IEEE Access
Multidisciplinary ⁞ Rapid Review ⁞ Open Access Journal

Received January 29, 2019, accepted February 17, 2019, date of publication March 7, 2019, date of current version March 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903629

A Distributed Multi-Agent Framework for Resilience Enhancement in Cyber-Physical Systems

FÁBIO JANUÁRIO[1,2], ALBERTO CARDOSO[3], AND PAULO GIL[1,2,3]

[1]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, Campus de Caparica, 2829-516 Caparica, Portugal
[2]Centre of Technology and Systems, UNINOVA, Campus de Caparica, 2829-516 Caparica, Portugal
[3]Centre for Informatics and Systems, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal

Corresponding author: Fábio Januário (f.januario@campus.fct.unl.pt)

This work was supported in part by the Fundação para a Ciência e Tecnologia, Portugal, under Grant SFRH/BD/85586/2012, Project UID/EEA/00066/2013, and Project UID/CEC/00326/2013.

ABSTRACT Cyber-physical systems (CPSs) are nowadays an important component of most industrial infrastructures and materialize the integration of control systems with advanced information technologies. Commonly, they aggregate distinct communication platforms and networked devices or nodes with different capabilities and goals. The resulting increase in complexity has inevitably brought into playing new challenges, namely, on the physical world and cyber space. In these systems, the development and gaining of state awareness and how to deal with the inherent vulnerabilities of the overall system are essential and also challenging. In this paper, the problem of resilience enhancement in CPSs is addressed based on a hierarchical multi-agent framework that is implemented over a distributed middleware, in which each agent carries out specific tasks. Physical and cyber vulnerabilities are taken into account, and state and context awarenesses of the whole system are targeted. The proposed framework ensures a minimum level of acceptable performance in case of physical disturbances and malicious attacks, as demonstrated by experiments on an IPv6-based test-bed.

INDEX TERMS Resilience, context awareness, distributed system, multi-agents, performance.

I. INTRODUCTION

Distributed networked-based controllers are becoming widely used in industrial plants, such as power generation units, oil refineries and chemical plants [1]. This technology consists of a wide range of heterogeneous devices with several levels of resources, which are interconnected through networking infrastructures [2]. The migration of these systems into a cyber space, bridging the cyber world of computing and communication with the physical world, led to a new paradigm, commonly referred to as Cyber-Physical Systems (CPSs) [3]. Essentially, CPSs consist of the integration of computing technologies, networking infrastructures and exogenous systems, aiming to monitor and control physical processes [4].

A typical CPS architecture comprises two main layers, including the physical layer and the cyber counterpart. The physical layer materializes an intelligent network of sensors and actuators, which is required to collect information from the environment and to actuate on a given physical system, whereas the cyber layer represents the decision-making framework and the communication infrastructure [5]. In large industrial control systems, the cyber layer is typically composed of a Supervisory Control and Data Acquisition (SCADA) system [6].

The integration orchestrated and mediated by CPSs poses a number of challenges in the context of supervision systems. Although these systems can allow the accommodation of some uncertainties and disturbances, in particular on the physical part, this is not the case with cyber space disturbances. Hence, a dedicated framework is needed for dealing with cyber, cognitive and human complex interdependencies, which altogether enhance the potential for faulty events, malfunctions and failures, and tend to exacerbate security vulnerabilities [7]. As such, critical vulnerabilities should be firstly identified and taken into account when designing a

The associate editor coordinating the review of this manuscript and approving it for publication was Mengchu Zhou.

supervision system, which should be provided with efficient processing of information and allow a correct assessment of the system behavior.

Security in CPSs has become a major concern in the last few years [8]. Malicious actors may exploit cyber vulnerabilities in order to mask the degradation of physical systems or injecting incorrect data to decision-making levels, regarding, for instance, the current system status or other critical functional information [9], [10]. Some reported malicious attacks on CPSs have shown that traditional protection/security mechanisms are not robust enough to accommodate or mitigate attacks. It should be mentioned that the current distributed networked control systems have not been designed to inherently include effective measures to deal with cyber attacks. They just rely on their supposedly anonymity to remain safe (see e.g. [11]–[13]).

This new class of control problems, however, requires a holistic and cross-layered approach, incorporating protection mechanisms in the design stage to deal with physical faults and, particularly, cyber vulnerabilities within the overall system and thus enhancing the networked supervision robustness.

The concept of resilience has been proposed to deal with this kind of problems [14], [15]. In the context of CPSs, resilience stresses the ability to accommodate faults or events, which otherwise may compromise the stability of the system and the underlying teleonomy [16]. Moreover, in the design stage, resilient frameworks should also consider all possible threats, namely physical and cyber threats, while maintaining an acceptable level of operational normalcy [17]. Here, threats are events that can hamper operational normalcy, destabilize networked control systems and degrade the closed loop system performance, including malicious attacks, human errors and complex latencies, along with interdependencies, network component malfunctions or communication breakdowns [18]. In this context, a given resilient networked control system over a distributed communication network is expected to be kept in operation with an acceptable performance, even in the case of faults, disruptions or attacks. This refers to the ability to ensure that system outputs are within acceptable operating thresholds, and that the normal operation of the system can ultimately be restored. If, despite of a local fault or attack, the system is capable of maintaining a given admissible performance, the whole system is referred as being resilient.

The association of Multi-Agent Systems (MASs) with CPSs and distributed control has gained considerable attention in the last few years [19], [20]. In literature one can find a number of surveys reporting recent advances in this context, along with theoretical design principles (see e.g. [21]–[23]). Additionally, the need for resilience enhancement has been stressed in some studies [24]–[26].

With respect to resilience compliance in CPSs, some research directions rely on centralized architectures, making use of machine learning techniques, such as fuzzy logic, neural networks and data mining and also knowledge fusion. In [27], a centralized architecture based on a neuro-fuzzy data fusion engine is proposed to improve state awareness. It provides real-time monitoring and analysis of complex critical systems. Input data from multiple sources are fused and combined into a robust anomaly indicators. Additionally, a neural network-based signal prediction is used to provide state-awareness in temporary unavailability of sensors. Centralized approaches are, however, somehow difficult to implement in heterogeneous environments, in which communication channels can also suffer from malfunctions and other disruptive events.

As for decentralized architectures, resilience can be improved based on models of system components, along with data quality metrics [28], [29], or by relying on agent-based techniques [30]–[32]. An autonomous decentralized resilient monitoring system able to dynamically adapt and to be reconfigured, depending on current conditions, is proposed in [28]. In this work, a data quality measure is associated with sensors, and process variables are estimated based on sensor measurements. This framework, however, requires modeling all the components of the system, as well as data quality metrics, which can be difficult to fulfill in a real distributed heterogeneous network. In [32], an integrated diagnostic and control strategy is proposed, founded on an agent-based design to ensure resilience in terms of stability and efficiency. This paper proposes a theoretical framework based on a three layer architecture, being its functionalities, hardware independence and intelligence also discussed. The proposed intelligent techniques may, however, be difficult to implement in some components of a heterogeneous network, such as on wireless nodes.

The problem of resilience enhancement in Wireless Sensor Networks (WSNs) has also been considered in several works (see e.g. [33], [34]). In [33], an intelligent resilient control architecture for a wireless networked control system is proposed. It focusses on the quality of control, by relying on an intelligent resilient control algorithm that ensures operational normalcy in face of wireless interference events, such as radio frequency jamming or signal blocking. This algorithm, however, takes only into account the WSN part, while the other components of a CPS are neglected.

Although several approaches have been suggested for improving resilience, many of them only focus on specific issues and scenarios, not taking into account the system as a whole. In addition, some of them are only concerned with physical aspects of the exogenous system, by applying Fault Detection and Isolation (FDI) specific algorithms and heuristics [28], [29], while others just address particular aspects of the cyber part [35], [36]. All these approaches have limitations in terms of implementation over a heterogeneous and distributed environment, where components do not possess the same resources. Finally, it should be mentioned that, to the best of the authors' knowledge, it is not straightforward to find empirical studies regarding the implementation of resilience enhancement techniques on CPSs. Considering the

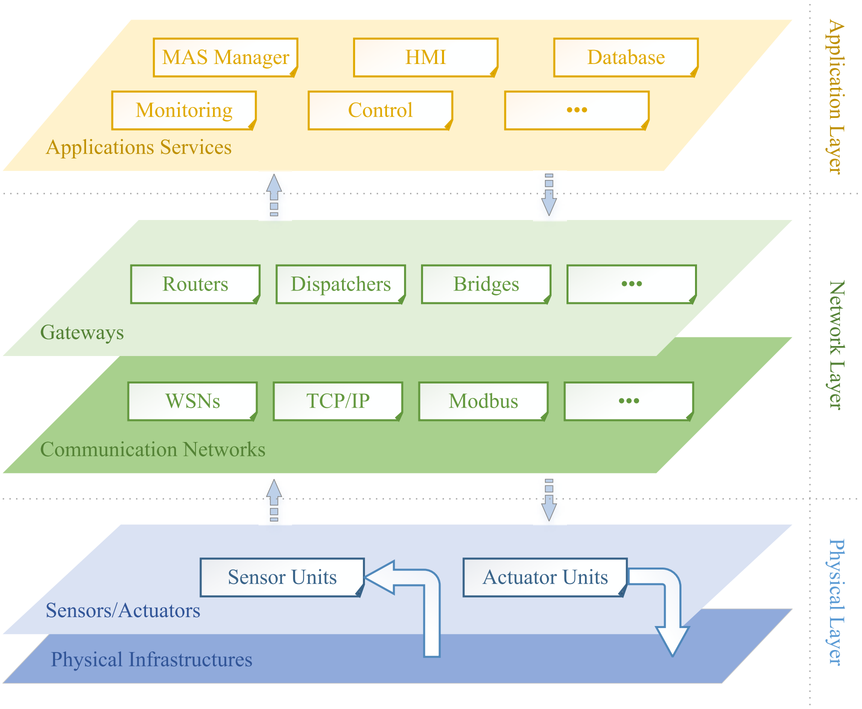

FIGURE 1. Architecture of a cyber-physical system.

limitations of current methodologies, there is a clear lack of solutions not only for deploying the required mechanisms to specific devices and hardware, but also to the system under monitoring and control.

