** Elements intervening in the infrastructure and the intecraction between them. | **Node:** Entities interacting in the environment. Typically they are known as an agent. |
| | **Links**: Interaction between nodes. |
| | **Topology:** Physical or logical configuration that describes the interaction and connectivity among all nodes. |
| **Synchronization** Allow the system to achieve a coupling on some variable or objective. | **Information exchanged:** Data, variables, or states exchanged among nodes |
| | ***Coalition:** Formation of teams based on the skills |
| | **Task allocation:** Assigning tasks to all nodes |
| | ***Consensus:** Agreement consented to and consummated by the nodes for the convergence of tasks |
| **Network controller** Control techniques are applied to reach an agreement and archive the objective proposed. | ***Estimation**: Determine the state of the system based on math models. |
| | **Monitoring**: Collecting data, processing, and analyzing it to control nodes |
| | **Control network algorithm**: Control actions apply to the system to fulfill the purposed tasks. |
| | **Control architecture:** Centralized or distributed control actions. |
| **Assessment** Supervision of the system | **Data analizing:** Useful information that can be interpreted. |
| | **Monitoring objective:** Comparison of the current system's state with respect to the defined goal. |
| | **Deviation detection:** Determine the error together with the causes that triggered it. A diagnostic is run. |
| | **Decision making:** Provide a solution and correct the deficiencies found |
| **Fault tolerance** Actions to maintain the system operating | **Robustness:** Maintain system functionality even in the presence of uncertainties, disturbances, or unexpected events. |
| | ***Fault detection**: Detect and act in the event of conditions outside the system tolerance range. |
| | ***Resilience:** Provide an acceptable level of service. Ability to return to equilibrium. |

use depends on the application. Each of the layers together with their respective components is described below.

## A. PHYSICAL NETWORK

This layer focuses on the elements intervening in the infrastructure of the system and the interaction between them.

### 1) NODE

The term node is often used in the field of complex networks to represent their more basic entities [53].

Considering the definition of agent proposed in the previous section, there is a certain similarity between agent and node, the latter being a broader term that allows a symbolic representation of an entity operating in the MAS. Further, the term node represents a greater number of elements in many applications [10].

The term node acquires different interpretations depending on the area of study, like: computers (Computer Science), cells (Biology), stations (Transportation Systems), mobile robots (Robotics), and generators (Power Systems), just to mention a few examples [54]. In the technological field, specifically in robotics, the term node commonly represents unmanned ground vehicles (UGV) or unmanned aerial vehicles (UAV), which are in charge of a task like: mapping, transporting, searching, etc [55]. In PS, a node could represent a group of generators, an agent that controls and supervises a set of generators, or a set of components necessary for producing energy that includes the generator, governor, sensors, and actuators, among the most important [5], [56].

### 2) LINKS

Even though a node can be a system itself, it also correlates with its peers through the called links. In general, a link represents the interactions between two nodes [57]. One of the strengths of a MAS is the constant interaction between its elements, so suitable management of the links can improve the performance of the system. Hence, in some applications, the use of links increases the scope of the sensory capabilities of its agents [58].

In a MAS, or complex systems or networks, a link could have a binary or a complex nature. In the former case, the link represents a connection or relationship between nodes, to mention a few examples: a physical communication channel, a bridge, a road, a pipe, or cables that connect the nodes. Meanwhile, in the latter case, the link carries a numerical value that quantifies the strength of the interaction or gives a measure about some features of the connection, some examples may be the bandwidth of the communication channel, data traffic with system information, or sending and receiving states between agents, etc. [59], [60]. Graph theory is widely used to describe and explain the interaction between nodes through their links.

### 3) TOPOLOGY

This component refers to the system connectivity, which explains how nodes interact with each other, or what information is available for each node [28], [61]. The topology is generally known as the configuration of a system. It can be static or fixed, established from the beginning, and remain unchanged throughout its life cycle. On the other hand, some systems have a dynamic topology, also known as switching topology, whose configuration changes as the system evolves or accordingly to its requirements [62]. More importantly, the topology helps to determine some dynamical features of the system and has a critical relevance in its robustness [63]. There are several topology alternatives, such as strongly connected, bus, mesh, star, tree, ring, and mixed.

### B. SYNCHRONIZATION

Synchronization occurs when the elements of a system have the same or common behavior over time, which could be the result of internal interactions or also the induction of external coordination forces [59]. This layer focuses on the elements that intervene on the synchronization so that the system reaches a coupling on some variables or objectives.

#### 1) INFORMATION EXCHANGED

The flow of information between agents through the links is the key to reach a common agreement. So the nodes involved in the system are responsible for sensing and processing the information corresponding to the environment in which they operate. This information is shared either among all, or their closest, agents to establish a synchronization for cooperation and achieve a multi-agent objective. This information has different nature, origin, and representation depending on the specific applications. For example, in fire emergency scenarios related to search and rescue operation [11], the agents share information such as victim status, victim location, fire status, fire location, and available resources, among others as numerical, Boolean, or arrays variables. In multi-robot SLAM [64], the robots either exchange positions obtained from odometry or a laser sensor, or a relevant area of a map obtained from the local measurements.

#### 2) COALITION

Due to the limited resources and capabilities of a single node, it can not complete a task by itself, or it is forced to reduce its performance. To avoid this situation, the nodes not only exchange information but also form *coalitions* that are composed of a group of cooperative nodes, which have some abilities or resources that help to solve a task. [65]. A coalition can be defined as an alliance between agents when forming teams that combine their resources and forces to meet a common goal.

In this component, nodes should generally have access to all the information and resources necessary to calculate the conditions for optimal action. However, there can be resource and communication constraints between them that affect the formation of groups [27].

Generally, the number of nodes, the task requirements, and the agent's capabilities determine the number of possible shapes of coalitions in the MAS. All these parameters must be considered and evaluated to find the best coalition shape. The various possibilities for coalitions are exponentially related to the number of nodes that the system has and the number of tasks to be accomplished. Thus, the coalition formation process consumes time and resources of the system [66]. To avoid a bottleneck in this process, there are seeveral methods such as Particle Swarm Optimization (PSO), game-based methodology, and genetic and evolutionary algorithms [67], [68], [69].

#### 3) TASK ALLOCATION

Nodes or coalitions in MAS have a variety of capabilities that allow the system to achieve its objectives. So that all nodes/coalitions contribute to achieving the system's goal, various tasks are defined and distributed among them. The former process is known as task decomposition, which consists of dividing a complex task (or objective) into simpler sub-tasks such that they can be carried out by a single node or a coalition [70]. The later process, known as task allocation, assigns the sub-task to the nodes and coalitions according to their capabilities to maximize and optimize the overall performance of the system and quickly fulfill the complex task with the least amount of resources [22], [71]. This process can be carried out several times while the system is running. In the event of a new task or if a node presents problems, a new task allocation will allow the system to restructure the distribution of sub-tasks based on the new requirements.

According to [22] the main goals of employing task allocation are:

- Maximizing the number of successfully execute tasks
- Maximizing the benefits of executing tasks
- Minimizing the resources consumed in executing tasks
- Minimizing the maximum time spent by agents to execute tasks

Also, as highlighted in many of the studies, the main elements involved in task allocation are: nodes, available resources in each node (i.e. energy, sensors, etc), task requirements, sequence of task execution, and constraint conditions (i.e. time or resources). For task allocation, many methods have been studied in order to improve the performance of the system in specific situations including: Ant Colony Optimization algorithm (ACO), PSO, contract network protocol, auction algorithm, cost matrices, or negotiation according to the availability of the agent [13], [21], [71].

#### 4) CONSENSUS

The convergence of the actions carried out by the nodes is the product of having consented to and consummated an agreement between them and is generally known as consensus [72]. Within a MAS, consensus can be established as a law where interaction rules are defined for the exchange of information between a node with its network neighbors with whom it can interact. Its implementation difficulty depends on the application requirements and their solution for convergence to agreement is generally the key to distributed coordination [73]. For example, the finite-time consensus issue can be solved locally in a MAS conformed by $n$ agents. This resolution is feasible provided that a certain condition, denoted as $\nu$, is met. This stipulates that for any initial state $x_i(0)$, located within the neighborhood defined by $|x_i(0) - x_j(0)| < \nu$, there exists a predetermined time interval

$T > 0$ that satisfies the requirement [72].

$$\lim_{t \to T} \left\| x_i(0) - x_j(0) \right\| < \nu \tag{1}$$

Additionally, consensus can occur naturally as the result of the coupling of the internal forces of the MAS itself, or also be the effect of external manipulation through external orders (*forced consensus*). The second one is a highly relevant issue in multi-agent systems [26], and it requires the appropriate design of a control input that allows several nodes to converge to a common agreement [73].

### *C. NETWORK CONTROLLER*

To coordinate the nodes and get a good system performance, the MAS requires a controller other than node's local one, or at least to coordinate the later ones. Components involved in this layer work together in the system to determine a control signal for each node focusing its behavior on the global objective.

#### 1) ESTIMATION

The good performance of the MAS is largely due to the correct determination of control orders. This, in turn, depends on the nodes knowing information about their neighbors or at least having an unbiased estimate of the unknown states of the system. [74]. The challenge lies in the fact that, in many complex systems, obtaining comprehensive and reliable information about their states and the environment is often unattainable due to factors like their substantial scale and the absence of sensors covering the entire system. Consequently, accurately gauging system performance and predicting when a node can complete its designated task becomes a demanding endeavor.

Hence, it is necessary to have tools with the capacity to find an approximate value of the inaccessible states of the MAS. Here, the estimation process determines suitable values of the states of the MAS based on math models and analyzing indirect measurements of a small fraction of the previously sensed variables. Some steps to estimate the states are: define the state variables, collect sensor data, determine the system model that describes how the state variables change over time, and evaluate the results [75].

#### 2) MONITORING

Task compliance inspection by system nodes is essential to achieve the global objective. For that reason, it is necessary that each node collects information, analyzes it, and tracks the progress of the performed activities, achieving in this way local monitoring, which is carried out at the level of each agent. In the local monitoring, each node needs to establish appropriate indicators like health status, autonomy, events, alarms, percentage of the task completed, etc. All of this is for progress measurement on assigned tasks and providing information on changes over time about the node. Considering that the purpose is to prevent nodes from behaving improperly by ensuring that local actions do not lead to undesired behaviors.

#### 3) CONTROL NETWORK ALGORITHM

This component is used to regulate the behavior of the nodes and hence the whole system through the manipulation of the control variables, typically using controllers. Currently, different investigations propose options for controllers that work at the network level. These control strategies use information from the entire system to generate the appropriate control signals in each node and thus fulfill its tasks aimed at the global objective. Some approaches propose algorithms based on common control techniques, such it is the case of [76], which uses a common PID control scheme applied to a leader-follower multi-agent system, while others use more complex controllers as a sliding mode control (SMC) or a fuzzy control [77].

#### 4) CONTROL ARCHITECTURE

The architecture of a system is often related to its topology because the connectivity and the nodes intervene in both concepts. However, the architecture focuses on how the control devices are integrated for the operation of the system. This term also includes how the control actions, control signals, or local references are assigned to these elements.

Two architectures are commonly studied and implemented in different types of systems: centralized and distributed architectures. When referring to the centralized case, the participation of a single element is taken into account as the main entity that defines control actions of all the rest of the nodes in the system [34]. Meanwhile, in a distributed architecture the entities are capable of defining control actions with a certain level of autonomy without depending on a central element. According to his philosophy, a decentralized architecture can be considered part of a distributed one. The difference is that in a decentralized architecture, the nodes make their decisions based on local information, while in a distributed architecture there is a negotiation with information between nodes being a feature that makes it more efficient.

Several works focus on analyzing and implementing the most convenient architecture for a given system. [78] analyzed the advantages and disadvantages of centralized and distributed configurations applied to the case of servers. While [17] studies the field of smart cities and uses a hybrid configuration described as centralized locally and distributed globally. In this case, the idea of integrating these architectures is to take advantage of the centralized hierarchy together with the distributed expansion versatility.

### *D. ASSESSMENT*

In this evaluation layer, the system seeks to reach a new level of interaction in which the user intervenes and it is possible to assess the functioning of the system.

