Chapter 10

Resilience Enhancement in Cyber-Physical Systems: A Multiagent-Based Framework

Fábio Emanuel Pais Januário, Joaquim Leitão, Alberto Cardoso and Paulo Gil

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.69356

Abstract

The growing developments on networked devices, with different communication platforms and capabilities, made the cyber-physical systems an integrating part of most critical industrial infrastructures. Given their increasing integration with corporate networks, in which the industry 4.0 is the most recent driving force, new uncertainties, not only from the tangible physical world, but also from a cyber space perspective, are brought into play. In order to improve the overall resilience of a cyber-physical system, this work proposes a framework based on a distributed middleware that integrates a multiagent topology, where each agent is responsible for coordinating and executing specific tasks. In this framework, both physical and cyber vulnerabilities alike are considered, and the achievement of a correct state awareness and minimum levels of acceptable operation, in response to physical or malicious disturbances, are guaranteed. Experimental results collected with an IPv6-based simulator comprising several distributed computational devices and heterogeneous communication networks show the relevance and inherent benefits of this approach.

Keywords: cyber-physical systems, artificial cognition, context awareness, distributed middleware, heterogeneous systems

1. Introduction

Modern societies are quite dependent on efficient, stable and secure operation of critical infrastructures. As a whole, they consist of a wide range of heterogeneous devices, with several levels of resources, which are interconnected using different networking technologies [1]. These environments rely heavily on communication infrastructures, in order to take advantage of

the distributed/grid or parallel computing paradigms. The migration of these systems into a cyber space that bridges the cyber world of computing and communication with the physical world is commonly referred in the literature as cyber-physical system (CPS). A CPS consists in the integration under the same umbrella of computing technologies, networking and physical processes, which aims at monitoring and controlling a given physical process [2].

This integration, however, raises a number of challenges in the context of traditional monitoring and control systems, particularly, with regard to defining a comprehensive framework for dealing with additional cyber, cognitive and human complex interdependencies, which ultimately enhance the potential for fault, malfunctions, failures or even security vulnerabilities [3]. CPSs over distributed heterogeneous environments present some vulnerabilities, which include efficient processing of information and correct assessment of the system behaviour. One main issue refers to faults and failures monitoring, being required to develop methods to identify, recover and mitigate such events. A second concern is the systems' vulnerability to cyber intrusion, where malicious actors may mask the system's degradation or relay false/fake data to higher management levels, regarding the current system's status [4]. Although the design of control systems may take into account uncertainty accommodation, namely physical disturbances, by appealing to a number of techniques such as robust control, adaptive control and stochastic control, it normally does not incorporate specific measures to deal with uncertainties associated with the cyber space.

Some previous malicious attacks on CPSs have shown that traditional protection/security mechanisms are not enough satisfactory to accommodate or mitigate such intrusions. In fact, most of the current systems have not been designed to include effective measures against cyber-attacks and have remained secured mostly through their anonymity. Anonymity, however, is no longer a guarantee of effective protection, making these systems more and more vulnerable, given the increasing likelihood of attacks. This issue is illustrated by the increasing number of incidents being reported (see [5]). In this context, cyber security gave rise to a new class of control problems, which demand a more holistic and cross-layer design approach, explicitly incorporating protection mechanisms for cyber-attacks within the overall system.

This chapter focuses on developing resilience mechanisms for such complex environments, involving a huge diversity of distributed physical devices along with a high heterogeneity of communication networks. The proposed approach makes use of agents embedded on a distributed middleware framework, where each agent is tailored for executing specific and coordinated tasks, namely, for detecting and recovering from cyber and physical malfunctions. The incorporation of these entities in the context of faults and failure diagnosis, or aiming at reducing the system vulnerability, is very appealing. They make possible the implementation of methods for resilience enhancement, including outliers detection and accommodation, as well as for maintaining the system in a safe operating state, in case of compromising events, such as communication link breakdown, or to account for security issues under the form of manipulation of data/configuration parameters by a malicious actor. For this purpose, some functionalities and attributes are provided to particular agents, so as to respond to environmental changes. To keep a permanent awareness of the overall system and react accordingly in case of compromising events, the developed mechanisms will allow to gauge the awareness of the context and the system states, including dedicated cognition functionalities.

The rest of this chapter is organized as follows. Section 2 discusses some important concepts, including that of resilience, state awareness and context awareness. Section 3 describes the proposed resilient approach and the underlying multiagent framework. Section 4 presents some results based on a simulation platform, while in Section 5, the main conclusions are drawn.

2. Resilience in cyber-physical systems

This work addresses the resilience enhancement in modern supervision systems over heterogeneous communication networks, where the physical and cyber issues are interconnected. CPSs comprise the integration of computing hardware, networking and physical processes aiming at monitoring and control a physical process, by means of feedback loops. These systems must operate dependably, safely, securely, efficiently and in real-time. A CPS roughly comprises two main layers, namely, the physical layer and cyber layer. The physical layer includes an intelligent network of actuators, sensors and additional hardware devices in order to collect information and control a physical system, while the cyber layer can be regarded as the decision-making setup, comprising information and communication devices. The cyber layer, particularly in what the industrial control systems is concerned, is typically composed of a Supervisory Control and Data Acquisition (SCADA) system [6]. The present work combines these two domains, as illustrated in **Figure 1**.

These CPSs typical consist of three main components: the control network, communication infrastructure and process network. The control network hosts all the required devices for both controlling the physical layer and providing the control interface to the process network. A typical control network can be composed of a mesh of Program Logic Controllers (PLCs), Remote Terminal Units (RTUs), as well as Wireless Sensor and Actuator Networks (WSANs). The communication infrastructure is used to interconnect different components of the system, providing a unique interface between control systems and field devices. Furthermore, the process network hosts the servers along with the Human Machine Interface (HMI) platform, which consists of servers and software packages that allow the connection to field equipment.

2.1. Resilient systems

The concept of resilience emerged, originally, associated with the fields of ecology and psychology. Nowadays, it is used in many different contexts focused on two notable areas, namely, organizational and information technology. Organizational resilience has been used to describe a movement among entities such as businesses, communities and governments to improve their ability to respond or react to and quickly recover from catastrophic events, such as natural disasters or terrorist attacks. On the other hand, the information technology resilience considers the stability and quality of service in face of threats on the computing and networking infrastructure [7].