This paper proposes enhancement resilience framework for complex CPSs, consisting of a diversity of distributed physical devices, in the context of heterogeneous communication networks. The approach makes use of a MAS embedded on a distributed middleware. Each agent is tailored for executing specific tasks, adapting its behavior and reacting accordingly, depending on its location and environmental changes. The architecture includes dedicated functionalities to keep a permanent awareness of the status and context of the overall system. The proposed architecture is assessed through experiments carried out on a IPv6-based CPS test-bed comprising several distributed devices.

The rest of this paper is organized as follows. Section II discusses the architecture of CPSs, including the main security vulnerabilities and threads. Section III describes the proposed approach for resilience enhancement and discusses the underlying multi-agent architecture. Section IV presents a case study based on a test-bed comprising a benchmark three-tank system and several distributed devices, while in Section V the main conclusions are drawn.

II. CYBER-PHYSICAL SYSTEMS

Cyber-Physical Systems consist of the integration of computational systems, networking and physical processes. These architectures are commonly represented by three main layers, including the physical, network and application layers, as sketched in Fig. 1. The physical layer can be split further into two layers. One representing the physical infrastructure, which comprises the exogenous system under control or monitoring, and another associated with sensors and actuators, which hosts all the required devices responsible for reading the environment and delivering control actions. The network layer implements the data transmission and mediates the interaction between the application layer and physical layer. On the other hand, the communication infrastructure supports communication between components of the system, including sensors, actuators and other integrated devices. Given the heterogeneity of these systems, different communication networks can coexist, such as, for instance, distributed low power wireless networks or TCP/IP networks, which are virtualised through the underlying network layer. These heterogeneous networks are interconnected through gateways, which are represented by the gateway layer. This layer can, in addition, include dispatchers to coordinate the communications over Wireless Sensor and Actuator Networks (WSANs) or routers to relay information delivered to destination devices. Finally, the application layer aggregates all the distributed applications found in a given CPS. It concerns software packages devoted to specific functions, such as for monitoring and remote control, management of network and agents, databases and Human Machine Interface (HMI) platforms, just to name out a few. In essence, they provide a user with several functionalities, allowing the abstraction of received data and enabling, somewhat transparently, the interaction between networks, devices and physical infrastructures.

TABLE 1. Security threats in cyber-physical systems layers [39]–[41].

Physical layer	Network layer	Application layer
Denial of Service (DoS)		
Physical attack	Rounting attack	Viruses, Trojan horses
Equipment failure	Sink node attack	Forged control commands
Electromagnetic interference	Black hole attack	Malicious code
Data tampering	Flooding attack	Privacy data leaking
Data intercept	Sinkhole attack	
Sybil attack	Wormhole attack	
Passive attack	Selective forwarding	

A. SECURITY VULNERABILITIES

The integration of many different devices in a CPS inherently has the potential of increasing the overall system vulnerability to faults and malicious attacks. In the following, some of the main vulnerabilities in a general CPS are discussed (see e.g. [37], [38]).

1) ISOLATION ASSUMPTION

Conventional design approach is propped up on the reliability and dependability paradigm. According to this concept, security issues in CPSs are not taken into account, as they are considered isolated from external environments, resulting from that there are no communication points with non-local networks. However, current CPSs are no longer shielded from external information sharing and, most importantly, they increasingly rely on open standard protocols, such as TCP/IP. Therefore, CPSs are inevitably becoming more and more vulnerable to intrusions and malicious attacks.

2) EXTENDED CONNECTIVITY

The growing need for interconnecting CPSs devices has led to the incorporation of services that commonly rely on open networks and wireless technologies. The combination of proprietary protocols designed for closed networks, together with standard open protocols and shared networks, have given rise to additional security vulnerabilities. Malicious attackers, by exploiting protocol and network vulnerabilities, can effectively target and compromise CPSs.

3) HETEROGENEITY

A critical issue regarding CPSs vulnerabilities involves the heterogeneity of embedded blocks. Different hardware components, such as sensors, actuators and sub-systems, together with software packages of different nature, either proprietary or commercial, for monitoring and controlling are typically incorporated in CPSs. As a result, each component and the corresponding integration, has its own security fragilities, which impact the overall CPS resilience.

4) MULTIPLE ACCESS

In a CPS, the generation, use and modification of data can be concurrently executed by multiple applications. This leads to an additional source of exposures, in terms of access control and authorization mechanisms. Moreover, the presence of a large number of remotely accessible field devices, which can be compromised by hacking actions, is another real concern.

5) PHYSICAL EXPOSURE

The physical exposure of CPS field devices, including sensors and actuators, deployed over large areas is itself a vulnerability. In case of inadequate physical surveillance, these devices may become vulnerable to physical tampering or even sabotage.

B. SECURITY THREATS

Taking into account the aforementioned vulnerabilities, CPSs are inherently a target for cyber and physical attacks. Table 1 shows, by layer, the main known attacks on CPSs [39]–[41].

Recent literature on this subject shows that the main incidents in cyber-physical infrastructures have occurred at the application layer level. Al-Mhiqani *et al.* [40] reported eighteen different CPS significant incidents between 2010 and 2017. The majority of these security events are associated with viruses, target attacks and account hijacking. These attacks intended to disrupt systems and extract sensitive data. The same conclusions can be drawn from [42], in which 359 industrial cyber-physical companies (e.g. manufactures, oil and gas, energy, government) in 21 countries across the world were interviewed. They reported that the main cause of incidents are malware/viruses, followed by target attacks.

1) PHYSICAL LAYER

Devices, including sensors and actuators, are in many cases deployed on large and unattended environments. On the other hand, communication capability, storage and data processing power of these nodes are somewhat limited. As such, traditional security mechanisms cannot be directly implemented on the physical layer, which makes it vulnerable to hacking and other kinds of malicious attacks. In addition, from the physical point of view, devices can be damaged by a physical attack, or their performance impacted due to equipment malfunction. Electromagnetic interference is another issue that can compromise the quality of raw data. Moreover, cyber attacks can also occur at this level. For instance, a Denial of Service (DoS) attack can be directed to any networked node, implying that the targeted device might stop providing reliable services due to network bandwidth overconsumption [43]. Data interception by hackers can also take place in the physical layer. In this case, an actor would intercept the communication channel and access transmitted data, while in a tampering event an attacker would modify a message. A sybil attack, on the other hand, compels a single device to have multiple identities, while a passive attack results in a collection of data by means of a sniffing method [39]. As such, security in the physical layer should involve the physical security of infrastructures, protection of collected data and execution of commands.

2) NETWORK LAYER

The network layer supports communication between devices. When the amount of data being relayed from node to node or a sink exceeds the capacity of a link, it leads to network congestion, and the overall system becomes partially compromised. Besides, gateway authentication and security policies between heterogeneous networks will also pose security issues in this layer.

The main sort of attacks consists of interrupting or forwarding the transmission of data. A routing attack interferes with the routing process, by sending fake routing information. A sink node attack aims to interrupt/block data transmission between networks by compromising a sink device, while in a black hole attack, the compromised device establishes routing connections with other devices, and next discards received packets, thus leading to a packet loss event [44]. In a flooding attack, the malicious attacker aims at exhausting the resources of network servers through a distributed DoS attack. In a sinkhole attack, a compromised device will try to attract all possible traffic from a particular region to itself. In a wormhole attack a malicious node will tunnel the packets it receives over a separate low latency channel to another point in the network. As such, the packets are sent from one malicious node to the other using this side channel [41]. Finally, in a selective forwarding attack, a malicious device deliberately looses some or all received packets. Given the above issues, it is crucial to ensure the security of communications in the network layer, including data integrity, confidentiality and consistency.

3) APPLICATION LAYER

The application layer is responsible for decision-making-based tasks and for providing actuating commands to specific nodes. It includes a wide range of applications, with various operating systems and security needs. The major security threats stem from malicious software and privacy requirements. The privacy of data in transmission, storage and presentation is a critical requirement [39]. This demands a stringent access control policy and authentication mechanisms, in order to ensure the protection of whole system. On the other hand, viruses, trojan horses and other sort of malware code in the application level can have a catastrophic impact on the physical layer [40]. In addition, malicious attackers can use or damage the system by forging control commands.