#### 1) DATA ANALYZING

The interaction of a system with its environment involves working with a large amount of data, obtained by the nodes' sensors, and later transforming it into conclusive information, known as data analysis, that helps to improve the system by: solving or explaining problems, predicting anomalies or in general providing useful information during the operation of it. As in any data management application, there might be cases of bad measurements or missing data, so the result after the analysis is inconsistent, which causes additional problems. For this reason, the methods used in this process must be chosen appropriately according to several factors such as the type of data, sampling rate, resolution, and system resources, among others [79], [80], [81].

#### 2) MONITORING OBJECTIVE

The monitoring process focuses on the current system's state with respect to the defined goal in the MAS. The MAS's objectives are the axis that delineates the system's operation, and any failure in their specification of leads to its malfunction [82]. Therefore, a process is required to monitor the use of resources and the general operation of the system while reaching the main system goal. Some works do not delve into the definition of monitoring, since it is understood. However, in [83] a clear definition is presented that can be applied to a system with a main goal: monitoring is *"a process that fully and precisely identifies the root cause of an event by capturing the correct information at the right time and at the lowest cost in order to determine the state of a system and to surface the status in a timely and meaningful manner"*. Monitoring tools have been widely used for tracking resource utilization and the performance of systems and networks.

#### 3) DEVIATION DETECTION

The term deviation is usually seen as outliers, errors, or even noise that come from the data collection carried out by sensors or other elements in the system [84]. The detection of deviations arises from the need to compare the measurements with respect to nominal values and detect which data is not consistent or does not comply with the initially established criteria [85]. In short, deviation detection attempts to determine the error together with the causes that triggered it. Then, a diagnosis is made that determines an inconsistency in the obtained information, which can be one of the first lines of defense to detect erroneous data. Also, it provides information about a potential system's operational fault profile. In this fashion, it is possible to recommend corrective actions to solve deficiencies in the system. System diagnosis is often based on user experience and lately, it has become automated using techniques such as information processing, data mining, and machine learning that provide fast and accurate diagnoses [86].

#### 4) DECISION-MAKING

The decision-making component is the final process carried out in the system in which the obtained information, optimization algorithms, and even the users' experience are used to generate a sequence of actions that will achieve the general objective of the system. This provides a solution and corrects the deficiencies found [7]. In this component, after evaluating whether or not the MAS is aligned concerning its objective, the consecutive actions and tasks to be carried out based on its current situation are defined. The result of this can be both typical orders if the MAS is behaving as expected, or they can also be orders for the total restructuring of the tasks of each agent if the MAS has deviated from its objective and can no longer perform it.

Decision-making is considered a difficult task due to the existing uncertainties related to dynamic factors in the environment where the system is located. These unknowns can come also from the actions that each agent performs to meet objectives at a local level [27]. This has been the motivation for developing various methods applied for decision making, such as game theory, reinforcement learning, swarm intelligence, and evolutionary computing among others, which seek to optimize system resources to the maximum [70].

### *E. FAULT TOLERANCE*

This is one ability of the system to continue functioning correctly and effectively, even in the presence of faults, errors, abnormal behavior, or failures in one or more of its agents or components. For use, several factors must be considered in its design so that can withstand failures of different kinds such as limited agent autonomy, failure and loss of communication between agents, changing or inaccessible environments, limited energy resources, etc. [24]. Although this property is not mandatory, consideration of its implementation is important. It should cover the most common failures, especially in vital applications such as the military, health, or PS.

The methods used for fault tolerance will depend on the system's requirements. For example, systems can handle faults through the concept of *fault detection, isolation, and recovery* (FDIR) [87], [88], [89]. Studies in computer science and robotics analyze faults as threats on hardware, firmware/OS, and application levels, which are studied as *cybersecurity* [90]. In PS, the idea of fault tolerance is explored as *Remedial Actions Schemes* (RAS) [91], [92]. In all of these applications, fault tolerance seeks the proper management of system resources in front of failures, and also focuses on three main components: fault detection, robustness, and resilience.

#### 1) ROBUSTNESS

This is the ability of the system to maintain its functionality and performance, even in the presence of uncertainties, disturbances, or unexpected events. A robust MAS can cope with changes and variations in the environment and continue providing the desired services and functionalities. The goal of robustness is to increase the system's performance, reliability, availability, and dependability, and ensure that the desired services and functionalities are provided even under adverse or challenging conditions. During the design and operation of a MAS, one pays special attention to the robustness of all its layers. This property is considered a pre-event or off-line concept since the system is generically designed to satisfy this characteristic [93]. A robust system's design considers not only the conditions in which it operates, but also additional disturbances that it can suffer during its normal operations such as hardware failures, load variations, or malicious software. Some works have focused on implementing robust systems for specific situations or under specific considerations [94].

#### 2) FAULT DETECTION

This is considered a logical component and an essential part of fault tolerance. It is capable of predicting the state of nodes and detecting unusual conditions of the system, through the use of the interaction between the elements in the MAS. Its main goal is to minimize the impact of faults on the system's performance, by quickly detecting and isolating the faulty agents or components, and taking appropriate corrective actions. This term could be studied accompanied by the term isolation and recovery, Fault Detection and Isolation (FDI), and FDIR systems. Whose aim is to determine the presence of faults events in a system, occurrence times, and location, and depending on the type and severity of them apply strategies to restore the functionality of a part or the whole system [89].

#### 3) RESILIENCE

Refers to the system's ability to maintain its functionality and adapt to changes, perturbations, or failures, even in the face of unexpected or uncertain events. A resilient MAS can recover from disruptions, adapt to changing circumstances, and continue providing the desired services and functionalities. Since the resilience methods used to correct node malfunction is different than the resilience methods against communication failures, the actions and behaviors of a resilient system depend on the type of failure, its location, or its magnitude. According to the field of study, the term resilience is usually found in conjunction with terms such as robustness or FDIR. In robotics, for example, the description of FDIR is normally used to encompass the aforementioned terms of fault detection and resilience. In this case, the system uses observers to estimate output values and take decisions if the real values are not within the estimated values. With this criteria also any type of information that may be harmful or malicious is isolated and strategies are considered to continue with the operation of the system [87], [88], [89]. Meanwhile, in PS, the concept of resilience is related to RAS, which by definition are event-based actions carried out to ensure the stability of the system in the presence of an event. Here, the output of the system or its response is constantly monitored and if changes occur outside of the normal operation, these actions are executed [91], [92].

## IV. FC-MAS IN TECHNOLOGICAL APPLICATIONS

Due to the versatility and benefits of MAS, its implementation in different technological areas has been carried out in recent years. Table 3 represents a dispersion of some works reviewed on various applications of MAS in the last years in different technological areas including some applications of MAS in PS, robotics, and even the Internet of Things. This section demonstrates the adaptability and generalization of the FC-MAS model through two cases. In the first example, we describe the transport of loads application using a multi-robot system in a distributed operation. In the second case, we describe the Load-Frequency Control (LFC) developed by the Automatic Generation Control (AGC) in a centralized manner in PS.

### A. MULTI-ROBOT SYSTEM

A multi-robot system (MRS) is a group of two or more robots capable of interacting with each other to achieve a common goal or set of goals such as cooperative formation, leader-follower, rendezvous problem, sweep coverage, etc. One of the challenges of designing MRS is ensuring coordination and communication between the robots to perform tasks that would be difficult or impossible for a single robot to accomplish alone. This can be achieved through various techniques, such as centralized control, distributed control, and consensus-based control. Additionally, designing MRS requires considering factors such as task allocation, motion planning, and obstacle avoidance. MRS can be applied in a variety of fields, including search and rescue, exploration, agriculture, transportation, and manufacturing. The advantages of using MRS include increased efficiency, robustness, flexibility, scalability, and the ability to perform tasks that would be dangerous for human workers.

As a case study, we will describe transports loads within a given area by a ground heterogeneous MRS [6], [12], analyzed from the perspective of our proposed framework presented in Section III. The global goal of MRS is to transport the load from one point to another under certain restrictions, ensuring their safety. Meanwhile, the local goal is to maintain a relative position with respect to its neighbors by following the planned trajectory at a given speed. Suppose that each robot is assigned a specific task, such as picking up a load, moving it to a specific location, or handing it off to another robot. The ground robot group is equipped with enough sensors to know some variables such as location, route, transported load, and speed. Additionally, everybody has built-in communication devices for sending and receiving information. The control system is based on a distributed architecture, where each robot has its own autonomy and communication with other robots to coordinate their actions and the development of the main task.

**TABLE 3.** Recent MAS applications according to different areas in the last seven years.

| Year/Area | Emergency Response | IoT and Smart Cities | Power Systems | Logistic and Delivery |
|---|---|---|---|---|
| **2022** | Blockchain in emergency management system [95] | Operator equipment resources for flexibility [96] | Resilience in generation, network, and load [97] | Pickup and Delivery with Autonomous Vehicles [98] |
| **2021** | Evaluation of covid-19 transmission [52] | | | Delivery with Reinforcement Learning [99] |
| **2020** | Search and rescue operations [100]<br>Earthquake rescue [49] | Modern industrial technologies [49]<br>Distributed task allocation [22] | | |
| **2019** | Disaster management with multi-criteria group decision-making [101] | Global consensus with internal delays and communication delays [42] | Optimal energy management and control [56] | Joint Delivery Systems in an uncertain environment [102] |
| **2018** | | Traffic management [45]<br>Blockchain for smart city [17] | | Robotic delivery service [103]<br>Smart traffic light management [46] |
| **2017** | Fire Emergency Search and Rescue Operation [11] | Fault detection, isolation, and recovery algorithms [88]<br>Cybersecurity issues [90] | Control and optimization for microgrids [104] | Coordination in logistic scenarios [105] |
| **2016** | | Remedial action schemes and defense systems [91] | Planning, market, management, operation and control [33] | Autonomous delivery tasks [106]<br>Supervision and monitoring of logistic spaces [107] |

### 1) PHYSICAL NETWORK

#### *a: NODE*

The ground robots' are the most basic components in this MRS, known as nodes or agents. Here, it is taken into account all the mechanisms that allow them to interact with the environment and with each other, such as infrared sensors or cameras for obstacle detection and evasion, GPS to determine its location, grippers to hold objects engines to move around, antennas for communication, and local commands to interact with the user [68].

#### *b: LINK*

For this application, the links can be considered in two types, physical and communication. The physical links consist of a series of interconnected mechanisms, such as a conveyor belt or a series of rollers, that allows the robots to move the load along a designated path. This is with the aim of unifying forces. While communication links are used to communicate with one another and coordinate their movements to ensure the load is transported safely and efficiently. It is common to use wireless communication, e.g. WiFi or radio frequency due to the range of coverage. Here, information transmission by the link can be subject to the bandwidth of the channel used [79]. Additionally, links can be customized to fit the specific needs of a particular application, making them a versatile solution for a wide range of industries.

#### *c: TOPOLOGY*

Due to the nature of the links, the MRS can have a variety of topologies, which can even change over time. Each of these configurations has its benefits and is chosen according to the application and will depend on the specific requirements of the task, such as the number of robots, the size of the load, the environment, and the desired level of coordination and flexibility. For this specific case, a mixed topology between tree and mesh topology is the most suitable. These provide the robustness and fault tolerance necessary to overcome the possibility of changing a topology when the nodes are in motion or the system requires reconfiguration.

### 2) SYNCHRONIZATION

#### *a: INFORMATION EXCHANGED*

Robots need to exchange information to effectively coordinate their actions and complete the task. Since this example focuses on the mobilization of loads to meet logistics objectives, the position and speed of the agent are the main data to be exchanged in the form of coordinates or through latitude and longitude points. This data is provided by a GPS, and allows the avoidance of collisions and maintaining efficient paths. Additionally, other relevant information to be shared are load weight and dimensions, load destination, operating status, battery-life charging status, and environmental conditions. With this, the robots can know your surroundings and they function smoothly.