Most researchers on information sciences define resilience purely in terms of the availability of the underlying system. Some claim that beyond availability, resiliency should include the ability to cope with threats of an unexpected and malicious nature, while others explicitly

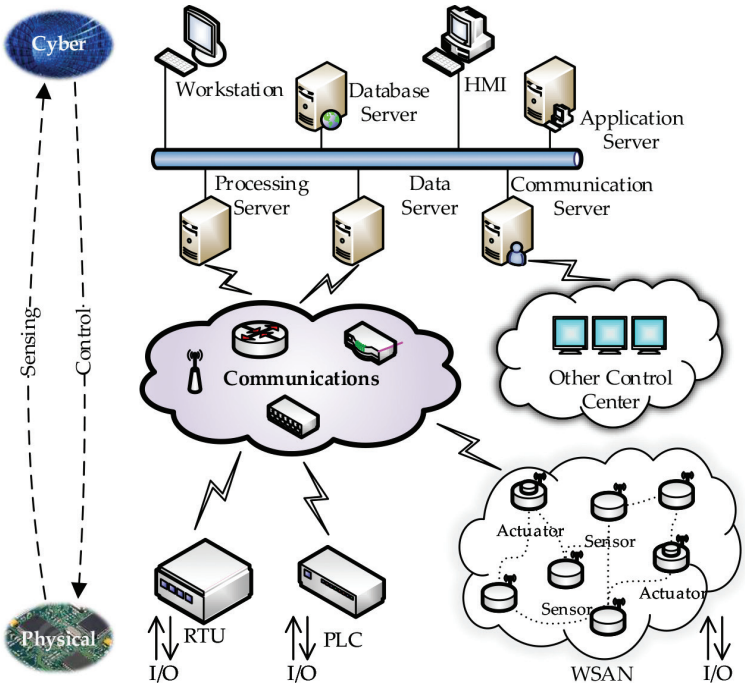

Figure 1. Example of a cyber-physical system.

include defence and recovery with respect to cyber-attacks (see [5, 8]). In the context of CPSs, resilience highlights the ability to accommodate faults or events that otherwise might compromise the stability of the system and the underlying goals.

Resilient control systems (RCSs), which are a part of CPSs, are a new control design paradigm that considers all possible threats, namely, cyber and physical aspects. In [9], it is suggested that 'Resilient control systems are those that tolerate fluctuations via their structure, design parameters, control structure and control parameters', where the presence of malicious actors is not considered. Another definition refers to as 'an effective reconstitution of control under attack from intelligent adversaries', being the resiliency only defined in terms of response to intelligent actors. In addition, in [10], it stated that 'A resilient control system is one that maintains state awareness and an accepted level of operational normalcy in response to disturbances, including threats of an unexpected and malicious nature'. This work extends the previous definition to CPSs, where threats are those events that can hamper normalcy and destabilize control system networks, including human error and malicious attacks, complex latencies and interdependencies, intermittent communication breakdown or even a network component malfunction or failure.

There are some architectures or frameworks in the literature for improving the resilience of these systems. In Ref. [11], a centralized architecture based on a fuzzy-neural data fusion engine is considered to increase the state awareness of RCSs. Its main goal is to provide real-time

monitoring and analysis of complex critical control systems. Nevertheless, this approach is somewhat difficult to implement in a heterogeneous decentralized environment, where communication channels can suffer from malfunctions. In Ref. [12], an intelligent resilient control algorithm for a wireless networked control system, based on a quantification of the concept of resiliency in terms of quality of control, is proposed. The authors developed an intelligent resilient control algorithm that ensures operational normalcy in face of wireless interference incidents, such as radio frequency jamming or signal blocking. In Ref. [13], an autonomous decentralized resilient monitor system able to dynamically adapt and reconfigure, depending on current conditions, is proposed. This framework, however, requires modelling the sensors and plants, as well as metrics for data quality, which can be difficult to fulfil in a real distributed heterogeneous network. Finally, Ref. [14] defined an integrated diagnostic and control strategy, relying on an agent design to be resilient in terms of stability and efficiency. Nevertheless, it should be mentioned that the proposed intelligent techniques may be difficult to implement in some components of a heterogeneous network, namely, on wireless nodes.

2.2. State awareness

State awareness is an imprecise concept that is difficult to quantify and, in addition, can change its meaning according to the context in which it is applied. When applied in CPSs, state awareness can be divided into two related categories: the ability to know or estimate the necessary control system states to maintain a stable closed loop operation, and the provision of sufficient knowledge of operation to make reliable informed decisions [10, 15].

According the control theory fundamentals, the observability and state awareness are to some extent related. However, the main differentiation is that observability is an intrinsic system property, while state awareness is the actual measurement or estimation of the system states. This leads to the definition proposed in Ref. [15], where state awareness is considered as the availability of the internal system's states, either through direct measurement $x(t)$ or through derived estimates $\hat{x}(t)$ based on an observer/estimator. As such, in the case of a cyber-attack, if state awareness can be maintained in the presence of manipulated measurements, the effects of an attack can be mitigated. Otherwise, if state awareness is not fulfilled, the system most likely will be uncontrollable, resulting in physical damage and injury.

On the other hand, it is important that sufficient knowledge of operating parameters, which represent a basis for decision-making, is provided in CPSs. In these systems, some requirements for establishing performance depend on a number of metrics that are commonly based on the use of collected data. For this purpose, Rieger and Gertman [10] argue that it is necessary to consider everything that might affect the system's normalcy, to be able to maintain state awareness. In control systems, these measures are cyber and physical security, process efficiency and stability, and process compliancy. Moreover, it is important to note that for gaining state awareness, it is not enough having all the necessary sources of data. What is actually fundamental is the necessary information extracted from the data that allows to maintain the normal operation of the system. In the case of a cyber-attack, a spectrum of state awareness is available from the normal operating regime of the system to an attacker having complete access and knowledge of the system and dynamics. The latter is the worst scenario,

because it is impossible to maintain the system under control, as being in such conditions, the shutdown of the system is the only viable option.