III. RESILIENT ARCHITECTURE

Providing resilience to a CPS implies its operation under an acceptable level of normalcy in response to disturbances, including malicious attacks. In addition, the whole system should be robust to faults and ‘‘thrive’’ or survive, even in the aftermath of a successful attack. This requires maintaining the awareness of the system status and the underlying context.

The proposed architecture for resilience enhancement is propped up on a distributed middleware, incorporating a MAS with the implementation of dedicated algorithms. The benefits of using agents within a distributed middleware framework has already been pointed out by the authors, although in a simplified way [45]. In the present work, that conceptual architecture is further extended, presenting the main agents and the underlying groups, which compose the framework, as well as their inherent functionalities.

In this framework, three types of agents are considered. Some are dedicated agents, tailored for executing specific and coordinated tasks, depending on the location they are deployed, and allowing the system to recover from specific malfunctions. Others are shared agents, which are present in all devices and layers. They guarantee the access to global properties, such as the maintenance of the awareness of physical and cyber contexts of the system. Finally, master agents allow the coordination of all subordinate agents located on each device. Given the distributed nature of these systems, the incorporation of agents provides flexibility in implementing functionalities, wherever they may be needed. In case a malfunction should be detected, the entire MAS has the ability to react accordingly in order to provide or strengthen the CPS resilience, while guaranteeing the safety status of the system, until the problem is completely fixed. The MAS is deployed along with the middleware, and uses collected and generated data to infer the context and the status of the underlying system. In the case of a compromising event, it reacts to ensure the security, integrity and privacy of data. Fig. 2 shows an overview of the proposed architecture.

As devices in a CPS commonly do not possess the same features, each component has its own middleware, which includes a set of agents with specific functionalities, distributed throughout the three main layers, depending on the layer. They are configured to provide the required functionalities, so as to cope with recognized vulnerabilities and to deal with malicious attacks.

Agents are grouped into five main groups: robust control agents, which ensure physical security and data quality; cyber security agents aiming to maintain the privacy and integrity of transmitted data; network reliability agents that ensure the correct operation of the communication infrastructure; MAS manager agents, which allow the management of the implemented MAS; master agent, which ensures proper communication between subordinate agents. In the following,

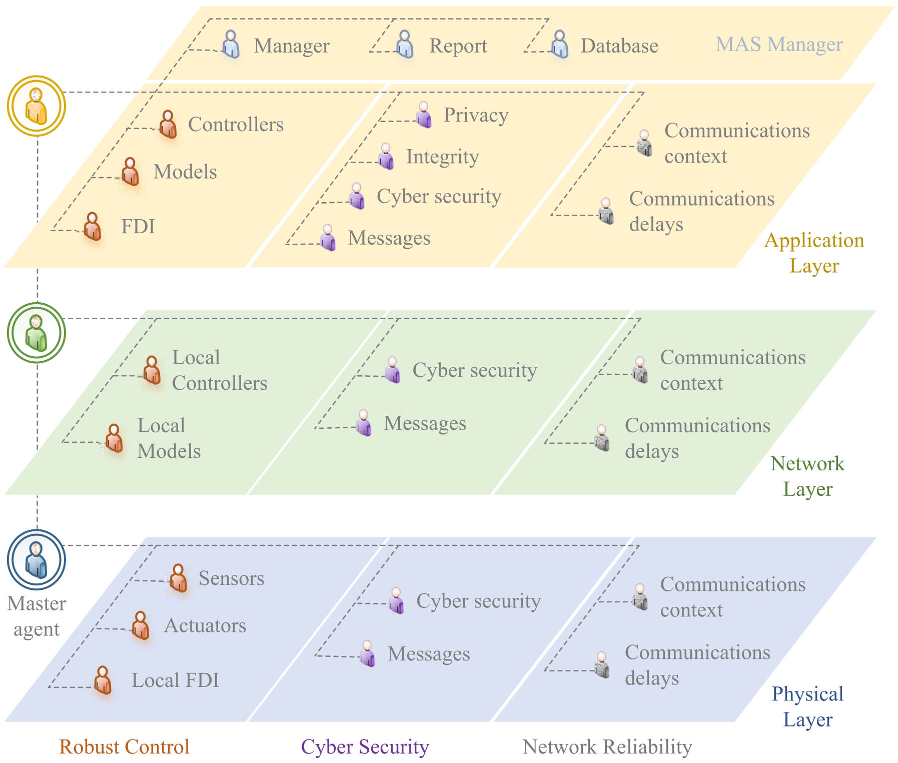

FIGURE 2. Proposed architecture.

these five groups are described, detailing by layer their main features and functionalities.

A. ROBUST CONTROL AGENTS

The robust control group comprises dedicated agents that are tailored to enhance the resilience with respect to control and monitoring of a physical systems, and guaranteeing the physical security and data quality. Remote robust and adaptive controllers are implemented through the underlying agents on the application layer. This layer can also host other robust control agents, such as those regarding the system models and system identification tools, which are used to predict the behavior of physical processes. In addition, it can also include Fault Detection and Isolation (FDI) mechanisms for ensuring data quality and physical security of sensors and actuators. The network layer incorporates local agents under the form of controllers and models that allow safety control during application malfunctions or unavailability. In this operation mode, a direct communication is established with available sensors and actuators, to ensure a minimum level of operational normalcy. Finally, the physical layer has deployed sensors and actuators agents to enable the interaction with exogenous systems. At this level, some FDI algorithms can be locally implemented, depending on the devices, namely for detecting and accommodating outliers.

B. CYBER SECURITY AGENTS

The cyber security agents group is devoted to deal with prior identified vulnerabilities of a device, on which they are deployed. Each layer includes a cyber security agent that implements defense cyber mechanisms and provides dedicated attack detection tools. In addition, another security level is implemented within the messages agent, which checks the structure and content of every transmitted message, in order to detect possible malicious attacks.

C. NETWORK RELIABILITY AGENTS

The network reliability agents group is composed of shared agents that cover all networked devices and aims to ensure the awareness of the whole communication infrastructure. In particular, the communication context agent is responsible for assessing the state of the communication infrastructure. These kind of agents exchange information among them, in order to get awareness in real time of the status of communications regarding all devices. Communications delays agents check the time stamp of sent and received messages for detecting possible deviations and anomalies.

D. MULTI-AGENT SYSTEM MANAGER AGENTS

The MAS manager group is only deployed at the application level. This group is composed of agents that enable the configuration and monitoring of the MAS. The manager agent is responsible for the management of the entire MAS, while the report agent provides the user with information about the current MAS status. The database agent allows the storage of important information. Besides, this agent can also have a database of the deployed agents to allow the remote programming by the manager agent, if a given problem with an agent is detected.

E. MASTER AGENTS

All devices have a master agent, which is responsible for ensuring the correct communication between subordinate agents and other master agents. The master agent is also responsible for guaranteeing that subordinate agents are working properly. An agent malfunction may impact the performance of the whole system. For this purpose, the master agent periodically verifies the communication status with subordinate agents, in order to check whether they are available. Algorithm 1 sketches the general main functions associated with master agents. The main events that can occur are presented in Table 2. As can be seen, the behavior of the master agent depends on inputs from subordinate agents. Whenever a master agent receives a message from a subordinate agent, or from an external agent, a security check is carried out within the cyber security group, in order to guarantee the integrity and conformity of received messages. In addition, other dedicated security agents work in the background, and whenever a security event is detected, they report the event to the corresponding master agent. If a given master agent needs to send data to another device, subordinate agents in the network reliability group will inform the corresponding master agent about which devices are working properly on the network, so as to decide where to send the data. Finally, master agents are also responsible for activating the mechanisms present in the robust control agents group, which guarantee an acceptable level of performance for the whole system, in case of disturbances.

Algorithm 1 Master Agent

```
Start(subordinate agents);
while true do
    Check agents;
    ev ← Event();   /* wait for new event
    */
    if Check message(ev) then
    /* CyberSecurity */
        Run command(ev);   /* see Table 2
        */
    else
        Discard message;
        Report to MAS manager;
    end
end
```

As a remark, it should be stressed that the proposed resilient architecture enhances the resilience of CPSs, by addressing some of their critical vulnerabilities, as described in section II-A. The framework does not assume the infrastructure isolation of external network environments, and takes into account cyber security policies, implemented through agents included in the cyber security group, and tailored to each particular device. To deal with heterogeneity of components, local agents which are dependent on the type and inherent characteristics of the underlying devices are accordingly deployed. Moreover, the inherent security vulnerabilities stemming from a diversity of communication protocols are mitigated by implementing communications through the distributed middleware. In this case, communication between devices are ensured by the corresponding master agents, which guarantee compliance of messages. The protection of data is dealt in the application layer, by agents belonging to the cyber security group, which ensure privacy and integrity of data, and provide mechanisms for authorization and access control. The physical security of sensors and actuators is addressed by agents included in the robust control group. They implement FDI mechanisms to assess the underlying physical status. Finally, the status of communications and devices availability are taken into account in the network reliability group. Information shared among these agents allows to maintain an updated database with the devices that are currently in normal operation and those being compromised.