#### *b: COALITION*

Inside coalitions, each robot has a specific role or task in the transportation process. For example, some robots may form groups for carrying the load, while others may be responsible for navigation or communication. This allows for the transport of more loads in fewer trips. Here, robots work together to transport the load efficiently and safely, taking into account factors such as the weight

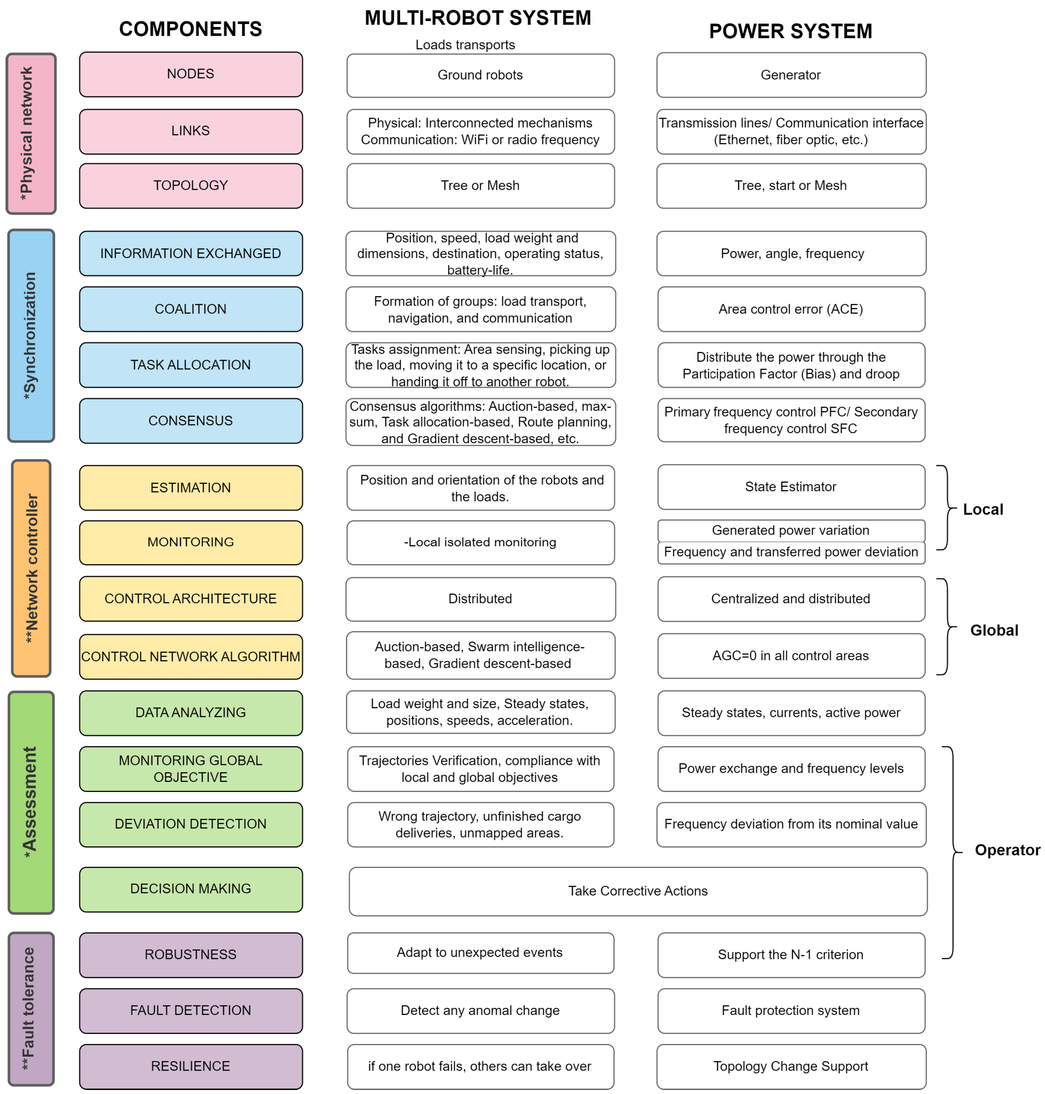


**FIGURE 2.** MAS framework applied to MRS and PS.

of the load, the terrain, and any obstacles that may be present. In this way, it increases efficiency, reduces labor costs, and minimizes the risk of injury to human workers.

#### *c: TASK ALLOCATION*

After the coalition, task allocation is the next step. This involves determining which robots should be assigned to which tasks in order to optimize the overall performance of the system. For this application should be considered factors such as the weight and shape of the object being transported, the terrain and obstacles in the environment, the capabilities of the individual robots, and the overall efficiency and effectiveness of the system. General tasks assigned to the majority of robots are area sensing, picking up the load, moving it to a specific location, or handing it off to another robot. Additionally, some specific robots may be managed for navigation or communication.

#### *d: CONSENSUS*

This component plays a crucial role to load transport tasks in MRS, as they enable the robots to coordinate and make decisions on how to distribute the load among themselves. They aim to minimize the total transportation time or maximize the total load transported. Some consensus algorithms for load transportation are Auction-based consensus, max-sum algorithm consensus, Task allocation-based consensus, Route planning consensus, Swarm intelligence-based consensus, Collision avoidance consensus, Load balancing consensus, and Gradient descent-based consensus. The choice of algorithm depends on the specific requirements of the load transport application and the characteristics of the MRS.

### 3) NETWORK CONTROLLER

#### *a: ESTIMATION*

To estimate the states several factors need to consider such as the number and types of robots involved, the characteristics of the loads being transported, and the environment in which the robots are operating. Typically, the position and orientation of each robot, the position and orientation of the loads, and the velocities of the robots and loads are the state variables to estimate. For this purpose, kinematic or dynamic models are used to describe the motion of the robots and loads. Estimation techniques such as Kalman filtering, particle filtering, or model predictive control are the most used to estimate the states of the complete system [75].

##### b: MONITORING

To develop this activity in MRS, local sensors are used, which provide information about the positions, speeds, and charges of the robots. In addition to monitoring the robots themselves, monitoring the loads that they are transporting is important too. Sensors can be placed on the loads to measure their weight, temperature, and other relevant parameters. This information is presented in real-time on a local monitor and is used to track the progress of each robot, detect any anomalies or failures, ensure that the loads are being transported safely, and optimize the system's performance [108].

##### c: CONTROL ARCHITECTURE

The current system works with a distributed architecture because each agent is in charge of generating its control actions, so it does not require a central entity to send it some type of signal to function. Each robot uses its sensors to perceive its environment and the objects it needs to transport. Additionally, they can communicate with each other to exchange information about the objects being transported, their locations, and the paths they are taking. For this application, the distributed architecture is ideal to facilitate the scaling of the system when the integration of new agents is required to transport heavier loads.

##### d: CONTROL NETWORK ALGORITHM

For the effective transport of loads in MRS several control algorithms have been proposed to coordinate all the agents [109]. The most common are:

- Distributed consensus algorithm, where each robot sends its value to its neighbors, and they update their values based on the received values. This process continues until all the robots have the same value.
- Auction-based consensus algorithm, robots bid for the task of carrying the load. The robot with the lowest bid is selected to carry the load, and the payment is distributed among the other robots based on their bids.
- Swarm intelligence-based consensus algorithm, robots work together as a swarm to transport the load. The swarm can adjust its behavior based on the load weight and shape to optimize the transport process.
- Gradient descent-based consensus algorithm, robots use gradient descent to find the optimal path for transporting the load. Agents communicate with each other to share information on the load's weight, shape, and terrain, and they collectively find the path with the minimum energy consumption.

#### 4) ASSESSMENT

##### a: DATA ANALYZING

This application can generate a significant amount of data that can be analyzed to improve the system's performance. For example, some data that can be collected and analyzed are load weight and size to optimize the number of robots required, robot speed and direction to ensure that they are moving efficiently and avoiding possible collisions, power consumption to reduce the load on certain robots, route optimization to identify areas where congestion is likely to occur, and finally performance metrics to evaluate the overall effectiveness of the system.

##### b: MONITORING GLOBAL OBJECTIVE

The information on the current state of the system is compared with the reference value that it must achieve. Here, robots are programmed to prioritize certain tasks or to adjust their behavior based on the current state of the system. For example, information such as the trajectory followed or the percentage of packages transported correctly can normally be viewed in a graphical interface and thus know how much is missing to achieve the overall objective.

##### c: DEVIATION DETECTION

Focusing on the global objective of this robotic system, it is necessary to have procedures that allow for identifying deviations from the objective. Some common deviations are alterations in the trajectory of the agents, agents not able to withstand the load, or non-fulfillment to transporting the load for a certain lapse of time, to name a few. In case the robots observe these anomalies in the measurements provided by the environment, they will use relevant information to conclude the possible causes of this erroneous behavior in the system. Subsequently, the consecutive action is to minimize said deviation by executing a set of control rules in the predefined algorithms.

##### d: DECISION MAKING

In the application, the robots need to consider possible external factors such as robot failure, the weight change of the loads, the distance between locations, and any obstacles in their path, that may affect the objective. If an adverse event occurs, commonly one robot takes hierarchical orders to take corrective actions that benefit the entire system. This agent charge of processing and sending new instructions to the other agents so that they can execute them according to their capacities and the agreements they have reached.

#### 5) FAULT TOLERANCE

##### a: ROBUSTNESS

The key to robustness is to design the system with redundancy and flexibility in mind so that it can adapt to unexpected events and continue to function effectively even in the face of individual robot failures or environmental disruptions. To solve the load transportation in MRS, adaptive algorithms that can adjust robot behavior in response to changing conditions are used. For example, if a robot encounters a particularly heavy load or an unexpected obstacle, the algorithm could adjust the robot's speed or path to ensure that it can safely navigate the environment and complete its task.

#### b: FAULT DETECTION

For this application, the goal of fault detection is to minimize the risk of failures or errors that could lead to delays or damage to the transported goods. Each robot is equipped with weight and location sensors that measure the load and position to detect any changes or deviations from what was agreed. If a sudden drop in weight is detected, the agents automatically trigger an alarm or alert to investigate the issue. In some cases, is necessary to send a replacement robot to take over the task of carrying the load, while the faulty robot is taken offline for maintenance.

#### c: RESILIENCE

In the context of load transportation in MRS, resilience can be achieved through several approaches, such as redundancy to ensure that if one robot fails, others can take over. Implementing Adaptive control and learning in each of the robots, task allocation, and considering in the design step some extra physical or communication links.

## B. POWER SYSTEMS

The second study case analyzed from MAS viewpoint is the Power System which we refer as PS. As is well known, PS is a system designed to provide a continuous and reliable electric power supply that meets certain quality requirements. And although the power requirements could be provided by an isolated generator connected to the loads directly, this operation way can compromise the service continuity during failures in the generation sources. In this regard, the interconnection of generators not only increases the system capacity, but also provides a backup mechanism between generators during fault of one generator.

During PS operation, the global goal is to keep the balance between generated and consumed power, which is why when a power variation is presented it is necessary to implement a system reconfiguration that restores such balance. The mentioned reconfiguration is accomplished by LFC and is related directly with local objective of nodes that is to regulate the power provision according to power imbalance. Regarding LFC, it is implemented in two stages. The first one called primary regulation is performed by the governors to achieve an steady state after power imbalance, although with frequency and transferred power deviated from their rated values. The second stage called secondary regulation is accomplished by AGC that change the power setpoint of generators to restore balance of active power, and as result two main objectives are accomplished: (1) restore system frequency to its rated value, (2) remove the transferred power deviation in transmission lines.

An important point to keep in mind is that PS is an interconnected system by nature in which AGC forces an additional level of synchronization. Therefore, we can talk about two kind of interconnections, first one is a natural connection created by transmission lines and the second one that is forced by the AGC's control schemes, which together enable generators in PS to describe a collaborative behavior against power unbalance. Finally, it is important to highlight that primary regulation is implemented by the using of local controller (speed governor), and supplementary regulation is implemented at dispatch center level.

### 1) PHYSICAL NETWORK

#### a: NODE

In PS, generator and its governor is considered as the node from frequency control problem viewpoint. This is because LFC scheme performs variations in the mechanical input power to drive a generator speed variation that contributes to regulating the system frequency. Under this consideration the generator can be identified as an elemental component because any change in its operating point has an effect on the system frequency value.

#### b: LINK

In PS there are two kind of links that enables information exchange between nodes. The first one are the transmission lines which allows to exchange power that couples the generators and drive them to converge to the same speed thus defining the frequency on whole PS. The second one is instead provided through a communication network that enables to send the desired generator operating point from dispatch center to governors mainly. Since the different nature of this links, we will define the first one as natural links and the second one as artificial links.

#### c: TOPOLOGY

As was just mentioned, the PS presents two kind of links that could define the topology. But in LFC the prevailing topology is related to transmission lines, i.e., to natural links. In this regard, PS presents a meshed topology in order to gain robustness and reliability. This is justified by the fact that PS arranges several generators interconnected between them to distribute the generated power requirements. As well as power unbalances that may appear suddenly could be counteracted too returning PS to an stable state after perturbation.