According to the above perspectives, in this work, state awareness is considered as the availability of the necessary information that allows to maintain an acceptable level of normal operation of the system. The required information includes that associated with the physical layer of the system with the availability of the internal system states, and that stemming from the cyber layer under the form of data regarding safety, stability and efficiency of the system.

2.3. Context awareness

Context awareness is a vital feature of modern systems, such as networked systems or monitoring and control systems. With the increase of heterogeneous systems, control and monitoring techniques are now being applied to several interconnected subsystems, including human interactions. This leads to the need for new and more intelligent and adaptive approaches, capable of understanding the system's environment and adjusting their operation, accordingly. These kinds of systems that address both the dynamisms and uncertainties are referred in the literature as context awareness systems. In order to properly define context awareness, it is necessary to first define what is meant by context.

2.3.1. Context

One of the most widely accepted definitions of the term context states that 'context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves' [16]. The reader is referred to Ref. [17] and references there in, for extensions and improvements to this definition.

Currently, the notion of context can no longer be seen as a set of numerical values that characterized the situation of an entity. Recent definitions of context consider it as a collection of measured and inferred knowledge that arises from the general activity of a context awareness system. Furthermore, context is not only present when an interaction between two entities occurs, but the absence of these interactions also carries valuable information about the system itself, especially in the field of wireless networking. This motivates the development of proper mechanisms to deal with context information in the presence of imperfect, ambiguous, wrong and unknown information.

Considering this perspective, in the present work, context is a collection of measured and inferred information (potentially containing uncertain, ambiguous and unknown segments), obtained from a highly dynamic and heterogeneous environment, characterizing its current situation.

2.3.2. Context awareness system

The definition of context awareness is highly related with that of context. In 1997 and 1998, early contributions to this topic addressed the concept of context awareness in the sense of detecting, interpreting and responding to aspects of a user's environment. In Ref. [16], a

system is said to be context awareness if it is able to support dynamic changes in its behaviour, as a response to perceived changes in its context. On the other hand, in Ref. [18], it is suggested that context awareness system means that one is able to extract, interpret and use context information and that can adapt its functionality to the current context of use.

Context awareness systems must cover reasoning and processing of uncertain, ambiguous and missing information [17]. The need for this support emerges with the presence of these systems in increasingly larger, dynamic, heterogeneous and less reliable environments. Concepts such as resilience and robustness can also be included in this definition.

In this work, context awareness system is a system capable of adjusting its operation based on perceived, processed and inferred context information, obtained from highly dynamic, heterogeneous and uncertain environments.

2.3.3. Information flow in context awareness cyber-physical systems

The field of CPSs has a wide range of areas of study and applications, resulting in the existence of a large array of context awareness CPSs proposed in the literature. Surveying the literature in this topic, it is possible to identify similarities in most approaches adopted by researchers, despite their diversity. **Figure 2** presents these similarities.

In an initial step, information from the system and environment needs to be collected, usually using sensor networks. Pre-processing tasks such as information categorization, data fusion and aggregation, imputation techniques, machine learning algorithms and inference of new information are usually applied at this point [17]. Once the information is collected and pre-processed, it must be modelled and stored. Databases and ontologies are two of the most popular and used solutions for this purpose. A context model allows a high-level description of the context by defining and characterizing entities and their relationships and can be either static or dynamic [19].

The third step is commonly referred to as context inference or reasoning. Its main goal is to deduce new information based on perceived (and stored) information, allowing for a deeper characterization of the system and its environment. Inference rules, Fuzzy Logic, Hidden

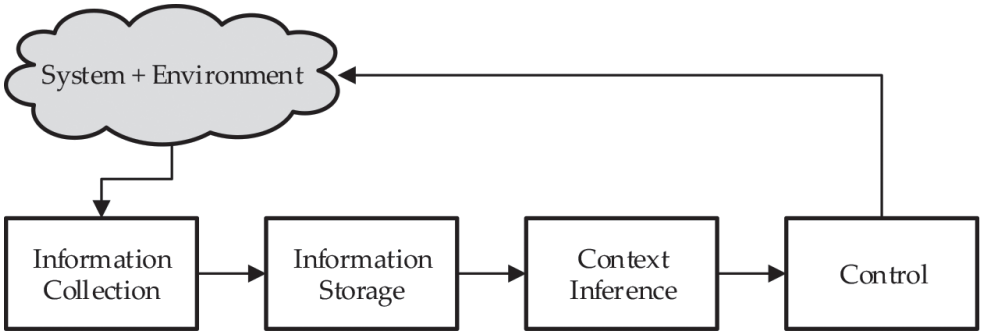

Figure 2. Information flow in context awareness CPSs.

Markov Models or Naive Bayes are some examples of context inference techniques adopted in the literature [16]. A context-aware CPS will process the output of the previously mentioned steps to identify changes in context and determine how to adapt its behaviour, in response to changes.

As CPSs usually monitor and control a given process, the fourth and final stage quite often consists in the formulation and solution of an optimization problem. Therefore, by formulating an optimization problem, a CPS can detect changes in the context and the need for adjusting its behaviour.

3. Resilient framework overview

The proposed approach for resilience enhancement is accomplished by incorporating dedicated algorithms and heuristics on a multiagent system (MAS) within a distributed middleware framework. Each agent is tailored for executing specific and coordinated tasks, namely, for detecting and recovering from physical and cyber malfunctions. The incorporation of these entities to cope with CPSs vulnerabilities is very appealing, as they provide flexibility in implementing specific functions, actions or countermeasures wherever they are needed within the network. **Figure 3** presents an overview of the proposed architecture, which comprises five layers.