IV. CASE STUDY

A. TEST-BED DESCRIPTION

The test-bed (Fig. 3) consists of a three-tank system, eight Crossbow TelosB wireless nodes, one single board computer, an IPv6 network and three desktop computers. The AMIRA DTS 200 three-tank system comprises three plexiglas cylindrical tanks supplied with distilled water. The liquid levels, h_1, h_2 and h_3, are measured through piezoresistive transducers sensors. The middle tank T_3 is connected to the other two tanks by means of circular cross-section pipes each including a manual ball valve. The main outlet of the system is located at tank T_2, which is directly connected to the collecting liquid reservoir, by means of a circular cross-section pipe incorporating an outflow ball valve, while the bottom outflow valves are assumed normally closed. This system is also provided with two pumps, $Pump_1$ and $Pump_2$, for feeding tanks T_1 and T_2 with distilled water (Fig. 4).

The WSAN infrastructure is built with Crossbow TelosB nodes, which leverage several industry standards to interoperate with other physical devices, such as USB, IEEE 802.15.4 and Zigbee. These nodes consist of low power wireless devices including native support for some of the most used open source operating systems by WSN community, namely TinyOS and Contiki. Supported network stacks include 6LowPAN and IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) routing protocol. Each node has Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) ports, to which sensors and actuators can be attached. The operating system used in the WSAN programming is the Contiki.

The WSAN includes eight Crossbow TelosB nodes, of which three nodes are configured as sensors (S_1, S_2, S_3), and used to collect the tanks' levels, namely h_1, h_2 and h_3, while the two additional nodes are used as actuators (A_1, A_2)

TABLE 2. Master agent events.

Application layer master agent		
External event	Sensor reading	Receives the reading from the sensor or model, analyses it with FDI algorithms, and sends the accommodated value to the controller.
Robust control group event	Control action	Inquires the network reliability group for the available destination nodes to sending a control action and sends it out.
	Model reading	Checks the sensor unavailability and sends the estimated model reading to the available controller.
Cyber security group event	Security message	Reports the user about a security issue and block the communications with this device until the problem is completely addressed.
Network reliability group event	Sensor unavailable	Activates remote model.
Mas manager group event	Command	Performs the command
Network layer master agent		
External event	Command	Performs the command
	Sensor reading	Receives the reading from the sensor node. Inquires the network reliability for the available destination nodes to send the reading and send it out.
	Control action	Receives the control action from the remote controller. Inquires the network reliability group for the available destination nodes to send the control action and sends it out.
Robust control group event	Local model reading	Checks the sensor unavailability and sends the estimated model reading to the available controller.
	Local control action	Checks the remote controller unavailability and sends the local control action to the node selected by the network reliability group.
Cyber security group event	Security message	Reports the user about a security issue and block the communications with this device until the problem is completely addressed.
Network reliability group event	Sensor unavailable	Activates local model.
	Controller unavailable	Activates local controller.
Physical layer master agent		
External event	Control action	Receives the control action, analyses it with FDI algorithms, and sends the value to the actuator node.
	Command	Performs the command.
Robust control group event	Sensor reading	Receives the reading from the sensor, analyses it with FDI algorithms, and sends the accommodated value to the node selected by the network reliability group.
Cyber security group event	Security message	Reports the user about a security issue and block the communications with this device until the problem is completely addressed.
Network reliability group event	Communications off	Activates the emergency mode.

associated with the two pumps. Two extra nodes (IT_1, IT_2) are used as intermediate nodes, on which the resilience enhancement methods are deployed. Finally, a sink node (*Sink*) is additionally included in the network, aiming to deploy a border router, which allows the routing of WSAN and the interaction with external networks through the gateway.

The gateway is implemented on a single board computer Raspberry Pi 2 Model B. This device is a compact embedded computer module based on a 900 MHz quad-core ARM Cortex-A7 CPU with 1 GB RAM. It includes 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port and a micro SD card slot. The Raspberry Pi can run the full range of ARM GNU/linux distributions. In this work, the Raspbian has been chosen as the operating system. The sink node is attached via a USB port to this single board computer, where the Tunslip is running and the router implemented. All of these devices allow IPv6 communication directly between the WSAN nodes and the remaining system, creating a Serial Line Protocol (SLIP) tunnel between the physical serial port and the virtual network interface [46].

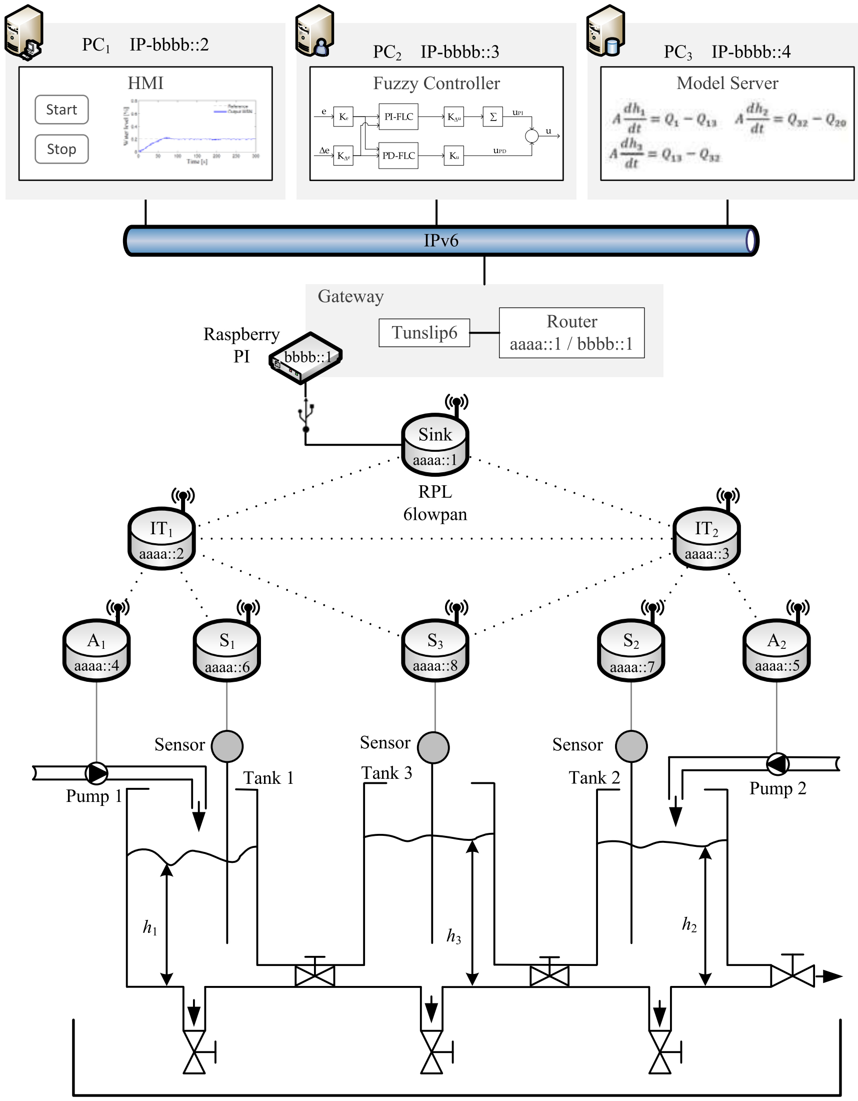

FIGURE 3. Laboratory test-bed.

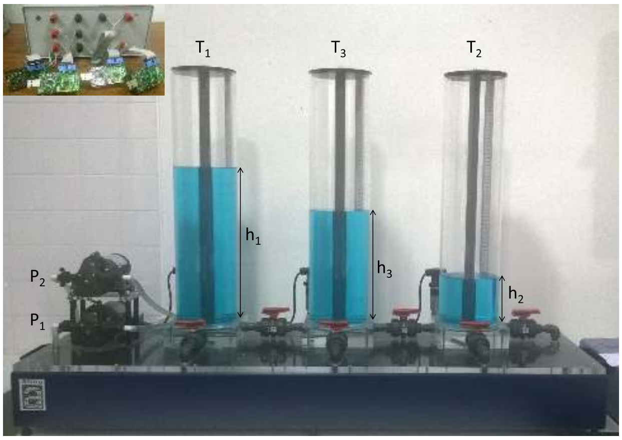

FIGURE 4. Three-tank system.

Three remote network devices located on PC$_1$, PC$_2$ and PC$_3$ allow the external interaction with the physical system over the WSAN. The HMI, is implemented on PC$_1$, through which the entire system can be configured and current states monitored, and allowing in addition interaction with a user. With respect to the remote controllers, they are implemented on PC$_2$. These controllers are of Mamdani-type Fuzzy PID controller (see [47]), being the control actions and readings delivered through the WSAN. In PC$_3$ a non-linear mathematical model of the three-tank system described by (1) is implemented [48]. This model is used as a soft-sensor, which allows to estimate sensors readings, in case of a compromising event taking place on sensor nodes or on the network.

$$\begin{aligned}A\frac{\mathrm{d}h_1}{\mathrm{d}t} &= \beta_1 \times u_1 - a_{13} \times Sn \times sgn(h_1 - h_3)\\ &\quad \times \sqrt{2g|h_1 - h_3|}\\ A\frac{\mathrm{d}h_3}{\mathrm{d}t} &= a_{13} \times Sn \times sgn(h_1 - h_3) \times \sqrt{2g|h_1 - h_3|}\\ &\quad - a_{32} \times Sn \times sgn(h_3 - h_2) \times \sqrt{2g|h_3 - h_2|}\end{aligned}$$

$$A\frac{\mathrm{d}h_1}{\mathrm{d}t} = \beta_2 \times u_2 + a_{32} \times Sn \times sgn(h_3 - h_2) \times\sqrt{2g|h_3 - h_2|} - a_{20} \times Sn \times \sqrt{2g \times h_2} \qquad (1)$$

Table 3 presents the corresponding model parameters. These values were obtained through measurements and optimization experiments performed on the AMIRA˙ DTS 200 three-tank system.