### 2) SYNCHRONIZATION

#### a: INFORMATION EXCHANGED

Regarding the information exchange LFC control scheme first takes advantage of natural coupling provided by transmission lines to perform the primary regulation through the governors. This first regulating stage is triggered by an active power unbalance that modifies the speed of its closer generator and that in turn produces a variation on exchange power with the rest of generators. As result, this power variation is propagated through the transmission lines making known to remaining generators about power unbalance. In answer each governor changes the mechanical input power in its generator helping counteract the power unbalance leading to the PS to an steady state operating point with a new frequency value. It is important to highlight that the primary regulation is accomplished by the governors in a distributed way thanks to the natural consensus. After in order to restore the rate value of system frequency, the secondary regulation will be performed by using its artificial links, i.e. the communication network. In this case the information is mainly send by dispatch center to governors to establish the new power reference on every system generator.

#### b: COALITION

From our analysis we conclude that the control area (CA) defined to regulate the power exchange between power companies is the coalition in LFC, since it is a group of generators which are governed under a common law based on the Area Control Error (ACE). ACE can be described by the expression (2):

$$ACE_n = B_n \Delta\omega_n + \sum_{i=1}^{n} \Delta P_{nij} \tag{2}$$

The control law of an area establishes that when the ACE is different from zero, control actions will be performed inside the area to restore the condition $ACE = 0$ and in turn to counteract PS unbalance.

#### c: TASK ALLOCATION

As mentioned above, if the ACE is different from zero a control action must be established for whole area according to active power unbalance. This control action should define the new power set-point for every generator inside the area. This additional amount of generated power in each generator depends on speed regulation or droop $R$ that is a parameter configured in every governor. This parameter acts as a participation factor that define the ‘‘task allocation’’ for every generator by defining the power percent by which each generator contributes to counteract the power unbalance.

#### d: CONSENSUS ALGORITHM

In the PS could be identified two kind of consensus, a natural or intrinsic consensus and an artificial or forced consensus. Natural consensus is observed when two or more generators which are interconnected converge to same speed without external control actions although no rated value necessary, i.e.,

$$\omega_1 \rightarrow \omega_2 \rightarrow \cdots \rightarrow \omega_n. \tag{3}$$

This advantage of natural consensus AGC is exploited by primary regulation to lead the system to a first stable operating scenario by creating a deviation on the power reference according to the regulation $R$ of every generator. Meanwhile secondary regulation eliminates frequency and transferred power deviation by the convergence of the deviation of the generators angles, i.e.,

$$\Delta\delta_1 \rightarrow \Delta\delta_2 \rightarrow \cdots \rightarrow \Delta\delta_n. \tag{4}$$

This behaviour could be considered as an artificial consensus because it is accomplished by dispatch center through communication links in order to modify the power output reference in each generator.

### 3) NETWORK CONTROLLER

#### a: ESTIMATION

Even though estimation could be associated to state estimation applications based on system measurements, in our analysis it refers to the PS or node capability to know the state variables of neighboring nodes or the whole system. About variable estimation on LFC problem, we separated the state variables that are involved in primary frequency regulation that are speed and angle of generators and variables related to secondary regulation which are system frequency and transferred powers. For primary regulation, every generator knows its own state variables (angle and speed) and the transferred power to the neighboring generators. These information allows to generator to ‘‘estimate’’ state variables of its neighboring from local variables only. In the other hand, dispatch center can directly measure system frequency in some PS locations as well as transferred power on transmission lines to know such variables, i.e. the value of required variables are available without any complex estimation procedure.

#### b: MONITORING

Since monitoring stage aims to enable to every node the knowledge about its own and the system situation, in the LFC context monitoring must detect the occurrence of a power unbalance. This monitoring is different in primary and secondary regulation because the first one depends on natural connection and the later on an artificial connection.

From generator viewpoint in primary regulation, unbalance is detected as an active power variation in the generator terminals because this variation is a result whether by a local or a remote active power variation. For secondary regulation in the dispatch center the key variable that must be monitored is system frequency, because a deviation from its rated value could be understood as an active power unbalance in PS.

#### c: CONTROL ARCHITECTURE

The LFC control architecture could be considered as a two-level hierarchical architecture because a power unbalance is counteracted in two stages as was mentioned before. In the first level is the primary regulation that is the closest level to the PS and works under a distributed architecture since the control actions are taken by the governors. In the other hand, at the second level in the architecture is the secondary control whose actions are defined in the dispatch center, therefore its architecture could be considered as centralized. From this we can conclude that the frequency regulation is performed under a hybrid architecture.

#### d: CONTROL NETWORK ALGORITHM

From the fact that at the PS level the objective is to counteract the active power unbalance, therefore the main aim is to reach the balance between generated power and consumed power, i.e.,

$$\sum_{i}^{n} P_{gi} = \sum_{j}^{m} P_{lj} \tag{5}$$

could be considered the basic relationship that will define any control algorithm for LFC. Under this consideration the classical LFC drives a change in mechanical input power of generators that in turn change the electric power generated to restore the power equilibrium. The additional amount of generated energy is defined on base of $ACE$ error minimization that in an ideal scenario must be zero, i.e.,

$$ACE_1 \rightarrow ACE_2 \rightarrow \ldots \rightarrow ACE_n \tag{6}$$

that in turn accomplishes

$$\Delta\delta_1 \rightarrow \Delta\delta_2 \rightarrow \ldots \rightarrow \Delta\delta_n \rightarrow 0 \tag{7}$$

resulting in a new power balance as well as a system frequency restoration and transferred power deviation elimination.

### 4) ASSESSMENT

#### a: DATA ANALYZING

Since data analyzing is an assessment stage at ''supervisory'' level, secondary regulation is considered only. That means data analyzing will provide useful information about system frequency and transferred power through transmission lines that enables the monitoring global objective. In this regard frequency and power through transmission lines are measured and centralized in dispatch center to calculate the current $ACE$ to evaluate if a power unbalance is presented.

#### b: MONITORING GLOBAL OBJECTIVE

PS must monitoring the active power balance by checking that system frequency must be maintained constant and ACE error must be zero. If any of both conditions isn't true, it could be understood as a power unbalance that requires corrective actions that should be applied and maintained while variation persists since it will mean that PS unbalance is not compensate yet.

#### c: DEVIATION DETECTION

As mentioned in Control network algorithm, global objective in LFC is the active power balance in order to let the system to an stable operating scenario. In this regard, PS monitoring if its objective is accomplished by detecting system frequency variations and deviations from scheduled values of transferred power between areas. In this regard,PS unbalance is detected from ACE value, i.e., if ACE value is different to zero, then control actions are needed in order to return frequency and transferred power to their scheduled values.

#### d: DECISION MAKING

After an unbalance of active power is detected, control actions are taken to counteract it by changing set-point of power output on some generators in the affected control area. In this stage, dispatch center defines the new operation points that are sent to generation station by a communication system, i.e. by an artificial link. After a new set-point for generators at station is configured according with characteristic of every one.

### 5) FAULT TOLERANCE

#### a: ROBUSTNESS

About robustness it must be taking in account that LFC leads PS generators to respond to a disturbance in a cooperative way by distributing the ''responsibility'' to each according its capability. Therefore it is possible to conclude that LFC scheme itself is a mechanism that provides robustness to PS. In addition, procedures for PS operation have established reliability criteria such as $N-1$ criterion that pretends that PS is able to withstand at all times an unexpected failure or outage of a single system component, has an acceptable reliability level.

#### b: FAULT DETECTION

From LFC viewpoint fault detection could be accomplished by a similar procedure of PS unbalance detection, that is by detecting frequency and transferred power deviation from their planned values. But it is important to consider that LFC is designed for counteracting power unbalances that are normal during PS operation an in which variables present a tolerable variation unlike fault conditions in which the variables could be deviated considerably. In this regard, although LFC can detect the fault, it cannot face a considerable power unbalance produced by it. Therefore fault detection task must be accomplished by a systemic protection scheme that operates in parallel to LFC and stays as a passive function while PS maintains its normal operating and performs its control action under a fault condition only.

#### c: RESILIENCE

In order to increase PS resilience several generators with different power production characteristics could be considered because they have the capability to deal different kinds of weather conditions, natural disasters, etc. Further, a coordinated scheduling of different power sources is effective to improve generation resilience. Also transmission networks need to have resilience to face disturbances caused by natural disasters or operating reconfiguration. Thus alternative corridors should be established in the event of a main transmission path faults. Also with a future vision, smart loads could be an alternative because they would have regulating abilities for contributing PS resilience [97].

## V. MAS DEPLOYMENT FLOW

Implementation of MAS phases like execution and start-up can be a quite complex operation due to the heterogeneity of the interconnected agents and the conditions of the environment in which they interact. Each application requires the integration of a set of algorithms, procedural protocols, and frameworks for its correct operation, but up to date, there is not a proposed organized scheme that guides what are the basic resources that researchers should consider to implement an application in MAS. As a consequence of the framework presented in section III, this work complements the proposal with a workflow for the practical implementation of multi-agent systems in any area. For this, workflows based on a sequence of stages and steps for centralized and distributed architectures in MAS are proposed. Fig.3 presents a general description of the necessary stages for the process of execution and commissioning of a MAS, seen from the engineering perspective. This proposal is based on defining certain implementation steps within three stages: Initialization, Supervision, and Control. Next, we proceed to describe the architectures, their stages, and the variants of the necessary steps according to the operation requirements for each application.

### A. ARCHITECTURES

*Centralized*: In the 60's the first MAS applications began with a centralized architecture, which is characterized by depending on a main agent called the control center or master, who is in charge of coordinating all the tasks of the rest of agents [34], [78]. This architecture is suitable for solving applications where they require a limited number of agents. Its greatest limitation is its capacity to expand, since by having to centralize all the information in a central agent, it must have a great processing and decision-making capacity to generate the control orders of the rest of the agents. For this case, centralized workflow is shown in Fig.4. Here, the proposed stages and those responsible for establishing a centralized MAS system are: the master agent, who coordinates the actions of the rest of the agents, will be in charge of executing the initialization and supervision stages, and also the global control sub-stage. The master will establish the global objective to be achieved and will be in charge of monitoring the status of the deviation from the progress of the agents' tasks. In the event of adverse circumstances within the MAS, the master will execute a decision-making function to try to return the system to operating conditions. While in the local control system stage, agents sense and manipulate the environment to fulfill their tasks. Here, the agents execute the control commands issued by the agent master For example, a Power System's frequency control coincides with being a multi-agent system with a centralized architecture [110]. Where, its control center decides and establishes the power of each of the generation units, while these change their power until reaching an energy balance according to the order received.