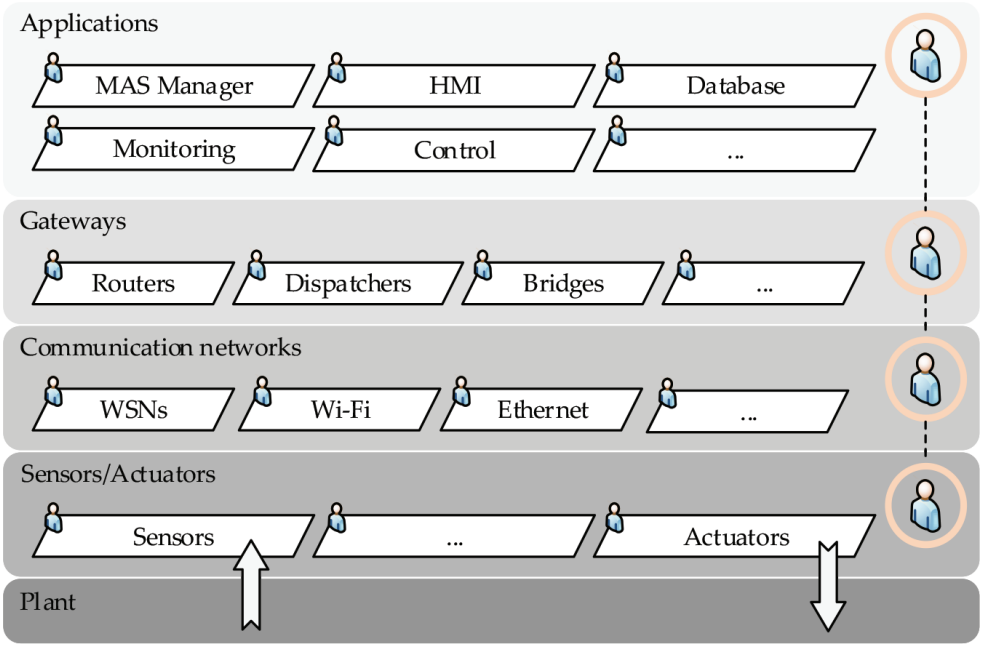

Figure 3. Resilience architecture layers.

The plant layer represents the physical infrastructure, consisting of a process under control or monitoring. This process can be a simple Single-input single-output (SISO) system or a complex system, such as a power distribution grid. To interact with the environment, it is necessary sensors and actuators that are represented in the sensors/actuators layer. Some of these devices can have computational capabilities, such as in the case of wireless nodes. The communication network layer allows the communication between the components of the system, namely, sensors, actuators and devices. These networks can be for example distributed low-power wireless networks or wired networks. The gateway layer allows the interconnection of different networks that can coexist in a CPS. This layer can have dispatchers to coordinate the communications on a WSAN or routers to relay information to the destination devices. Finally, the application layer provides a user with a number of applications, by allowing in a transparent way the interaction with networks, devices and plants.

In this approach, each layer comprises a set of agents with specific functions, which depend on where they are installed. All agents are managed by a master agent that belongs to the underlying layer and is responsible for ensuring the communication between subordinate agents and with master agents of other layers. It should be noted that in the context of CPSs, the physical system may consist of several distributed subsystems, which lead to replicating these layers for each subsystem. In this case, the master agents are responsible for ensuring the communication between these distributed layers. The proposed MAS topology implements a distributed middleware possessing the standard functions of integration, monitoring and configuration, apart from the local agents that ensure the resilience enhancement. The integration facilitates the data transfer between the process/device infrastructures and CPSs components. The monitoring function evaluates the performance of the middleware at runtime by applying some metrics. Finally, the configuration enables the definition of commands to configure data uploaded and downloaded from devices, and also the subordinate agents.

3.1. Agent features

The features of the agents proposed in this work can be aggregated into three groups (**Figure 4**): physical, cyber and multiagent system. These groups are interconnected and aim to improve the overall system resilience. It should be noted that data are collected by physical devices in the physical system, and subsequently processed by the MAS distributed over the architecture, for ensuring security, integrity and privacy to applications. Based on collected data, the MAS can assess the context and the state awareness of the system to react accordingly.

The physical system includes all physical devices, such as sensors, actuators, transductors, motors, etc. At this level, resilience is achieved by enforcing that reads/writes in the sensors/actuators are correct, not subject to any malicious attack and are not corrupted. The multiagent system aims to check the behaviour of agents and, therefore, of the entire MAS. These agents are responsible for coordinating all communications, along with the implementation of resilient mechanisms and tools. It is important to ensure in this level that the underlying agents are working properly and not suffering from any malicious attack or malfunction. The cyber system is responsible for maintaining the system security and privacy communications, and includes methods, tools and metrics implemented for improving resilience, from a cybersecurity point of view.

Figure 4. Agent features.

3.2. Agent behaviour

In the case of heterogeneous distributed CPSs, subsystems may all not possess the same characteristics and vulnerabilities, so the developed agents must be configured to provide the necessary functionalities to each subsystem. In the case of a detected event, the entire MAS has the ability to act accordingly in a way to provide resilience to the system, while guaranteeing the safety status of the system until the problem is completely addressed.

For this purpose, agents should adapt to the environment dynamics as illustrated in **Figure 5**. The behaviour of an agent depends on following four basic attributes:

- *State awareness*—State awareness has been described in Section 2.2. To take any action on the physical system, the agent needs to know the state of the system. Furthermore, the agent itself should contribute to keep the state awareness of the system, which is important to ensure resilience;
- *Context awareness*—The agent must act and behave in accordance with the context where it is installed/running. The agent needs to adapt, for example, to the physical platform where it is running and to problems taking place around them;
- *Agent awareness*—The awareness of an agent is important as attacks and malfunctions can also compromise the agent itself and contribute to degrading the entire system. As such, it is crucial to ensure that an agent is working properly;
- *User commands*—An agent can be configured by user commands.

http://dx.doi.org/10.5772/intechopen.69356

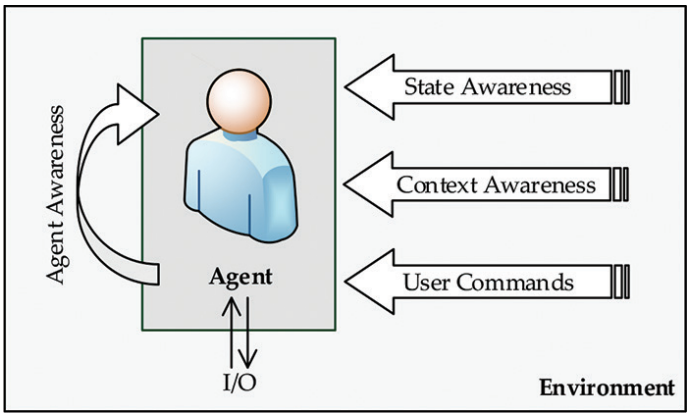

Figure 5. Agent behaviour.