TABLE 3. AMIRA˙ DTS 200 parameters.

Parameter	Description	Nominal Value
β_1	Constant	$3.85^{-5}\ m^3/V.s$
β_2	Constant	$3.70^{-5}\ m^3/V.s$
a_{13}	Outflow coefficient	0.4473
a_{32}	Outflow coefficient	0.4429
a_{20}	Outflow coefficient	0.7357
Sn	Section of connection pipe	$5^{-5}\ m^2$
A	Section of tank	$1.46^{-2}\ m^2$
g	Gravity	$9.8\ m/s^2$

B. MULTI-AGENT FRAMEWORK

The multi-agent framework considered in this case study is presented in Table 4. According to Fig. 1, agents are distributed over the application, communication networks and sensors/actuators layers. Each agent belongs to one of the groups defined in Fig. 2, and is responsible for a specific task, and are coordinated by the corresponding master agent.

The following agents were implemented in this case study:

1) MASTER AGENT

The master agent main goal is to carry out extensive management routines related to subordinate local agents and to coordinate required communications. These agents were implemented according to the guidelines described in Section III-E.

2) SENSOR AGENT

The sensor agent belongs to the robust control group. This agent is responsible for collecting data from the plant and accommodating possible outliers in raw readings. The local detection and accommodation of outliers is based on the approach suggested in [49].

3) ACTUATOR AGENT

The actuator agent belongs to the robust control group, and is responsible for sending a particular control action received at the actuator node to the corresponding DAC port.

4) CONTROL AGENT

The control agent belongs to the robust control group. In the control application, this agent receives sensor readings and implements a control algorithm in order to return a control action to be sent out through the WSAN. When deployed on intermediate nodes IT_1 and IT_2, it implements a State Variable Feedback (SVF) controller, by relying on the most recent reference signal received from the server. Computed control actions are subsequently sent to the corresponding actuator node, whenever the remote controller control action is not available. This SVF controller is designed taking into account a linearised dynamic model of the plant.

5) MODEL AGENT

The model agent is included in the robust control group. Its main goal is to predict the physical system behavior, as well as other important components of the system. Additionally, this agent receives sensor readings and control actions in order to update predictions of the plant behavior. It is crucial to ensure a safe operation mode whenever the sensors' readings are not available. When deployed on the model application, it incorporates a non-linear mathematical model of the plant described by (1), whereas in the intermediate nodes IT_1 and IT_2 it assumes a linear discrete-time state-space model derived from (1).

6) SAFETY AGENT

The safety agent belongs to the robust control group. It is responsible for ensuring a provisional emergency mode in case of a communication link breakdown with actuator nodes. In emergency mode, the actuator node feeds the system with a control action pre-defined by the designer, which guarantees its stabilization. This operation mode is only held for a short time interval, in order to reset/resume communications. After this time threshold is exceeded the system is shut down for safety reasons.

7) REPORT AGENT

The report agent belongs to the MAS manager group. It allows a safe interaction between users, agents and the system, by processing users' requests. This agent also provides users with relevant information regarding the system operation and alarms.

8) SECURITY AGENT

The security agent belongs to the cyber security group. It is responsible for analyzing important variables of the system for coherence, as well as the structure of messages. It is internally defined with admissible maximum and minimum values for these variables. Whenever there are outside admissible bounds an alarm message is triggered and sent to the HMI. In addition, the underlying variable is adjusted to be within the predefined threshold. The structure of the messages sent and received by nodes and between agents is very stringent. If the security agent detects any difference, a message error is triggered and sent to the HMI, while the current message is ignored.

9) CONTEXT AGENT

The context agent belongs to the network reliability group. This agent is responsible for keeping the awareness of context regarding communications. All components have one context

TABLE 4. Multi-agent framework.

<table>
<tr><th colspan="2">HMI application (PC_1)</th><th colspan="2">Controller application (PC_2)</th><th colspan="2">Model application (PC_3)</th><th rowspan="2">Application Layer</th></tr>
<tr><td colspan="2">• Master agent
• Report agent
• Contex agent
• Security agent</td><td colspan="2">• Master agent
• Control agent
• Context agent
• Security agent</td><td colspan="2">• Master agent
• Model agent
• Context agent
• Security agent</td></tr>
<tr><th colspan="6">Intermediate node (IT_1, IT_2)</th><td rowspan="2">Network Layer</td></tr>
<tr><td colspan="6">• Master agent
• Model agent
• Control agent
• Contex agent
• Security agent</td></tr>
<tr><th colspan="3">Sensor node (S_1, S_2, S_3)</th><th colspan="3">Actuator node (A_1, A_2)</th><td rowspan="2">Physical Layer</td></tr>
<tr><td colspan="3">• Master agent
• Sensor agent
• Context agent
• Security agent</td><td colspan="3">• Master agent
• Actuator agent
• Context agent
• Safety agent
• Security agent</td></tr>
</table>

agent, which periodically checks the communications among components. In the case of communication failure with any of the nodes, these type of agents alert the corresponding master agent where to send out the data, by temporarily reconfiguring the network. A communication failure between two nodes is detected when one node is unable to communicate or receive a reply from another node, over three or more sampling times.

C. EXPERIMENTS

In the following experiments, the control goal is to keep the outputs from the three-tank system, namely the level of tanks T_1 and T_2 (Fig. 3) at prescribed values. This is carried out by feeding the two pumps, $Pump_1$ and $Pump_2$, with appropriate control actions provided by the underlying controllers. Readings are collected from wireless sensors' ADCs at a frequency of 1 Hz, while actuating commands are sent to pumps at the same frequency. Furthermore, in order to characterize the way the system is operating, in terms of faulty events regarding the whole system, Table 5 presents the possible operating states for both the sensor and the actuator.

Regarding the figures showing results from experiments, plots on the top show the true sensor reading in blue (OS), the sensor value relayed through the WSAN and used by controllers is shown in red (OW), while the reference signal is presented in black (R). The middle plots correspond to control actions, in which the blue color refers to the applied control signal on wireless actuator nodes (IA), and the red color represents the computed control action (RCS). Finally, the bottom plots show the system status, where the blue color corresponds to the sensor status and the red color to the actuator status.

TABLE 5. Operating states.

Sensor status	
Lost	A sensor reading was lost.
Normal	Normal operation. The sensor node sends a reading to an intermediate node, which relays it to the remote controller.
Sensor node	A sensor node sends directly the reading to the remote controller.
Remote model	A reading is estimated/predicted by the remote model.
Local model	A reading is estimated/predicted using a local model deployed on an intermediate node.
Actuator status	
Lost	A control actuation value was lost.
Normal	Normal operation. The remote controller sends an actuation value to an intermediate node which relays it to an actuator node.
Remote controller	The remote controller sends the actuation value directly to an actuator node.
Local controller	A control action is computed by a local controller deployed on an intermediate node.
Safety controller	Actuator node is in safe operation mode.

1) NORMAL OPERATION

The first experiment concerns the case where the system is in normal operation, with no faults or disturbances acting on the system. Fig. 5 and Fig. 6 show the corresponding behavior of the three-tank system. In this scenario, the deployed MAS collects readings from sensor nodes and feeds the actuator with a control action provided by the remote controller, through the corresponding DAC ports. As can be observed, the levels of tanks T_1 and T_2 are driven to the underlying

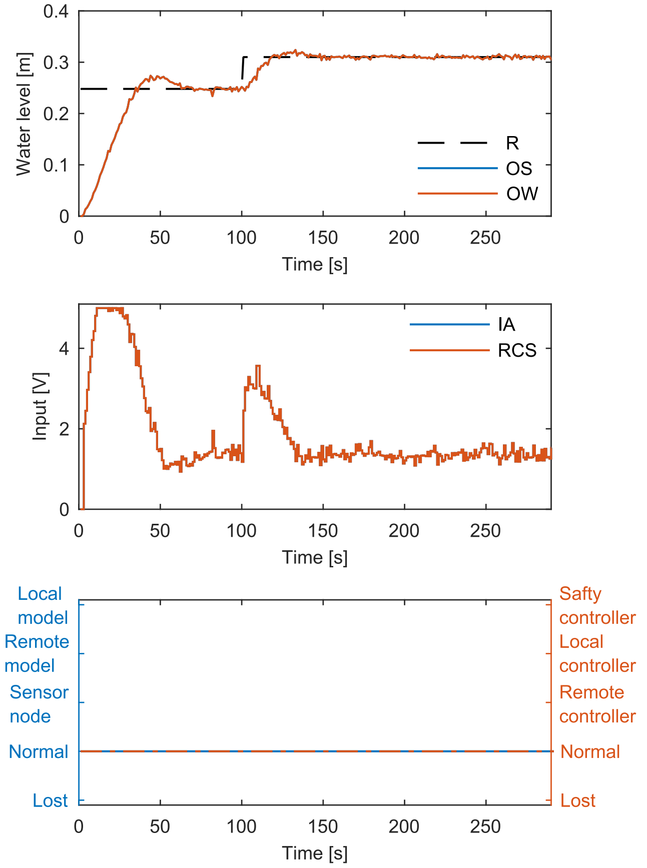

FIGURE 5. Normal operation: Tank T_1. R-Reference, OS-Output sensor, OW-Output WSAN, IA-Input actuator, RCS-Remote control signal.