*Distributed* : The development of MAS applications with a distributed architecture has gained importance in the last decade because of the limitations of the centralized architecture. This architecture distributes the main functions, previously performed by a master, to the autonomous agents (senior agents). This is achieved by defining a set of interaction rules between the agents. The challenge is to coordinate the decentralization and distribution of decision-making processes among individual autonomous agents, so that each agent actions do not interfere with the activities of their neighbors, while achieving a common objective. This analysis considers new aspects of centralized architecture, such as instruction execution, primary resource distribution, communication channel usage, command hierarchy, establishment of interaction rules, etc. Several proposals from different areas have attempted to solve this challenging task. Topics such as energy efficiency, real-time performance, coexistence, interoperability, security, and privacy have been addressed in relation to how distributed control has facilitated the rapid development of Industry 4.0 [111], [112]. Similarly, methodological proposals demonstrate the feasibility of distributing control models to achieve greater benefits from MAS [113], [114]. Its implementation opens new challenges such as the difficulty in distributing and coordinating the tasks of each agent, the synchronization problem, and obtaining consensus on the variables of interest [35]. This architecture allows the scalability of a MAS system with the incorporation of new agents [115]. In case the application requires a master agent as a global supervisor, this will be responsible for executing only the initialization and supervision stages. While in the control stage, autonomous agents now with greater processing capacities perform main tasks such as coordination, decomposition, and decision-making on their own. Through the implementation of distributed algorithms, the agents establish the adequate exchange of information and each one is capable of coordinating their own actions for the fulfillment of their individual tasks, which in turn converge in accomplishing successfully the common global objective. The proposed workflow for this case is shown in Fig. 5. In this case, since each of the autonomous agents determines its own orders, the system itself can work faster because it no longer waits for the master to make and send the decisions. However, the master will now take care of

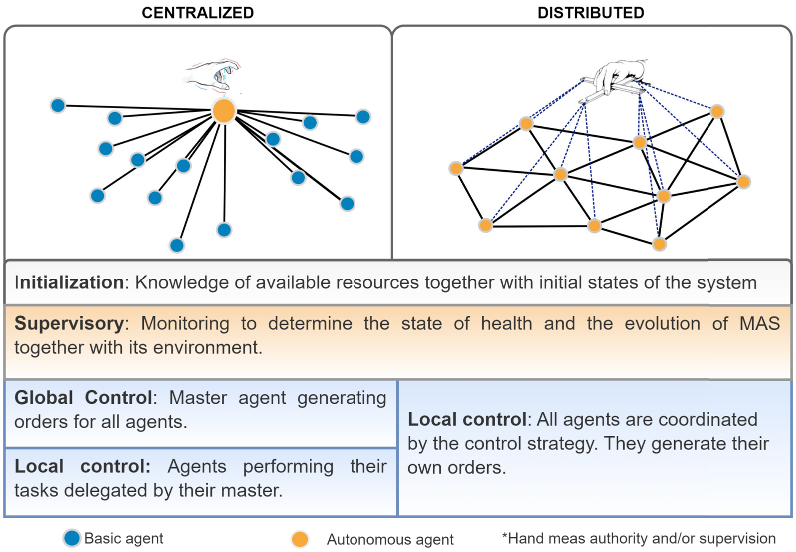


FIGURE 3. Centralized and distributed architectures.

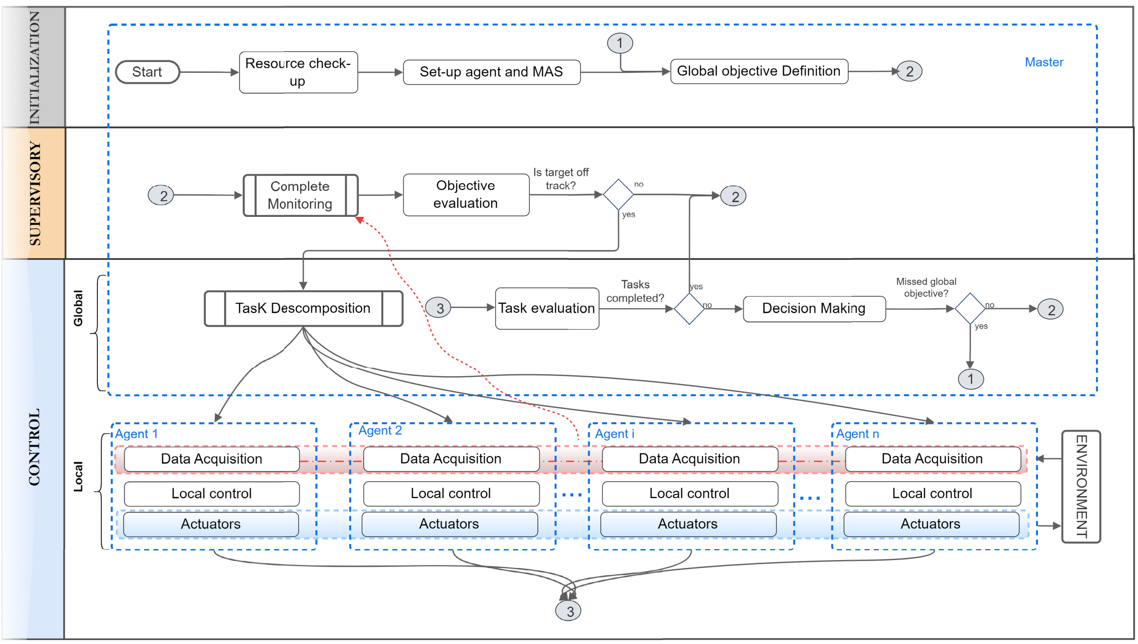


FIGURE 4. Workflow centralized.

hierarchical functions such as objective definition and also monitoring for system convergence.

### B. PROPOSED STAGES AND STEPS

In the proposed workflow, a stage refers to a group of steps that collectively aim to achieve a specific goal, and they need to be followed sequentially to establish a MAS. Depending on the architecture type, these stages can be executed either by the master agent in a centralized scheme or by the senior agents in a distributed scheme. The proposed stages include initialization, supervision, and control. Figure 3 illustrates these stages arranged hierarchically from top to bottom. The utilization of individual steps within these stages depends on the specific requirements of the MAS application under development. This reliance is influenced by factors such as architecture, available resources, number of agents, communication infrastructure, etc. [78]. The design of the proposed workflow aims to accommodate a wide range of applications, but it also maintains the flexibility to exclude certain steps that may not apply to specific MAS applications or when their execution is carried out by an external operator.

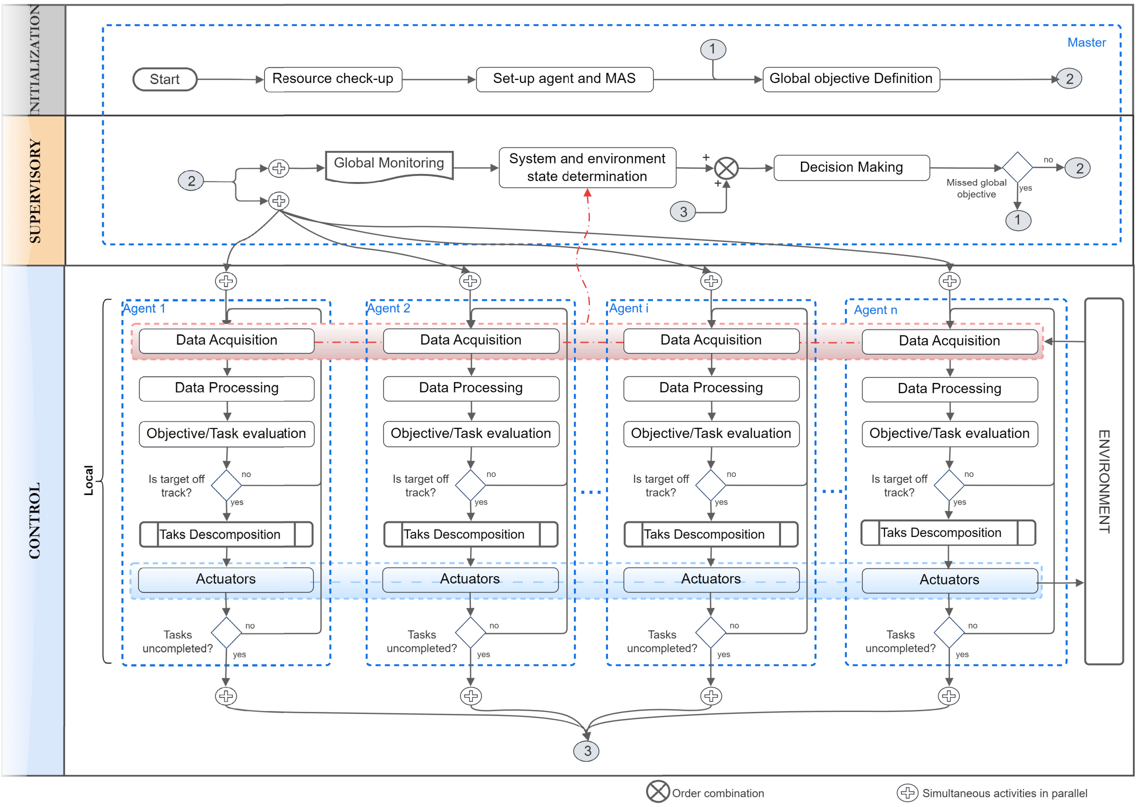


**FIGURE 5.** Workflow distributed.

The following sections describe each stage in detail, along with their corresponding steps.

### 1) INITIALIZATION

Like any process that requires initialization, a MAS is no exception. This stage is typically carried out by a supervisor (sometimes referred to as a global master) in either a centralized or distributed scheme. Its primary objective is to construct a MAS application by leveraging available resources and their initial states. In this regard, launching any application involves an analysis of the available resources to ensure the operability and autonomy of each agent. Following that, the system variables are fully initialized at both the agent and global levels. Lastly, the MAS establishes or proposes the global objective to be achieved [108]. The recommended steps for this stage are as follows:

*Resource check-up*: A crucial aspect of initializing any Multi-Agent System (MAS) is gaining a comprehensive understanding of all the physical and logical resources at hand. By assessing their availability, it becomes possible to define objectives with varying degrees of scope. The aim of this step is to quantify the MAS's state of health, offering insights into the scope and limitations when formulating global objectives based on the available resources [8]. To accomplish this, a detailed inventory of the equipment should be compiled, including the number and current status of agents, communication channels and infrastructure, computational resources of the supervisor, primary energy sources, and, if feasible, additional resources within the agents' operating environment.

*Set-up agent and MAS*: After defining the resources, the next step involves activating the main components of the MAS. This involves too initiating communication between the agents. The initial activation of these components will provide information on the initial conditions of each entity within the MAS [28], [34].

*Global objective Definition*: In this step, the design of functions and activities to be implemented within a Multi-Agent System (MAS) is undertaken. The primary aim is to establish the operational objective, which defines how MAS agents will interact with their environment to accomplish specific tasks [5], [28]. The scope of the overall objective may be constrained by factors such as resource availability, component state, and the creative abilities of the MAS implementation team. Depending on the engineering application, the overall goal can range from a complex set of multiple tasks to a relatively simple exercise. Regardless, it is crucial to regularly evaluate the MAS's progress towards achieving the proposed global objective. Additionally, it should be noted that advancements in technology, techniques, and algorithms provide new opportunities to flexibly update the global objective based on potential new requirements arising during the decision-making process of the MAS [6].

#### 2) SUPERVISORY

We can analyze the second stage of superior hierarchical order from two perspectives. Firstly, a central entity (referred to as the master agent) oversees and evaluates the overall alignment of the MAS in achieving the proposed objective. Secondly, each agent is responsible for local monitoring internally to validate the fulfillment of its assigned task if is a distributed architecture. Regardless of the scenario, a continuous monitoring function is active to assess the condition and progression of the MAS and its environment. This evaluation of information helps determine whether or not the MAS will achieve its objective [59], [108].

*Monitoring*: In this particular step, the focus is on the execution of a fundamental workflow function. This function is responsible for supervising the acquisition and processing of data in order to determine the current state of the agents and their environment [82], [108]. To carry out this task, at least two layers of a Multi-Agent System (MAS) are involved: the physical network layer and the synchronization layer. These layers rely on the measurement acquisition infrastructure installed either in the agents themselves or within the MAS environment. The primary objective of this step is to consistently identify patterns of deviation or errors in task completion or the overall objective. This is crucial for issuing accurate control commands. Within a MAS, the monitoring function is executed, and the results are presented through a visualizer, display, or a comprehensive SCADA system. This ensures that operators have access to up-to-date information regarding the agents' behavior and information exchange.

*Decision Making*: The decision-making capacity is a crucial element that applies to both individual agents and the overall MAS [7]. It is an integral part granted to the elements of the MAS to fulfill their assigned tasks effectively. Various prominent algorithms in this field include Markov decision processes (MDP), Partially Observable MDP, game theory, swarm intelligence, graph theoretic models, reinforcement learning, dynamic programming, evolutionary computing, and neural networks [8]. The primary goal of this process is to assess the cost/benefit objective function and make optimal decisions that enable each agent to fulfill its designated task while ensuring the MAS as a whole achieves its intended objective.

#### 3) CONTROL

To achieve the global objective, the tasks can be distributed among the components of a MAS. This stage is crucial for the proper functioning of the MAS, and its implementation steps depend on the chosen architecture. If the application is centralized, the master agent takes full responsibility for generating orders for all agents. On the other hand, in a distributed architecture, this stage is carried out by the agents coordinated by the control strategy [61], [63]. This process involves several steps, including task decomposition and evaluation. It can occur after the initialization or during the operation of the MAS, in which case the system may have intentionally or unintentionally changed, resulting in a shift in its global goal or consensus state.