4. Case study

The proposed framework for resilience enhancement is evaluated through simulations using a CPS simulator composed of a nonlinear benchmark system model, namely, a Continuous Stirred Tank Reactor (CSTR), a WSAN and additional remote devices, including a remote controller, a server where the model of the plant is running and a HMI.

4.1. Testbed simulator

The testbed simulator consists of three main components (see **Figure 6**), including a Simulink-based simulator, COntiki OS JAva Simulator [20] (COOJA) and remote devices deployed in the MATLAB environment. All of these components are distributed over five computers.

The plant, whose goal is to control or monitor, is described by a mathematical model implemented in the MATLAB-Simulink environment. In addition, ADCs and DACs associated with the sensor node and the actuator node, respectively, are included in the simulation setup to allow the interaction with the plant. To collect/send data from/to the plant and to implement the communication with remote devices, a WSAN is implemented and simulated in the COOJA environment. The COOJA simulator can emulate the operation of a real wireless device and its networking behaviour. At this level, all wireless network components are defined, by including node address, network topology, routers, along with the software that will run on each wireless node. The remote devices, namely, the controller, model-server and HMI, are considered transparently similarly as directly connected to physical hardware.

Communication and time synchronization between Simulink and COOJA is carried out using available plug-ins in the GISOO project (see [21]). GISOO is a virtual testbed for simulating wireless CPS, which integrates these two simulators and also enables users to evaluate actual embedded code for the wireless nodes in realistic experiments. Additionally, the testbed allows the communication between the WSAN and remote devices using a tunslip created in a Linux

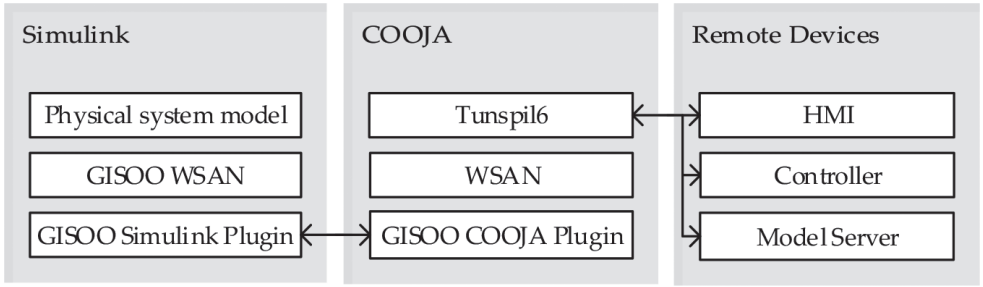

Figure 6. Testbed simulator.

environment. Finally, is should be mentioned that all the devices and nodes in this testbed have an IPv6 address, which allows the communication by User Datagram Protocol (UDP).

4.1.1. CSTR plant

The CSTR benchmark system comprises a constant volume reactor cooled by a co-current single coolant stream, as shown in **Figure 7**, where an irreversible exothermic reaction in a liquid medium takes place within reservoir (see [22]). The reactor's main purpose is to deliver the concentration of the outlet effluent C_A at a prescribed value, by manipulating the coolant flow rate q_c circulating in the reactor's jacket. The process can be described by the following differential equations:

$$\frac{dC_A}{dt} = \frac{q_A}{V}\left(C_{A,i}(t) - C_A(t)\right) - \gamma_0 C_A(t) \exp\left(-\frac{E}{R \cdot T_A(t)}\right) \tag{1}$$

$$\begin{aligned}\frac{dT_A}{dt} = {} & \frac{q_A}{V}\left(T_{A,i}(t) - T_A(t)\right) - \gamma_1 C_A(t) \exp\left(-\frac{E}{R \cdot T_A(t)}\right) + \\ & \gamma_2 q_c(t) \left(1 - \exp\left(-\frac{\gamma_3}{q_c(t)}\right)\right)\left(T_{C,i}(t) - T_A(t)\right)\end{aligned} \tag{2}$$

where C_A and T_A denote the concentration and temperature in the tank, assuming that the reactor is perfectly mixed and q_c the coolant flow rate. The remaining parameters of the system borrowed from [23] are presented in **Table 1**.

Taking into account the nominal values for the CSTR shown in **Table 1**, the operating region is constrained to:

$$\begin{aligned} 0 < C_A < 1.00 \text{ mol/l} \\ T_A > 350.00 \text{ K} \\ 0 \le q_c \le q_{c,max} \text{ l/min} \end{aligned} \tag{3}$$

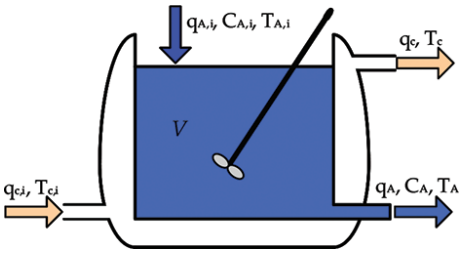

Figure 7. CSTR plant.

4.1.2. WSAN

The WSAN infrastructure includes three Crossbow TelosB nodes within the simulation environment (**Figure 8**). One of the nodes is set as a sensor and used to collect the concentration of the reactor's outlet effluent C_A, while a second node is used as actuator, associated with the coolant flow rate q_c circulating in the reactor's jacket. In addition, the remaining node is included as a sink, deploying a border router that implements the Routing Protocol for Low Power and Lossy Networks (RPL). The border router together with Tunslip allows IPv6 communication with WSAN nodes. In normal operation, the sensor node sends collected data to the applications, and the actuator node receives data from the applications through the sink node. However, all nodes have the ability to communicate directly one another, whenever necessary.