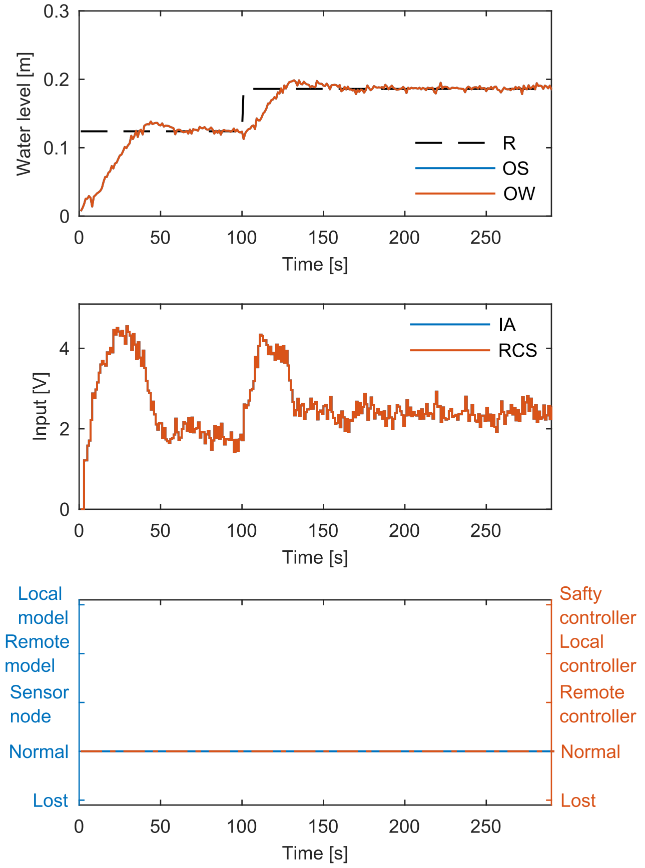

FIGURE 6. Normal operation: Tank T_2. R-Reference, OS-Output sensor, OW-Output WSAN, IA-Input actuator, RCS-Remote control signal.

reference levels, by means of a coordinate smooth behavior of $Pump_1$ and $Pump_2$. Moreover, there are no faulty events regarding the wireless sensors and actuators associated with the T_1 and T_2, as can be confirmed from the bottom plots in Fig. 5 and Fig. 6.

2) JAMMING ATTACK

In the second experiment, a jamming attack was implemented on the test-bed, in particular, on the sink node. This event prevents the sink node from forwarding any data to nodes and remote applications, thus compromising the gateway and the link with remote devices. Fig. 7 and Fig. 8 show the results from an attack taking place between time 102 and 202 second. As can be observed, the resilience enhancement MAS-based architecture is able to accommodate the underlying malicious attack, allowing to keep T_1 and T_2 levels around the most recent corresponding references, received from the remote controller. This is achieved by incorporating a safeguard mechanism, in which the SVF controller deployed on intermediate nodes IT_1 and IT_2 (control agent in Table 4) computes the discrete-time actuating signal and sends it to the corresponding actuator node. When the communications are finally resumed, which happens after the jamming attack is blocked, the normal operation of the entire system is re-established. In this scenario, given the unavailability of communication with the remote controller, there is a reconfiguration of the whole control system in order accommodate the compromising event, and thus contributing to a mitigated impact on the overall closed loop performance. Table 6 shows some performance metrics associated with this faulty event in Fault Sink description.

3) NODE LOST

The last experiment concerns the case of a node lost, where communication with sensor node S_1 is lost due to node malfunction. This can be due to, for instance, a power failure, congestion on the radio receptor or resulting from a cyber attack. Fig. 9 shows just the results concerning tank T_1. In this scenario, when the node S_1 is under a fault event between time 102 and 202 second, the model agent deployed on the model application (Table 4) sends a prediction for T_1 output to the underlying controller. As can be observed from Fig. 9, the implemented MAS can effectively accommodate the underlying wireless node malfunction, in such a way that the impact of the faulty event on the closed loop performance is effectively mitigated until the proper wireless sensor operation is restored. Table 6 shows some performance metrics associated with this faulty event in Fault S_1 description.

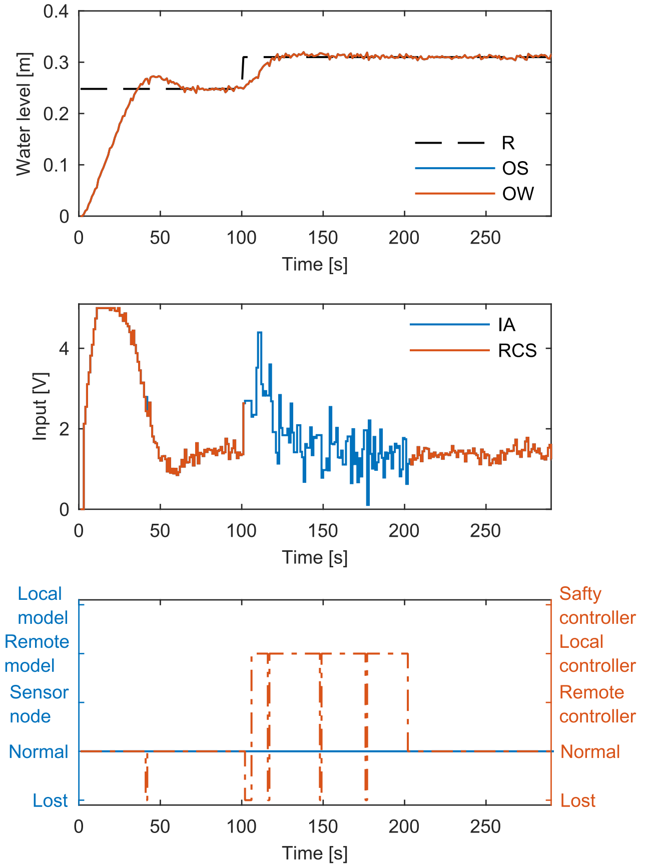

FIGURE 7. Jamming attack: Tank T_1. R-Reference, OS-Output sensor, OW-Output WSAN, IA-Input actuator, RCS-Remote control signal.

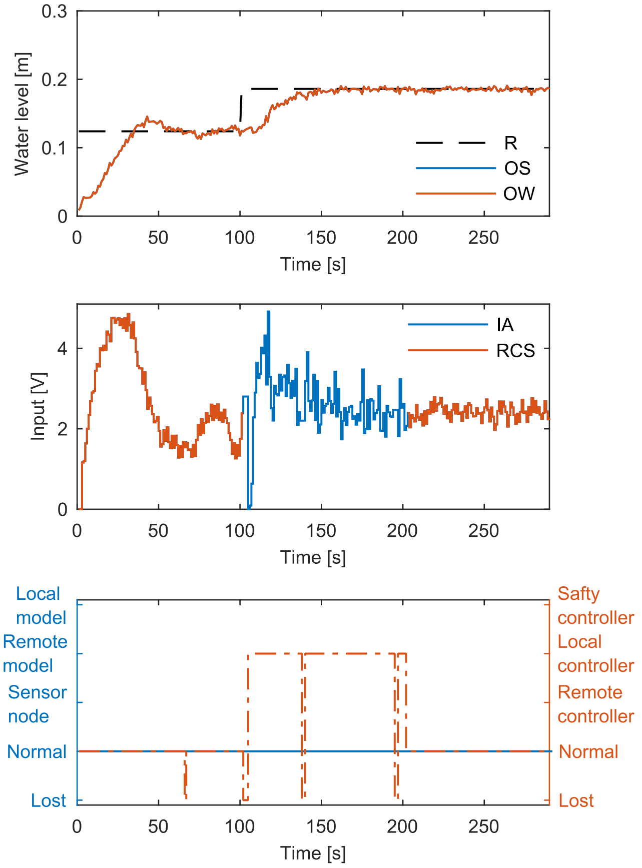

FIGURE 8. Jamming attack: Tank T_2. R-Reference, OS-Output sensor, OW-Output WSAN, IA-Input actuator, RCS-Remote control signal.

TABLE 6. Performance metrics.