During this stage, the performance of each agent in fulfilling their assigned task is individually analyzed. In the centralized case, local-level controllers are implemented within the agents to manipulate their variables of interest and accomplish their tasks [56]. In contrast, in a distributed architecture, this stage is considered as part of the General Control System stage. The internal controllers can be simple or complex and are implemented based on the characteristics and computational resources available in the agents. Commonly used controllers include comparators, hysteresis, and proportional-integral-derivative (PID) controllers due to their ease of implementation. However, agents with advanced capabilities can now include controllers such as fuzzy, fuzzy-PID, neural networks, and sliding mode control (SMC) [28].

*Data Acquisition*: This step involves measuring the environmental variables either through sensors installed in the agents or strategically distributed sensors in the work area [47]. The collected data is then shared with the supervisory entity or neighboring agents within the network using various communication channels. It is important to note that since the data is exposed to the work environment, it is susceptible to noise. Therefore, if noise is present, a filtering stage should be implemented to mitigate its impact. This approach helps reduce the bandwidth usage of the communication network and minimizes computational efforts required for other processes [80].

*Data Processing*: This step initiates the processing of the gathered information concerning the environment and the progress made by the agents in reaching their objectives. Depending on the task assigned to the agents in aMAS, one or more of these agents will be responsible for processing the data and transforming it into meaningful information that has been predefined. This relevant information will then be disseminated across the network [80]. If there is any loss or alteration of information during communication, data mining algorithms can be employed to eliminate or rectify erroneous data.

*System and environment state determination*: In this step, the current state of the system and the environment in which the agents are performing their tasks is determined through the implementation of a series of techniques and strategies. This information plays a crucial role in supervision as it enables the assessment of whether the MAS is progressing towards its objective or if it has deviated from it [25], [28]. During the initial execution, after establishing the global objective, this step quantifies the availability and operational status of resources. This enables the identification of resources and facilitates the distribution of activities.

*Objective evaluation*: This step focuses on identifying any discrepancies or errors in the MAS concerning the defined global objective if it is under wide supervision. Too, it aims to detect deviations in the variables of interest compared to their reference values if the supervision is local. If the system is nearing its objective, the workflow will return to the monitoring step without requiring any control actions. Conversely, if deviations exist, the workflow will proceed to implement algorithms that facilitate achieving the necessary consensus. The algorithms used to assess deviations can range from basic methods, such as comparing the reference value to the actual measurement, to more intricate approaches involving numerous calculations and estimations [57]. The latter plays a crucial role in understanding state deviations, changes in the global task, or the MAS itself, and taking appropriate actions accordingly [10].

*Task Decomposition*: In this step, the analysis of the global objective takes place, and its distribution into specific tasks is planned to incorporate it into the MAS based on the unique capacities and functions of each agent. In a basic MAS, this task is typically handled by the designer beforehand. However, in certain cases, certain algorithms related to task decomposition are employed in subsequent steps of the process, such as coalition formation and team allocation [5], [70]. This phase of the process is crucial for attaining the global goal, as an optimal system response can only be achieved through the proper allocation of activities among all components.

##### a: RESOURCE ALLOCATION

Due to the MAS's global objective and depending on the agents and the environment, often one needs to analyze the resources available. This sub-process is in charge of analyzing the environment, distributing the resources and tasks, and creating teams of agents. This step requires specific algorithms for this purpose which includes Artificial Intelligence (AI), Machine Learning (ML), adaptive algorithms among others [22], [71]. One of the most important consideration when distributing and optimizing resources is the amount of time or energy that an agent needs to reach certain state related with the global objective.

##### b: COALITION

This sub-process is in charge of creating groups or teams of agents to maximize the performance of the agents in the MAS while achieving the global goal. Here, the main focus is the global objective (or consensus) without considering the environment [65], [66]. Together with the resource and task allocation, this sub-process can accomplish an optimal performance when creating teams. As in resource allocation, the algorithms used for this purpose are based on AI and ML.

##### c: TASK ALLOCATION

In this step, the global task or the specific task are assign to each team or agent. In other words, this part of the process defines the action that must take each team or agent to achieve the global goal or a consensus [21], [71].

*Task evaluation*: The tasks in the MAS are assigned to one or several agents according to their capacity and their willingness to operate [84]. This step focuses on evaluating the fulfillment of these tasks and achieving the desired objective. In the event that the agents are capable of fulfilling their previously assigned tasks, the workflow will return to the monitoring step since everything is working as expected. But in the event that for internal reasons (exhaustion of useful life, lack of fuel, lack of communication, etc.) or external reasons (such as changes in the environment with conditions not foreseen in the design) to the agents, the tasks may be redefined by a supervisor based on available resources. This last case will give way to decision making.

*Actuators*: This step corresponds to the execution of orders by the agents in order to achieve their objective or proposed goal, as appropriate [47], [55]. Here the agents will see the need to interact with the environment and, if necessary, manipulate it at their convenience through their actuators or final elements. Depending on the control actions, the changes that the environment may undergo can be small or large.

### C. WORKFLOW EXAMPLE

Below, we use the proposed workflow to describe the start-up of the LFC in a PS carried out by AGC in a centralized way, specifically, for the problem of primary and secondary regulation. We consider that the PS starting operating conditions are a stable state, that is, all the generators have their power reference $P_{ref_i}$ and there is a balance between demand and generation ($\sum_i^n P_{gi} = \sum_j^m P_{lj}$). Furthermore, it is assumed that the generators have been synchronized concerning the slack bar.

#### 1) INITIALIZATION

*Resource check-up*: The first step consists of determining the exact resources that PS has, such as the number of generators, transmission lines, and auxiliary equipment. It is necessary to know the operating status of each one of them to determine whether or not it is in a condition to be integrated into the LFC.

*Set-up agent and MAS*: The control center executes the optimal power flow to establish the operating scenario of

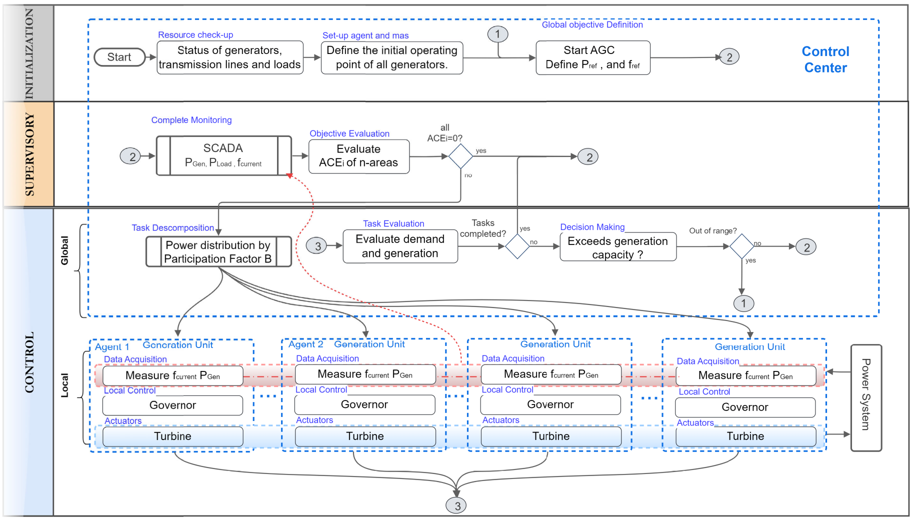


**FIGURE 6.** Workflow example of PS. The red line represents the feedback to close the control loop. All generators send their measurements to the SCADA.

each of the generators, thus giving its initial condition. With this, the power of each generator is known before a load variation event in the system.

*Global objective Definition*: The global goal of the LFC in any PS is to keep the balance between demand and generation, and also maintain the frequency of the system constant, so the control center deploys the AGC described above into action. This controls centrally, the primary and secondary regulation stages so that the system always remains within its nominal operating values. Here, the references of the system are defined, both in Power ($P_{ref}$) and frequency ($f_{ref}$).

### 2) SUPERVISORY

*Monitoring*: It is necessary to know the current state of the PS to determine if it effectively meets the balance condition between generation and demand. The variables that are measured in each ACE in the system correspond to the power generated, the power transferred, and its current frequency. These variables are monitored in the SCADA and in a later phase they will be used to control the balance of power exchange.

*Objective Evaluation*: Corresponds to the evaluation of $ACE_n$ to eliminate power deviations in each of the generation centers and finally maintain the balance between generation and demand. If frequency variations concerning the reference are detected, the ACE calculates the total power that must be compensated in the overall system.

### 3) CONTROL

*Task Decomposition*: The power that has been determined by the ACE to compensate for frequency variations is sent to the generators through a participation factor $B_i$ to distribute the power among them thus avoiding their competition. The local goal of each generator is to increase its power production. It only takes what it is entitled to.

*Task evaluation*: Compliance with frequency regulation occurs when the ACE is equal to zero, that is, there are no frequency deviations, so the required balance between generation and demand has been reached. If the ACE is different from zero, the system determines the power that each generator must absorb, which is done in the decision making step. This task is constantly performed.

*Decision Making*: It is activated when there are deviations with respect to the objective, for instance, when the consumers' consumption is higher or lower than the power produced by the generators, or when the power consumed exceeds the maximum power that the generators can provide. Depending on the situation, the system should decide to turn on/off generators, redistribute the power generation, trip load, or reconfigure the network so the system can continue operating.

*Actuators*: Hydraulic and thermal energy are typically used as primary resources, which are then transformed into electrical energy as they pass through the mechanical and electrical components of the generator. In this case, actuators such as valves and gates, are used to regulate resource flow and control the turbine's speed, which in turn moves the generator's rotor. These actuators are located inside the governor and interact with the turbine.

## VI. CONCLUSION AND FUTURE DIRECTIONS

This study introduces FC-MAS, which offers an effective abstraction for component integration and resources, facilitating the understanding and development of engineering applications in the MAS. Each layer focuses on specific aspects, such as physical network, synchronization, network controller, assessment, and fault tolerance. Furthermore, it outlines a comprehensive workflow that describes the essential stages and steps multi-agent systems must follow to accomplish their objectives. To illustrate the practicality of this layer model, we apply it to the domains of robotics and power systems, showcasing its effectiveness. Additionally, we provide a comprehensive compilation of key terms pertinent to the study of multi-agent systems, accompanied by real-world examples demonstrating MAS applications.

During the development of this project, some basic approaches have not been addressed in depth, so its full development is proposed for future work. Such is the case of the selection of essential components, since for this work these have been chosen according to their incidence and relevance in the information collected. Similarly, it is necessary to mention the existing discrepancy with the terms that intervene in a multi-agent system in the fields of study analyzed; although an agreement was established by naming the components of the model presented, future works could focus on improving this foundation, assigning terms capable of being used in more areas of application of MAS. Furthermore, future research may concentrate on the topic of interoperability and compatibility when integrating the FC-MAS model with existing or developing MAS technologies. This will entail consideration of potential integration issues or conflicts that may arise, as well as the necessity to balance MAS performance when system resource constraints or limitations have to be taken into account.

Being FC-MAS a new model based on the compilation of different works, no applications were found in which each and every one of the proposed components has been implemented. So the systems proposed in the case studies have been adjusted in such a way that all the components intervene in its operation. It can be noted that both the user and the tasks are not treated as explicit components of the model.