4.1.3. Remote devices

Three remote devices are used to allow the interaction with the rest of the system. Each of these devices is located on a remote computer. Through the HMI, it is possible to configure

Parameter	Description	Nominal Value
q_A	Process flow rate of component A	100 l min^{-1}
$C_{A,i}$	Feed concentration of component A	1.00 l min^{-1}
$T_{A,i}$	Feed temperature	350.00 K
$T_{C,i}$	Inlet coolant temperature	350.00 K
E/R	Activation energy	1.00×10^4 K
V	Volume of the tank	100.001 l
γ_0	Pre-exponential factor	7.20×10^{10} min^{-1}
γ_1	Constant	1.44×10^{13} l K min^{-1}
γ_2	Constant	0.01 l^{-1}
γ_3	Constant	7.00×10^2 min^{-1}

Table 1. CSTR parameters.

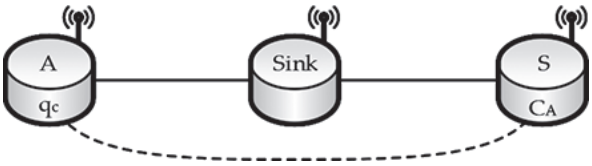

Figure 8. WSAN topology.

the entire system and observe current states, allowing the interaction with a user. The model-server is used to record historic data of the system and provide a model of the physical process to predict its response. This model is based on Eqs. (1) and (2), considering constrains in Eq. (3) and the constants in **Table 1**. Finally, the remote controller is implemented based on a Mamdani-type Fuzzy PID controller (**Figure 9**).

4.2. Multiagent system framework

The multiagent framework developed for this testbed is presented in **Figure 10**. Considering the layers presented in **Figure 3**, it is possible to observe that in this case the developed agents are distributed over the applications and the sensors/actuators layer. Each agent is responsible for a specific task and is coordinated by a master agent. Moreover, every message transmitted over the network comprises a header and a payload. The message payload contains the Message Type: the message can be originated from the systems' applications or from a node; Device ID: denoting the device address; Control ID: the command flag for local agents; Agent ID: agent's identifier; Agent MSG: data provided by an agent.

Master Agent—The master agent's main goal is to carry out extensive management routines related to subordinate local agents and to coordinate all communications. This agent is also responsible for monitoring the status of all local agents and, in case of an agent crash, to relaunch them.

Security Agent—The security agent is responsible for periodically analyzing important variables of the system for coherence, as well as the messages' structure. If any anomaly is detected, the system is switched to a safe mode operation.

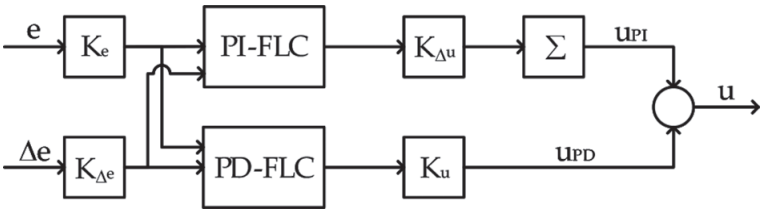

Figure 9. PID-Fuzzy logical control schematics.

http://dx.doi.org/10.5772/intechopen.69356

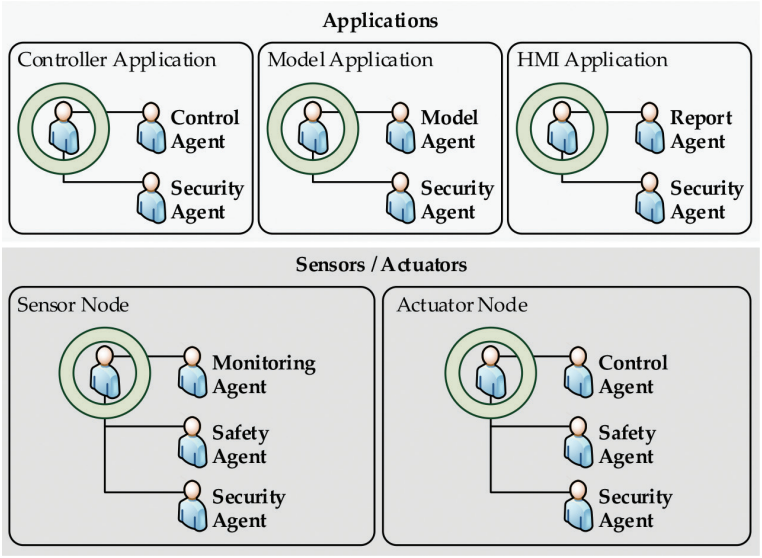

Figure 10. Multiagent framework.

Monitoring Agent—The monitoring agent is responsible for collecting data from the environment and accommodating possible outliers in raw readings. The local detection and accommodation of outliers is based on the approach suggested in [24]. It is also in charge for checking whether readings are within valid limits of operation, defined by a user.

Control Agent—In an actuator node, the control agent is responsible for sending to a digital-to-analogue converter the control actions received from the control application. This agent is also responsible for testing the periodicity of received control actions, through which communication disturbances or even a breakdown can be detected. If it is the case, the system switches to a safe mode operation. In the control application, this agent receives the sensor readings and applies a control algorithm to return the respective control action to the actuator's node. This agent is also responsible for testing the periodicity of sensor readings, and if any malfunction is detected, the system switches similarly to a safe operation mode.

Model Agent—The model agent's main goal is to predict the physical system behaviour, as well as other important components of the system. Besides, this agent receives sensor readings and control actions in order to update the underlying models. Further, this agent is crucial to ensure a safe operation mode whenever it is not possible to collect sensor readings.

Safety Agent—The safety agent is responsible for ensuring a safe operation mode, which is needed in case of communication link breakdown, remote controller's malfunction or even user induced errors/commands. In safe mode, the sensor's node safety agent, considering the context and the underlying problem or malfunction, will decide where to send readings. The actuator node's safety agent, in a similar way, decides if received control actions should be used. If they are to be rejected, local control actions based on a prescribed heuristic, such as an on-off approach, will provide the underlying control actions, and assuming the most recent available reference signal. In the case where it is not possible to have access to sensor readings, a predictive model would be used instead.

Report Agent—The purpose of this agent is to provide a user with relevant information from the system operation. It also allows the interaction between users, agents and the system, by processing users' requests.