Description	RMSE	MAE	QC	Reliability
Normal Operation	3.54×10^{-2}	1.90×10^{-2}	100%	100%
Fault IT_1	3.59×10^{-2}	1.90×10^{-2}	98.12%	97.77%
Fault IT_2	3.52×10^{-2}	1.89×10^{-2}	113.37%	98.27%
Fault S_1	3.31×10^{-2}	1.70×10^{-2}	113.73%	97.52%
Fault S_2	3.71×10^{-2}	2.73×10^{-2}	87.23%	96.53%
Fault S_1 and S_2	3.71×10^{-2}	2.47×10^{-2}	88.19%	96.29%
Fault Sink	4.03×10^{-2}	2.31×10^{-2}	74.14%	96.53%
Fault Sink and S_1	3.95×10^{-2}	2.41×10^{-2}	77.42%	96.78%
Fault Sink and S_2	3.99×10^{-2}	2.63×10^{-2}	75.96%	97.28%
Fault Sink without MAS	7.53×10^{-2}	6.29×10^{-2}	6.61%	0%

4) RESULTS DISCUSSION

To assess the effectiveness of the proposed architecture aiming to improve the resilience of the closed loop system, four performance metrics are used for the considered fault events, with the corresponding values presented in Table 6. Two of these metrics are based on the control error, namely the Root Mean Square Error (RMSE) (2) and Mean of Absolute Error (MAE) (3). They both allow to evaluate the benefits provided by the proposed framework in order to accommodate faults. The third metric allows the comparison between the normal performance of the system and its operation in the case of faulty events. This metric is regarded as a Quality Control (QC) measure and is computed according to (4). The fourth metric aims to address the reliability of the communication infrastructure, which is dependent on the quality of communication, and is impacted by cyber vulnerabilities of the system. The reliability is calculated according to (5).

$$\begin{aligned} RMSE =& \sqrt{\frac{1}{101}\sum_{k=102}^{202}[ref_1(k)-h_1(k)]^2} \\ &+ \sqrt{\frac{1}{101}\sum_{k=102}^{202}[ref_2(k)-h_2(k)]^2} \qquad (2) \end{aligned}$$

$$\begin{aligned} MAE =& \frac{1}{101}\sum_{k=102}^{202}[|ref_1(k)-h_1(k)| \\ &+|ref_2(k)-h_2(k)|] \qquad (3) \end{aligned}$$

$$\begin{aligned} QC(\%) =& \left[\frac{\sum[ref_1(k)-h_{1n}(k)]^2}{\sum\left[ref_1(k)-h_{1f}(k)\right]^2}\right. \\ &\left.+\frac{\sum[ref_2(k)-h_{2n}(k)]^2}{\sum\left[ref_2(k)-h_{2f}(k)\right]^2}\right] \times 100 \qquad (4) \end{aligned}$$

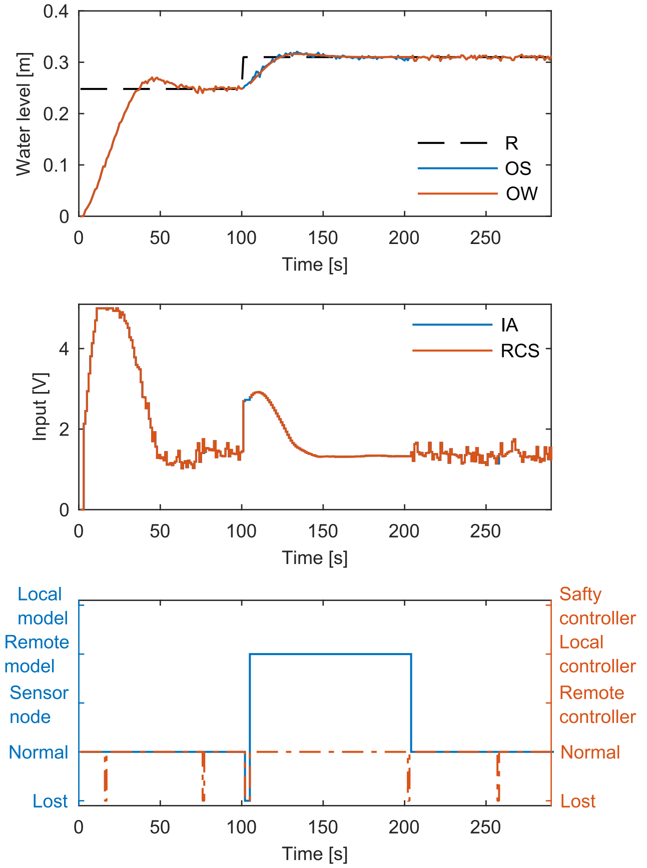

FIGURE 9. Node S_1 lost: Tank T_1. R-Reference, OS-Output sensor, OW-Output WSAN, IA-Input actuator, RCS-Remote control signal.

with h_{1n} and h_{2n} the tanks' levels in normal operation, and h_{1f} and h_{2f} the tanks' levels in fault operation.

$$Reliability(\%) = \frac{\sum pkt_{act\leftarrow} + \sum pkt_{ctrl\leftarrow}}{\sum pkt_{\rightarrow act} + \sum pkt_{\rightarrow ctrl}} \times 100 \quad (5)$$

with $pkt_{act\leftarrow}$ received packets by actuators, $pkt_{ctrl\leftarrow sens}$ received packets by controllers, $pkt_{\rightarrow act}$ the packets sent to actuators and $pkt_{\rightarrow ctrl}$ the packets sent to controllers.

Taking into account RMSE and MAE, the errors in normal operation are similar to those in the case of intermediate node IT_1 and IT_2 faults. When these nodes are compromised, the MAS promptly detects this occurrence and promotes a direct connection between the remote controller and the corresponding sensor and actuator. In the case of sensor faults, the lack of noise in predicted outputs leads to a better behavior of the remote controller, for a fault on S_1. As for a fault on S_2, there is a larger error, which is due to model-plant mismatch, stemming from non-linearities not included in the model and related to the outflow from the system at tank T_2. When the communication with remote applications is lost, the values of these metrics increase. Despite a quick reaction from the MAS in launching local controllers, they have slower dynamics than that of the remote controller. The same conclusions can be drawn from the QC metric. In this case, it can be observed that in two situations (fault on IT_1 and S_1) the value of this metric is larger than 100%, meaning a better system performance during these faults than in the normal operation case.

The reliability metric shows clearly the effectiveness of the proposed framework in redirecting the communications and applying the necessary countermeasures to maintain the correct operation of the whole system, in case of fault events compromising communications. Packet loss is mostly due to user-defined time threshold within the context agents. It should be pointed out, however, that during a fault event, the number of packets transmitted over the WSAN will naturally increase, which leads to a higher likelihood of a packet being lost.

Taking into account the reported experiments, it is clear that the proposed architecture is invaluable in dealing, at least, with the tested faults. This behavior highlights the relevance of the framework in enhancing the overall resilience of the CPS. Moreover, this architecture maintains an awareness of the system status, along with the underlying context in face of adverse events, allowing the CPS operation in an acceptable level of normalcy.

As a remark, it should be stressed that although the considered case study does not have a high complexity, it is possible to scale this approach up to more complex systems with a greater number of devices. Given the distributed nature of the architecture, the implemented agents are adapted to the corresponding devices, taking into account their limitations. However, in some networks the amount of messages transmitted between agents may compromise the effectiveness of the approach. In such scenarios, the underlying CPS can be split into smaller sub-systems, which will be interconnected with a more effective network.

V. CONCLUSION

A distributed middleware based on a hierarchical multi-agent framework is proposed in this work to enhance the resilience of CPSs over heterogeneous networks. The CPS is analyzed taking into account its layers, namely the physical, network and application layers, where the main vulnerabilities and the potential physical and cyber attacks are discussed. These vulnerabilities were addressed based on five groups of agents, namely master agents, MAS manager, robust control agents, cyber security agents and network reliability agents, which provide the necessary flexibility and countermeasures. They allow deploying specific functions, to address cyber security and physical security issues. The developed hierarchical methodology embeds and prioritizes incoming information to ensure state and context awareness, which is used in accommodating resilience-compromising events. A case study consisted of a CPS test-bed was considered in order to assess the feasibility and performance enhancement capabilities of the proposed framework. Based on three experiments, including a normal operation, a jamming attack on the sink node and a sensor node loss, the effectiveness and relevance of the proposed approach in dealing with malicious attacks and faults on CPS devices has been demonstrated.

REFERENCES

[1] L. Labaka, J. Hernantes, and J. M. Sarriegi, "Resilience framework for critical infrastructures: An empirical study in a nuclear plant," *Rel. Eng. Syst. Saf.*, vol. 141, pp. 92–105, Sep. 2015.

[2] T. Xu and A. J. Masys, *Critical Infrastructure Vulnerabilities: Embracing a Network Mindset*. Cham, Switzerland: Springer, 2016, pp. 177–193.

[3] Z. A. Vale, H. Morais, M. Silva, and C. Ramos, "Towards a future SCADA," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2009, pp. 1–7.

[4] A. Romanovsky and F. Ishikawa, *Trustworthy Cyber-Physical Systems Engineering*. Boca Raton, FL, USA: CRC Press, 2016.

[5] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Design Automat. Conf. (DAC)*. New York, NY, USA: ACM, Jun. 2010, pp. 731–736.

[6] S. Ali, S. Qaisar, H. Saeed, M. Khan, M. Naeem, and A. Anpalagan, "Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring," *Sensors*, vol. 15, no. 12, pp. 7172–7205, Mar. 2015.

[7] X. Jin, W. M. Haddad, and T. Yucelen, "An adaptive control architecture for mitigating sensor and actuator attacks in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 6058–6064, Nov. 2017.