## REFERENCES

[1] S. Greengard, *The Internet of Things*. Cambridge, MA, USA: MIT Press, 2021.
[2] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," *J. Cleaner Prod.*, vol. 252, Apr. 2020, Art. no. 119869.
[3] B. W. Wirtz, J. C. Weyerer, and C. Geyer, "Artificial intelligence and the public sector—Applications and challenges," *Int. J. Public Admin.*, vol. 42, no. 7, pp. 596–615, May 2019.
[4] S. A. DeLoach, M. F. Wood, and C. H. Sparkman, "Multiagent systems engineering," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 11, no. 3, pp. 231–258, 2001.
[5] A. M. Uhrmacher and D. Weyns, *Multi-Agent Systems: Simulation and Applications*. Boca Raton, FL, USA: CRC Press, 2009.
[6] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern., C Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
[7] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 3, pp. 514–529, Sep. 2018.
[8] M. Oprea, "Applications of multi-agent systems," in *Information Technology*. Berlin, Germany: Springer, 2004, pp. 239–270.
[9] K. P. Sycara, "Multiagent systems," *AI Mag.*, vol. 19, no. 2, p. 79, 1998.
[10] J. Leite, A. Omicini, L. Sterling, and P. Torroni, *Declarative Agent Languages and Technologies: First International Workshop, DALT 2003, Melbourne, Australia, July 15, 2003, Revised Selected and Invited Papers*, vol. 2990. Berlin, Germany: Springer, 2004.
[11] V. Nunavath and A. Prinz, "Visualization of exchanged information with dynamic networks: A case study of fire emergency search and rescue operation," in *Proc. IEEE 7th Int. Advance Comput. Conf. (IACC)*, Jan. 2017, pp. 281–286.
[12] G. H. Merabet, M. Essaaidi, H. Talei, M. R. Abid, N. Khalil, M. Madkour, and D. Benhaddou, "Applications of multi-agent systems in smart grids: A survey," in *Proc. Int. Conf. Multimedia Comput. Syst. (ICMCS)*, Apr. 2014, pp. 1088–1094.
[13] D. Maldonado, J. Cepeda, S. Gamboa, J. A. Torres, and P. J. Cruz, "Optimum frequency response with decentralized drivers in power system," *IET Gener., Transmiss. Distribution*, vol. 17, no. 9, pp. 1985–1998, May 2023. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/gtd2.12774
[14] M.-T. Kuo and S.-D. Lu, "Design and implementation of real-time intelligent control and structure based on multi-agent systems in microgrids," *Energies*, vol. 6, no. 11, pp. 6045–6059, Nov. 2013.
[15] B. Wang, J. Wang, B. Zhang, and X. Li, "Global cooperative control framework for multiagent systems subject to actuator saturation with industrial applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1270–1283, Jul. 2017.
[16] B. Wang, J. Wang, B. Zhang, W. Chen, and Z. Zhang, "Leader–follower consensus of multivehicle wirelessly networked uncertain systems subject to nonlinear dynamics and actuator fault," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 492–505, Apr. 2018.
[17] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Gener. Comput. Syst.*, vol. 86, pp. 650–655, Sep. 2018.
[18] B. Wang, W. Chen, B. Zhang, and Y. Zhao, "Regulation cooperative control for heterogeneous uncertain chaotic systems with time delay: A synchronization errors estimation framework," *Automatica*, vol. 108, Oct. 2019, Art. no. 108486. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109819303383
[19] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4972–4983, Jun. 2017.
[20] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.
[21] H. Chuin Lau and L. Zhang, "Task allocation via multi-agent coalition formation: Taxonomy, algorithms and complexity," in *Proc. 15th IEEE Int. Conf. Tools Artif. Intell.*, 2003, pp. 346–350.
[22] J. Zhou, X. Zhao, X. Zhang, D. Zhao, and H. Li, "Task allocation for multi-agent systems based on distributed many-objective evolutionary algorithm and greedy algorithm," *IEEE Access*, vol. 8, pp. 19306–19318, 2020.
[23] A. M. Farid, "Multi-agent system design principles for resilient operation of future power systems," in *Proc. IEEE Int. Workshop Intell. Energy Syst. (IWIES)*, Oct. 2014, pp. 18–25.
[24] K. Potiron, A. E. F. Seghrouchni, and P. Taillibert, *From Fault Classification to Fault Tolerance for Multi-Agent Systems*, vol. 67. Cham, Switzerland: Springer, 2013.
[25] M. Wooldridge, *An Introduction to Multi-Agent Systems*. Hoboken, NJ, USA: Wiley, 2009.
[26] Y. Li and C. Tan, "A survey of the consensus for multi-agent systems," *Syst. Sci. Control Eng.*, vol. 7, no. 1, pp. 468–482, 2019.
[27] P. G. Balaji and D. Srinivasan, "An introduction to multi-agent systems," in *Innovations in Multi-Agent Systems and Applications-1*. Berlin, Germany: Springer, 2010, pp. 1–27.
[28] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
[29] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1999.
[30] A. Kantamneni, L. E. Brown, G. Parker, and W. W. Weaver, "Survey of multi-agent systems for microgrid control," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 192–203, Oct. 2015.
[31] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications—Part I: Concepts, approaches, and technical challenges," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1743–1752, Nov. 2007.
[32] S. T. Goonatilleke and B. Hettige, "Past, present and future trends in multi-agent system technology," *J. Européen des Systèmes Automatisés*, vol. 55, no. 6, pp. 723–739, Dec. 2022.

[33] M. H. Moradi, S. Razini, and S. Mahdi Hosseinian, "State of art of multiagent systems in power engineering: A review," *Renew. Sustain. Energy Rev.*, vol. 58, pp. 814–824, May 2016.
[34] A. Sujil, J. Verma, and R. Kumar, "Multi agent system: Concepts, platforms and applications in power systems," *Artif. Intell. Rev.*, vol. 49, no. 2, pp. 153–182, Feb. 2018.
[35] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: A brief overview," *Inf. Sci.*, vol. 380, pp. 117–131, Feb. 2017.
[36] H. Peng, J. Wang, S. Wang, W. Shen, D. Shi, and D. Liu, "Coordinated motion control for a wheel-leg robot with speed consensus strategy," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1366–1376, Jun. 2020.
[37] M. Haenlein and A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence," *California Manage. Rev.*, vol. 61, no. 4, pp. 5–14, Aug. 2019.
[38] C. E. Garcia, M. R. Camana, and I. Koo, "Machine learning-based scheme for multi-class fault detection in turbine engine disks," *ICT Exp.*, vol. 7, no. 1, pp. 15–22, Mar. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959521000096
[39] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Sci. Robot.*, vol. 5, no. 49, Dec. 2020, Art. no. eabe4385.
[40] J. R. Marden and J. S. Shamma, "Game theory and control," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, pp. 105–134, May 2018.
[41] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–33, 2020.
[42] W. Qian, Y. Gao, and Y. Yang, "Global consensus of multiagent systems with internal delays and communication delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 1961–1970, Oct. 2019.
[43] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-agent Systems With JADE*, vol. 7. Hoboken, NJ, USA: Wiley, 2007.
[44] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, and A. Drogoul, "Building, composing and experimenting complex spatial models with the GAMA platform," *GeoInformatica*, vol. 23, no. 2, pp. 299–322, Apr. 2019.
[45] H. Hamidi and A. Kamankesh, "An approach to intelligent traffic management system using a multi-agent system," *Int. J. Intell. Transp. Syst. Res.*, vol. 16, no. 2, pp. 112–124, May 2018.
[46] L. Cruz-Piris, D. Rivera, S. Fernandez, and I. Marsa-Maestre, "Optimized sensor network and multi-agent decision support for smart traffic light management," *Sensors*, vol. 18, no. 2, p. 435, Feb. 2018.
[47] F. Derakhshan and S. Yousefi, "A review on the applications of multiagent systems in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 5, May 2019, Art. no. 155014771985076.
[48] A. Barriuso, G. Villarrubia González, J. De Paz, Á. Lozano, and J. Bajo, "Combination of multi-agent systems and wireless sensor networks for the monitoring of cattle," *Sensors*, vol. 18, no. 2, p. 108, Jan. 2018.
[49] A. V. Nazarova and M. Zhai, "The application of multi-agent robotic systems for earthquake rescue," in *Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms*. Cham, Switzerland: Springer, 2020, pp. 133–146.
[50] A. Sharma, "Multi-agent system applications to fight COVID-19 pandemic," *Apollo Med.*, vol. 17, no. 5, p. 41, 2020.
[51] P. A. Cisse, J. M. Dembele, M. Lo, and C. Cambier, "Multi-agent systems for epidemiology: Example of an agent-based simulation platform for schistosomiasis," in *Agents and Multi-Agent Systems for Health Care*. Cham, Switzerland: Springer, 2017, pp. 131–153.
[52] B. M. Castro, Y. de Abreu de Melo, N. F. dos Santos, A. L. da Costa Barcellos, R. Choren, and R. M. Salles, "Multi-agent simulation model for the evaluation of COVID-19 transmission," *Comput. Biol. Med.*, vol. 136, Sep. 2021, Art. no. 104645.
[53] E. Estrada, *The Structure of Complex Networks: Theory and Applications*. London, U.K.: Oxford Univ. Press, 2012.
[54] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, Mar. 2001.
[55] J. Liu, Y. Pan, M. Li, Z. Chen, L. Tang, C. Lu, and J. Wang, "Applications of deep learning to MRI images: A survey," *Big Data Mining Analytics*, vol. 1, no. 1, pp. 1–18, Mar. 2018.
[56] M. W. Khan, J. Wang, M. Ma, L. Xiong, P. Li, and F. Wu, "Optimal energy management and control aspects of distributed microgrid using multi-agent systems," *Sustain. Cities Soc.*, vol. 44, pp. 855–870, Jan. 2019.
[57] L. A. Amaral and J. M. Ottino, "Complex networks," *Eur. Phys. J. B*, vol. 38, no. 2, pp. 147–162, 2004.
[58] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.
[59] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Phys. Rep.*, vol. 424, nos. 4–5, pp. 175–308, 2006.
[60] S. Pajevic and D. Plenz, "The organization of strong links in complex networks," *Nature Phys.*, vol. 8, no. 5, pp. 429–436, May 2012.
[61] K. You and L. Xie, "Network topology and communication data rate for consensusability of discrete-time multi-agent systems," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2262–2275, Oct. 2011.
[62] Y. Chen, J. Lu, X. Yu, and D. J. Hill, "Multi-agent systems with dynamical topologies: Consensus and applications," *IEEE Circuits Syst. Mag.*, vol. 13, no. 3, pp. 21–34, 3rd Quart., 2013.
[63] X. F. Wang, "Complex networks: Topology, dynamics and synchronization," *Int. J. Bifurcation Chaos*, vol. 12, no. 05, pp. 885–916, May 2002.
[64] M. T. Lazaro, L. M. Paz, P. Pinies, J. A. Castellanos, and G. Grisetti, "Multi-robot SLAM using condensed measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1069–1076.
[65] W. Gruszczyk and H. Kwasnicka, "Coalition formation in multi-agent systems—An evolutionary approach," in *Proc. Int. Multiconference Comput. Sci. Inf. Technol.*, Oct. 2008, pp. 125–130.
[66] J. Yang and Z. Luo, "Coalition formation mechanism in multi-agent systems based on genetic algorithms," *Appl. Soft Comput.*, vol. 7, no. 2, pp. 561–568, Mar. 2007.
[67] C. E. Garcia, M. R. Camana, and I. Koo, "Ensemble learning aided QPSO-based framework for secrecy energy efficiency in FD CR-NOMA systems," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 649–667, Sep. 2023.
[68] R. J. Alattas, S. Patel, and T. M. Sobh, "Evolutionary modular robotics: Survey and analysis," *J. Intell. Robotic Syst.*, vol. 95, nos. 3–4, pp. 815–828, Sep. 2019.
[69] P. Kravets, "Game method for coalitions formation in multi-agent systems," in *Proc. IEEE 13th Int. Scientific Tech. Conf. Comput. Sci. Inf. Technol. (CSIT)*, vol. 1, Sep. 2018, pp. 1–4.
[70] E. Lavendelis, A. Liekna, A. Nikitenko, A. Grabovskis, and J. Grundspenkis, "Multi-agent robotic system architecture for effective task allocation and management," in *Proc. 11th WSEAS Int. Conf. Electron., Hardware, Wireless Optical Commun., Proc. 11th WSEAS Int. Conf. Signal Process., Robot. Automat., Proc. 4th WSEAS Int. Conf. Nanotechnol.*, Cambridge, U.K., 2012, pp. 167–174.
[71] L. Wang, Z. Wang, S. Hu, and L. Liu, "Ant colony optimization for task allocation in multi-agent systems," *China Commun.*, vol. 10, no. 3, pp. 125–132, Mar. 2013.
[72] L. Kocarev, *Consensus and Synchronization in Complex Networks*. Berlin, Germany: Springer, 2013.
[73] Y. Zheng, Y. Zhu, and L. Wang, "Consensus of heterogeneous multi-agent systems," *IET Control Theory Appl.*, vol. 5, no. 16, pp. 1881–1888, 2011.
[74] S. Das and J. M. Moura, "Distributed state estimation in multi-agent networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 4246–4250.
[75] C. Liu, Y. Wang, D. Zhou, and X. Shen, "Minimum-variance unbiased unknown input and state estimation for multi-agent systems by distributed cooperative filters," *IEEE Access*, vol. 6, pp. 18128–18141, 2018.
[76] A. Shariati and M. Tavakoli, "A descriptor approach to robust leader-following output consensus of uncertain multi-agent systems with delay," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5310–5317, Oct. 2017.
[77] S. I. Han, "Prescribed consensus and formation error constrained finite-time sliding mode control for multi-agent mobile robot systems," *IET Control Theory Appl.*, vol. 12, no. 2, pp. 282–290, Jan. 2018.
[78] D. N. Serpanos and A. Bouloutas, "Centralized versus distributed multimedia servers," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1438–1449, Nov. 2000.
[79] T. Van Gog, F. Paas, W. Savenye, R. Robinson, M. Niemczyk, R. Atkinson, T. E. Johnson, D. L. O'Connor, R. M. Rikers, and P. Ayres, "Data collection and analysis," in *Handbook of Research on Educational Communications and Technology Third Edition*. Taylor and Francis, 2008, pp. 763–806, doi: 10.4324/9780203880869-61.
[80] M. D. LeCompte, "Analyzing qualitative data," *Theory Pract.*, vol. 39, no. 3, pp. 146–154, 2000.
[81] S. Brandt and S. Brandt, *Data Analysis*. Cham, Switzerland: Springer, 1998.
[82] N. B. Harmancioglu, O. Fistikoglu, S. D. Ozkul, V. P. Singh, and M. N. Alpaslan, "Definition of monitoring objectives," in *Water Science and Technology Library*. Dordrecht, The Netherlands: Springer, 1999, pp. 95–110, doi: 10.1007/978-94-015-9155-3_5.