4.3. Experiments

This section is devoted to assessing the proposed enhancing resilience framework, evaluating agent's behaviour along with the network in dealing with particular vulnerabilities on the RCS. In this context, readings are sampled from the sensors' ADCs, at a frequency of 2 Hz. To assess the effectiveness of the proposed MAS framework, in all the experiments, the control goal was to keep the concentration of the outlet effluent C_A at some prescribed values.

Two experiments were carried out, and the outcomes were discussed. In the following figures, normal operation of the system without any resilience framework is shown in blue, the operation with the MAS-proposed framework is in red, and the reference signal is presented in black. On the other hand, the flow rate refers to the underlying control actions.

4.3.1. Jamming attack

This experiment considers a jamming attack in the sink node, which prevents it from forwarding any data to other nodes and applications and receive data from the WSAN. **Figure 11** shows an attack occurring between 3.06 and 3.30 min. As can be observed, the MAS is effective in maintaining the concentration level in the neighbourhood of the most recent received reference from the server, by incorporating a safeguard approach where sensor node sends readings directly to the actuator node. When communication is restored, jamming is blocked, and the normal operation of the system is resumed.

4.3.2. Node lost

In this experiment, the communication with the sensor node is lost, due to node's malfunction. Possible causes may include power failure or congestion on the radio receptor, just to name out a few. In this case, as can be observed in **Figure 12**, the model application sends to the controller an estimation of the system output between 2.40 and 3.30 min. Once again, the MAS is effective in keeping the normal operation of the whole system until the sensor is again functional.

http://dx.doi.org/10.5772/intechopen.69356

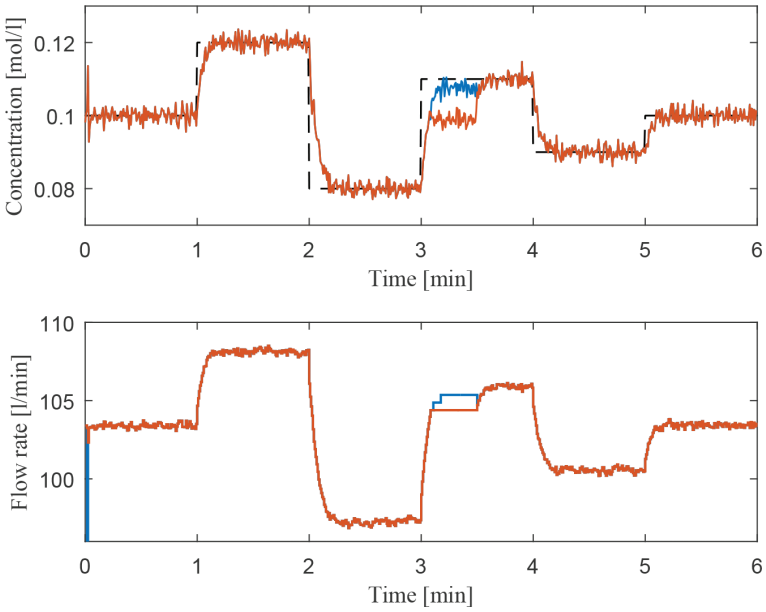

Figure 11. CSTR system: jamming attack.

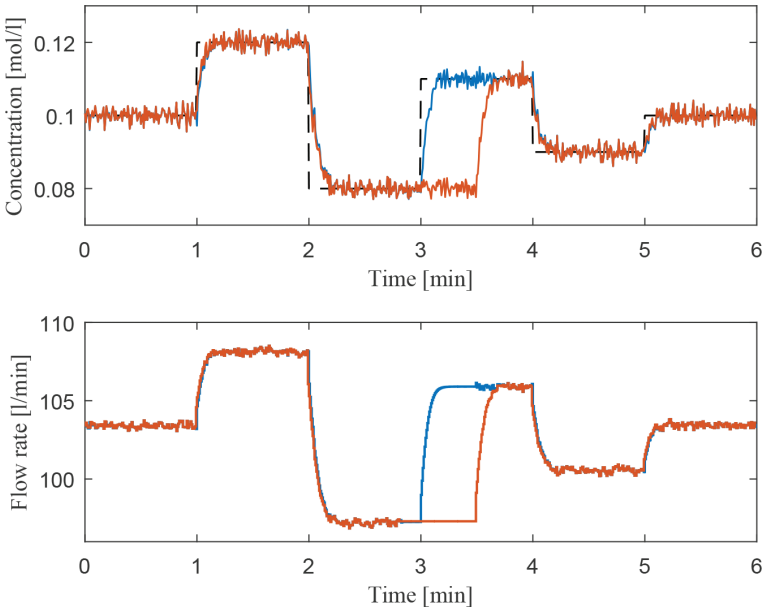

Figure 12. CSTR system: node lost.

5. Conclusions

This work dealt with a multiagent-based framework capable of improving the resilience of cyber-physical systems, as well as providing the necessary flexibility to deploy specific functions where actions or measures are needed. This allows obtaining improved responses at execution time, autonomy, services continuity and superior levels of scalability. The proposed framework focuses essentially on issues related to cyber-security and physical-security. In addition, the development of a hierarchical methodology embeds and prioritizes incoming information to ensure state awareness and context awareness, while managing and optimizing the system's response. In addition, a testbed simulator is presented, comprising a CPS with a physical process where some illustrative experiments were conducted to assess the relevance of the proposed approach. Results showed the effectiveness and pertinence of the proposed multiagent framework in the context of CPS.

Acknowledgements

Fábio Januário acknowledges Fundação para a Ciência e Tecnologia (FCT), Portugal, for the Ph.D. Grant SFRH/BD/85586/2012.