[8] T. Cruz *et al.*, "Improving cyber-security awareness on industrial control systems: The CockpitCI approach," *J. Inf. Warfare*, vol. 13, no. 4, pp. 27–41, 2015.

[9] C. Rieger, Q. Zhu, and T. Basar, "Agent-based cyber control strategy design for resilient control systems: Concepts, architecture and methodologies," in *Proc. 5th Int. Symp. Resilient Control Syst.*, Aug. 2012, pp. 40–47.

[10] R. Mitchell and I. R. Chen, "Modeling and analysis of attacks and counter defense mechanisms for cyber physical systems," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 350–358, Mar. 2016.

[11] Y. Yuan, Q. Zhu, F. Sun, Q. Wang, and T. Baçar, "Resilient control of cyber-physical systems against Denial-of-Service attacks," in *Proc. 6th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2013, pp. 54–59.

[12] R. E. Johnson, "Survey of SCADA security challenges and potential attack vectors," in *Proc. Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Nov. 2010, pp. 1–5.

[13] R. W. Anwar, M. Bakhtiari, A. Zainal, A. H. Abdullah, and K. N. Qureshi, "Security issues and attacks in wireless sensor network," *World Appl. Sci. J.*, vol. 30, no. 10, pp. 1224–1227, 2014.

[14] E. Hollnagel and C. P. Nemeth, "Resilience engineering perspectives, preparation and restoration," in *Ashgate Studies in Resilience Engineering*, vol. 2. Boca Raton, FL, USA: CRC Press, 2016.

[15] D. D. Woods and E. Hollnagel, *Resilience Engineering: Concepts and Precepts*. Boca Raton, FL, USA: CRC Press, 2017.

[16] R. Arghandeh, A. von Meier, L. Mehrmanesh, and L. Mili, "On the definition of cyber-physical resilience in power systems," *Renew. Sustain. Energy Rev.*, vol. 58, pp. 1060–1069, May 2016.

[17] D. D. Woods, "Four concepts for resilience and the implications for the future of resilience engineering," *Rel. Eng. Syst. Saf.*, vol. 141, pp. 5–9, Sep. 2015.

[18] C. G. Rieger, D. I. Gertman, and M. A. McQueen, "Resilient control systems: Next generation design research," in *Proc. 2nd Conf. Hum. Syst. Interact.*, May 2009, pp. 632–636.

[19] Z. Li, C. Zang, P. Zeng, H. Yu, and H. Li, "MAS based distributed automatic generation control for cyber-physical microgrid system," *IEEE/CAA J. Automatica Sinica*, vol. 3, no. 1, pp. 78–89, Jan. 2016.

[20] S. M. M. Rahman, "Cyber-physical-social system between a humanoid robot and a virtual human through a shared platform for adaptive agent ecology," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 190–203, Jan. 2018.

[21] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.

[22] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 427–438, Feb. 2013.

[23] V. P. Singh, N. Kishor, and P. Samuel, "Distributed multi-agent system-based load frequency control for multi-area power system in smart grid," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5151–5160, Jun. 2017.

[24] A. M. Farid, "Multi-agent system design principles for resilient coordination & control of future power systems," *Intell. Ind. Syst.*, vol. 1, no. 3, pp. 255–269, 2015.

[25] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber–physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.

[26] A. W. Righi, T. A. Saurin, and P. Wachs, "A systematic literature review of resilience engineering: Research areas and a research agenda proposal," *Rel. Eng. Syst. Saf.*, vol. 141, pp. 142–152, Sep. 2015.

[27] D. Wijayasekara, O. Linda, M. Manic, and C. Rieger, "FN-DFE: Fuzzy-neural data fusion engine for enhanced resilient state-awareness of hybrid energy systems," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2065–2075, Nov. 2014.

[28] H. E. Garcia, W.-C. Lin, S. M. Meerkov, and M. T. Ravichandran, "Resilient monitoring systems: Architecture, design, and application to boiler/turbine plant," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2010–2023, Nov. 2014.

[29] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Distributed fault detection and isolation resilient to network model uncertainties," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2024–2037, Nov. 2014.

[30] W. Zeng and M.-Y. Chow, "Resilient distributed control in the presence of misbehaving agents in networked control systems," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2038–2049, Nov. 2014.

[31] G. De La Torre, T. Yucelen, and J. D. Peterson, "Resilient networked multiagent systems: A distributed adaptive control approachy," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 5367–5372.

[32] C. Rieger and Q. Zhu, "A hierarchical multi-agent dynamical system architecture for resilient control systems," in *Proc. 6th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2013, pp. 6–12.

[33] K. Ji and D. Wei, "Resilient control for wireless networked control systems," *Int. J. Control, Automat. Syst.*, vol. 9, no. 2, pp. 285–293, Apr. 2011.

[34] P. Li, L. Li, G. Song, and Y. Yu, "Wireless sensing and vibration control with increased redundancy and robustness design," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2076–2087, Nov. 2014.

[35] X. Wang *et al.*, "Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8539991

[36] X. Wang *et al.*, "Optimizing content dissemination for real-time traffic management in large-scale Internet of vehicle systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1093–1105, Feb. 2018.

[37] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for SCADA systems," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1836–1846, Nov. 2008.

[38] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.

[39] Y. Gao *et al.*, "Analysis of security threats and vulnerability for cyber-physical systems," in *Proc. 2013 3rd Int. Conf. Comput. Sci. Netw. Technol.*, Oct. 2013, pp. 50–55.

[40] M. N. Al-Mhiqani *et al.*, "Cyber-security incidents: A review cases in cyber-physical systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 1, pp. 499–508, 2018.

[41] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Comput. Ind.*, vol. 100, pp. 212–223, Sep. 2018.

[42] B. Advantage and K. Lab, "The state of industrial cybersecurity 2017," Bus. Advantage, London, U.K., Tech. Rep., 2017. [Online]. Available: https://go.kaspersky.com/rs/802-IJN-240/images/ICSWHITEPAPER.pdf

[43] M. Healy, T. Newe, and E. Lewis, "Security for wireless sensor networks: A review," in *Proc. IEEE Sensors Appl. Symp.*, Feb. 2009, pp. 80–85.

[44] V. Amruth, J. Kuriakose, M. T. B. Lahori, Shehanaz, K. Sushanth, and A. Kumar, "Attacks that downturn the performance of wireless networks," in *Proc. Int. Conf. Comput. Commun. Control Autom.*, Feb. 2015, pp. 122–128.

[45] F. E. P. Januário, J. Leitão, A. Cardoso, and P. Gil, "Resilience enhancement in cyber-physical systems: A multiagent-based framework," in *Multi-Agent Systems*. Rijeka, Croatia: InTech, Sep. 2017.

[46] F. Januário, A. Santos, L. Palma, A. Cardoso, and P. Gil, "A distributed multi-agent approach for resilient supervision over a IPv6 WSAN infrastructure," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2015, pp. 1802–1807.

[47] F. Januário, S. Amâncio, L. Catarina, P. Luis, A. Cardoso, and P. Gil, "Outliers accommodation in fuzzy control systems over WSAN," in *Intelligent Decision Technologies*, vol. 255. Amsterdam, The Netherland: IOS Press, 2013, pp. 334–343.

[48] Amira, "DTS200 laboratory setup three—tank—system," Amira, Duisburg, Germany, Tech. Rep., 2002.

[49] A. Cardoso, A. Santos, G. B. Nunes, and P. Gil, "A multi-agent approach for outlier accommodation in wireless sensor and actuator networks," in *Proc. 10th Portuguese Conf. Autom. Control*, 2012, pp. 1–6.

FÁBIO JANUÁRIO was born in Almada, Portugal, in 1988. He received the M.Sc. degree in electrical engineering from the Faculty of Sciences and Technology, NOVA University of Lisbon, Portugal, in 2012, where he is currently pursuing the Ph.D. degree.

He is currently with the Energy Research Group, Centre of Technology and Systems, UNINOVA. His research interests include wireless sensors and actuator networks, resilient systems, multi-agent systems, and distributed communication networks.

ALBERTO CARDOSO was born in Coimbra, Portugal. He received the Licentiate degree in electrical engineering, the M.Sc. degree in systems and information technologies, with specialty in systems and control, and the Ph.D. degree in informatics engineering from the University of Coimbra.

He has been an Assistant Professor with the Faculty of Sciences and Technology, University of Coimbra, since 2006. His research interests include intelligent systems, fault-tolerant control, networked control systems, wireless sensor and actuator networks, multi-agent systems, data analysis, online experimentation, and remote and virtual laboratories.

PAULO GIL was born in Lisbon, Portugal. He received the Licentiate degree in mechanical engineering and the M.Sc. degree in informatics engineering, with specialty in system and information technologies, from the University of Coimbra, Coimbra, Portugal, and the Ph.D. degree in electrical engineering, with specialty in automatic control, from the Universidade NOVA de Lisboa, Campus de Caparica, Lisbon, Portugal.

He has been an Assistant Professor with the Department of Electrical Engineering, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, since 2004. His current research interests include distributed control systems, networked control systems, wireless sensor and actuator networks, intelligent systems, and multi-agent systems.