[83] K. Fatema, V. C. Emeakaroha, P. D. Healy, J. P. Morrison, and T. Lynn, "A survey of cloud monitoring tools: Taxonomy, capabilities and objectives," *J. Parallel Distrib. Comput.*, vol. 74, no. 10, pp. 2918–2933, Oct. 2014.
[84] A. Arning, R. Agrawal, and P. Raghavan, "A linear method for deviation detection in large databases," in *Proc. KDD*, 1996, vol. 1141, no. 50, pp. 972–981.
[85] E. Toth and S. Chawla, "Group deviation detection methods: A survey," *ACM Comput. Surveys*, vol. 51, no. 4, pp. 1–38, Jul. 2019.
[86] S. P. Kavulya, K. Joshi, F. D. Giandomenico, and P. Narasimhan, "Failure diagnosis of complex systems," in *Resilience Assessment and Evaluation of Computing Systems*. Berlin, Germany: Springer-Verlag, 2012, pp. 239–261, doi: 10.1007/978-3-642-29032-9_12.
[87] C.-H. Lin, H.-J. Chuang, C.-S. Chen, C.-S. Li, and C.-Y. Ho, "Fault detection, isolation and restoration using a multiagent-based distribution automation system," in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, 2009, pp. 2528–2533.
[88] T. J. Leung, J. H. Rife, P. Seiler, and R. Venkataraman, "Comparison of fault-tree models for fault detection, isolation, and recovery algorithms," *J. Aerosp. Inf. Syst.*, vol. 14, no. 9, pp. 517–522, Sep. 2017.
[89] M. Tipaldi and B. Bruenjes, "Survey on fault detection, isolation, and recovery strategies in the space domain," *J. Aerosp. Inf. Syst.*, vol. 12, no. 2, pp. 235–256, Feb. 2015.
[90] G. W. Clark, M. V. Doran, and T. R. Andel, "Cybersecurity issues in robotics," in *Proc. IEEE Conf. Cognit. Comput. Aspects Situation Manage. (CogSIMA)*, Mar. 2017, pp. 1–5.
[91] Y. Zhang, M. E. Raoufat, and K. Tomsovic, "Remedial action schemes and defense systems," in *Smart Grid Handbook*. Wiley, 2016, pp. 1–10, doi: 10.1002/9781118755471.sgd032.
[92] J. Wen, P. Arons, and W. E. Liu, "The role of remedial action schemes in renewable generation integrations," in *Proc. Innov. Smart Grid Technol. (ISGT)*, 2010, pp. 1–6.
[93] J. A. Torres, P. J. Cruz, R. Vizuete, and R. Fierro, "On resilience and heterogeneity in robotic networks," in *Cooperative Localization and Navigation*. Boca Raton, FL, USA: CRC Press, 2019, pp. 141–158.
[94] J. Lin, K. Dzeparoska, S. Q. Zhang, A. Leon-Garcia, and N. Papernot, "On the robustness of cooperative multi-agent reinforcement learning," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, 2020, pp. 62–68.
[95] Y. Wang and H. Chen, "Blockchain: A potential technology to improve the performance of collaborative emergency management with multi-agent participation," *Int. J. Disaster Risk Reduction*, vol. 72, Apr. 2022, Art. no. 102867. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212420922000863
[96] A. Khakimov, I. A. Elgendy, A. Muthanna, E. Mokrov, K. Samouylov, Y. Maleh, and A. A. A. El-Latif, "Flexible architecture for deployment of edge computing applications," *Simul. Model. Pract. Theory*, vol. 114, Jan. 2022, Art. no. 102402.
[97] C. Wang, P. Ju, F. Wu, X. Pan, and Z. Wang, "A systematic review on power system resilience from the perspective of generation, network, and load," *Renew. Sustain. Energy Rev.*, vol. 167, Oct. 2022, Art. no. 112567. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032122004658
[98] J. Los, F. Schulte, M. T. J. Spaan, and R. R. Negenborn, "An auction-based multi-agent system for the pickup and delivery problem with autonomous vehicles and alternative locations," in *Dynamics in Logistics*, M. Freitag, A. Kinra, H. Kotzab, and N. Megow, Eds. Cham, Switzerland: Springer, 2022, pp. 244–260.
[99] J. J. Damanik, H. Kasan, and H.-L. Choi, "Solving delivery assignment in hybrid-transit network using multi-agent reinforcement learning," in *Robot Intelligence Technology and Applications*, vol. 6, J. Kim, B. Englot, H.-W. Park, H.-L. Choi, H. Myung, J. Kim, and J.-H. Kim, Eds. Cham, Switzerland: Springer, 2022, pp. 485–497.
[100] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.
[101] S. M. Ghavami, J. Maleki, and T. Arentze, "A multi-agent assisted approach for spatial group decision support systems: A case study of disaster management practice," *Int. J. Disaster Risk Reduction*, vol. 38, Aug. 2019, Art. no. 101223. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212420919303218
[102] N. Firdausiyah, E. Taniguchi, and A. G. Qureshi, "Modeling city logistics using adaptive dynamic programming based multi-agent simulation," *Transp. Res. E, Logistics Transp. Rev.*, vol. 125, pp. 74–96, May 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554518309049
[103] R. Limosani, R. Esposito, A. Manzi, G. Teti, F. Cavallo, and P. Dario, "Robotic delivery service in combined outdoor–indoor environments: Technical analysis and user evaluation," *Robot. Auto. Syst.*, vol. 103, pp. 56–67, May 2018.
[104] M. W. Khan and J. Wang, "The research on multi-agent system for microgrid control and optimization," *Renew. Sustain. Energy Rev.*, vol. 80, pp. 1399–1411, Dec. 2017.
[105] A. Farinelli, N. Boscolo, E. Zanotto, and E. Pagello, "Advanced approaches for multi-robot coordination in logistic scenarios," *Robot. Auto. Syst.*, vol. 90, pp. 34–44, Apr. 2017.
[106] B. Arbanas, A. Ivanovic, M. Car, T. Haus, M. Orsag, T. Petrovic, and S. Bogdan, "Aerial-ground robotic system for autonomous delivery tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5463–5468.
[107] B. Bona, L. Carlone, M. Indri, and S. Rosa, "Supervision and monitoring of logistic spaces by a cooperative robot team: Methodologies, problems, and solutions," *Intell. Service Robot.*, vol. 7, no. 4, pp. 185–202, Oct. 2014.
[108] G. Prennushi, G. Rubio, and K. Subbarao, "Monitoring and evaluation," in *A Sourcebook for Poverty Reduction Strategies*. Washington, DC, USA: World Bank, 2002, pp. 30–107.
[109] H. Farivarnejad and S. Berman, "Multirobot control strategies for collective transport," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 5, no. 1, pp. 205–219, May 2022, doi: 10.1146/annurev-control-042920-095844.
[110] F. M. Gonzalez-Longatt and J. L. Rueda, *PowerFactory Applications for Power System Analysis* (Power Systems). Springer, 2014.
[111] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.
[112] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
[113] R.-A. Luchian, S. Rosioru, I. Stamatescu, I. Fagarasan, and G. Stamatescu, "Enabling industrial motion control through IIoT multi-agent communication," in *Proc. IECON 47th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2021, pp. 1–6.
[114] V. Lesi, Z. Jakovljevic, and M. Pajic, "Reliable industrial IoT-based distributed automation," in *Proc. Int. Conf. Internet Things Design Implement.*. New York, NY, USA: Association for Computing Machinery, Apr. 2019, pp. 94–105, doi: 10.1145/3302505.3310072.
[115] R. Moghadam and H. Modares, "Resilient adaptive optimal control of distributed multi-agent systems using reinforcement learning," *IET Control Theory Appl.*, vol. 12, no. 16, pp. 2165–2174, Nov. 2018.

**DIEGO MALDONADO** received the engineering degree in electronics and control and the master's degree in smart grids from the Escuela Politecnica Nacional, Ecuador, in 2014 and 2020, respectively, where he is currently pursuing the Ph.D. degree (researcher) in the field of distributed control. He is an Assistant Professor with the Departamento de Automatizacion y Control, where he has taught electronics and automation engineering degree. His research interests include automatic systems and distributed control applied to power systems.

**EDISON CRUZ** received the engineering degree in electronics and control from the Escuela Politecnica Nacional, in 2020. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer, and Biomedical Engineering, Southern Illinois University. His research interests include multi-agent systems, optimal control, and magnetic field manipulation using permanent magnets.

**JACKELINE ABAD TORRES** received the B.S. degree from the Escuela Politecnica Nacional, Quito, Ecuador, in 2008, and the M.S. and Ph.D. degrees in electrical engineering from Washington State University, in 2012 and 2014, respectively. She held a Fulbright Grant, in 2010. She is currently an Associate Professor with the Departamento de Automatizacion y Control Industrial, Facultad de Ingenieria Electrica y Electrnica, Escuela Politecnica Nacional. Her current research interests include structural analysis and controller design of dynamical networks with applications to sensor/vehicle networking, epidemic control, and power systems network control.

**PATRICIO J. CRUZ** (Member, IEEE) received the B.S. degree in electronic and control engineering from the Escuela Politecnica Nacional (EPN), Quito, Ecuador, in 2005, and the M.Sc. degree in electrical engineering and the Ph.D. degree in engineering from The University of New Mexico (UNM), Albuquerque, NM, USA, in 2012 and 2016, respectively. During his graduate studies, he was a Research Assistant with the Multi-Agent, Robotics, and Heterogeneous Systems (MARHES) Laboratory, UNM. From 2019 to 2022, he was a Coordinator of the Master Programs with the School of Electrical Engineering, EPN. He is currently an Associate Professor with the Departamento de Automatizacion y Control Industrial (DACI), EPN. He has been the Director and a Collaborator of different research projects in areas, such as multi-agent system coordination, control of unmanned aerial systems (UAVs), and the development of platforms for rehabilitation robotics. He is the author or coauthor of more than 25 papers in international and regional conferences and journals. As part of his work, he published two book chapters focused on cooperative control of robotic heterogeneous systems.

**SILVANA DEL PILAR GAMBOA BENITEZ** received the degree in electronics and control engineering and the master's degree in control systems from the Escuela Politecnica Nacional, Ecuador, in 2004 and 2008, respectively, and the Ph.D. degree in electrical engineering from the Electrical Energy Institute, Universidad Nacional de San Juan, Argentina, in 2018. She is currently a Professor with the Faculty of Electrical and Electronic Engineering, Escuela Politecnica Nacional. She is an Administrator of the Industrial Networks and the SCADA Laboratory. Her research interests include monitoring, protection, and control systems, SCADA, communications networks, and their application in industrial processes and in power systems.

• • •