Author details

Fábio Emanuel Pais Januário[1,2]*, Joaquim Leitão[2], Alberto Cardoso[2] and Paulo Gil[1,2,3]

*Address all correspondence to: f.januario@campus.fct.unl.pt

1 Electrical Engineering Department, Faculty of Science and Technology, NOVA University of Lisbon, Campus de Caparica, Portugal

2 CISUC – Center for Informatics and Systems of the University of Coimbra, Coimbra, Portugal

3 Centre of Technology and Systems (CTS), UNINOVA, NOVA University of Lisbon, Campus de Caparica, Portugal

References

[1] Xu T, Masys A.. Critical Infrastructure Vulnerabilities: Embracing a Network Mindset. In: Masys A., editor. Exploring the Security Landscape: Non-Traditional Security Challenges. 1st ed. Cham: Springer International Publishing; 2016. p. 177-193. DOI: 10.1007/978-3-319-27914-5_9

[2] Romanovsky A., Ishikawa F., editors. Trustworthy Cyber-Physical Systems Engineering. Boca Raton: CRC Press; 2016. 462 p.

[3] Jin X., Haddad WM., Yucelen T. An Adaptive Control Architecture for Mitigating Sensor and Actuator Attacks in Cyber-Physical Systems. IEEE Transactions on Automatic Control. 2017;PP(99):1. DOI: 10.1109/TAC.2017.2652127

[4] Rieger C, Zhu Q, Basar T. Agent-based cyber control strategy design for resilient control systems: Concepts, architecture and methodologies. In: 5th International Symposium on Resilient Control Systems. IEEE; 2012. pp. 40-47. DOI: 10.1109/ISRCS.2012.6309291

[5] Yuan Y, Zhu Q, Sun F, Wang Q, Basar T. Resilient control of cyber-physical systems against Denial-of-Service attacks. In: 6th International Symposium on Resilient Control Systems (ISRCS). IEEE; 2013. pp. 54-59. DOI: 10.1109/ISRCS.2013.6623750

[6] Ali S, Qaisar S, Saeed H, Khan M, Naeem M, Anpalaga A. Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. Sensors. 2015;**15**(4):7172-7205. DOI: 10.3390/s150407172

[7] Hollnagel E., Nemeth C.P., editors. Resilience Engineering Perspectives, Volume 2: Preparation and Restoration. Boca Raton: CRC Press; 2016. 310 p.

[8] Arghandeh R, Meier A, Mehrmanesh L, Mili L. On the definition of cyber-physical resilience in power systems. Renewable and Sustainable Energy Reviews. 2016;**58**:1060-1069. DOI: http://dx.doi.org/10.1016/j.rser.2015.12.193

[9] Mitchel SM, Mannan MS. Designing resilient engineered systems. Chemical Engineering Progress. 2006;**102**(4):39-45

[10] Rieger C, Gertman D, McQueen M. Resilient control systems: Next generation design research. In: 2nd Conference on Human System Interactions. IEEE; 2009. pp. 632-636. DOI: 10.1109/HSI.2009.5091051

[11] Wijayasekara D, Linda O, Manic M, Rieger C. FN-DFE: Fuzzy-Neural data fusion engine for enhanced resilient State-Awareness of hybrid energy systems. IEEE Transactions on Cybernetics. 2014;**44**(11):2065-2075. DOI: 10.1109/TCYB.2014.2323891

[12] Ji K, Wei D. Resilient control for wireless networked control systems. International Journal of Control, Automation and Systems. 2011;**9**(2):285-293. DOI: 10.1007/s12555-011-0210-7

[13] Garcia H, Lin W, Meerkov S. A resilient condition assessment monitoring system. In: 5th International Symposium on Resilient Control Systems. IEEE; 2012. pp. 98-105. DOI: 10.1109/ISRCS.2012.6309301

[14] Rieger C, Villez K. Resilient control system execution agent (ReCoSEA). In: 5th International Symposium on Resilient Control Systems. IEEE; 2012. pp. 143-148. DOI: 10.1109/ISRCS.2012.6309308

[15] Melin A, Ferragut E, Laska J, Fugate D, Kisner R. A mathematical framework for the analysis of cyber-resilient control systems. In: 6th International Symposium on Resilient Control Systems (ISRCS). IEEE; 2013. pp. 13-18. DOI: 10.1109/ISRCS.2013.6623743

[16] Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the internet of things: A survey. IEEE Communications Surveys & Tutorials. 2014;**16**(1):414-454. DOI: 10.1109/SURV.2013.042313.00197

[17] Yürür Ö, Liu CH, Sheng Z, Leung VCM, Moreno W, Leung KK. Context-awareness for mobile sensing: A survey and future directions. IEEE Communications Surveys & Tutorials. 2016;**18**(1):68-93. DOI: 10.1109/COMST.2014.2381246

[18] Truong H, Dustdar S. A survey on context-aware web service systems. International Journal of Web Information Systems. 2009;**5**(1):5-31. DOI: http://doi.org/10.1108/17440080910947295

[19] Adomavicius G., Tuzhilin A. Context-Aware Recommender Systems. In: Ricci F., Rokach L., Shapira B., Kantor, P., editors. Recommender Systems Handbook. Boston, MA: Springer US; 2011. p. 217--253. DOI: 10.1007/978-0-387-85820-3

[20] Contiki OS. Contiki: The Open Source OS for the Internet of Things [Internet]. Available from: http://www.contiki-os.org/ [Accessed: 6 March 2017]

[21] Aminian B, Araujo J, Johansson M, Johansson K. GISOO: A virtual testbed for wireless cyber-physical systems. In: IECON 2013—39th Annual Conference of the IEEE. IEEE; 2013. pp. 5588-5593. DOI: 10.1109/IECON.2013.6700049

[22] Nguyen H-N. A benchmark problem: The non-isothermal continuous stirred tank reactor. In: Constrained Control of Uncertain, Time-Varying, Discrete-Time Systems. Cham: Springer International Publishing; 2014. pp. 181-187. DOI: 10.1007/978-3-319-02827-9_8

[23] Lightbody G, Irwin G. Direct neural model reference adaptive control. IEE Proceedings—Control Theory and Applications. 1995;**142**(1):31-43. DOI: 10.1049/ip-cta:19951613

[24] Januário F, Amâncio S, Catarina L, Luis P, Cardoso A, Gil P. Outliers accommodation in fuzzy control systems over WSAN. In: Volume 255: Intelligent Decision Technologies. Netherlands: IOS Press; 2013. pp. 334-343. DOI: 10.3233/978-1-61499-264-6-334