Adaptive Supervisory Framework for Cyber-Physical Systems

Joaquim Leitão*, Paulo Gil†, Bernardete Ribeiro* and Alberto Cardoso*
*Center for Informatics and Systems of the University of Coimbra (CISUC)
Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal
{jpleitao, bribeiro, alberto}@dei.uc.pt
†Centre of Technology and Systems (UNINOVA-CTS)
Department of Electrical Engineering, Universidade NOVA de Lisboa, Monte de Caparica, Portugal
psg@fct.unl.pt

***Abstract*—Cyber-Physical Systems (CPS) result from the aggregation of the cyber and physical worlds into a single spatially distributed macro-system, relying on sensor and actuator networks to perform its optimised management.**

CPS must be capable of reacting to perceived and inferred knowledge about the systems they coordinate, demonstrating context-aware and self-adaptive characteristics. Popular solutions proposed in the literature are inspired by feedback-loop control theory, according to which a repeated analysis of the state of the managed environment is performed, followed by a review and adjustment of operational objectives and constraints.

Despite current advances CPS still face important challenges, among which the integration of heterogeneous entities (where human presence must also be accounted for), high-order predictive models, lack of a holistic design perspective and inclusion of context information are highlighted.

The proposed research aims to contribute towards a clarification of these gaps by addressing the design of CPS, integration of dynamic and heterogeneous entities, and improving their overall optimised behaviour under changes and uncertainties in perceived and inferred context information. To this end an Adaptive Supervisory Framework is intended to be developed.

***Index Terms*—Cyber-Physical Systems, Context-Awareness, Human-in-the-Loop, Internet of Things, Supervision.**

I. Introduction

Over the last years technology has taken over most of our daily routines, becoming an integral and necessary part of almost any everyday life domain. A growing and vast range of technological resources is being shifted into equipments used by many of us in our daily routines.

Besides their high processing power, devices featured in this undergoing digital revolution are enhanced with communication, perception, intelligence and reasoning capabilities. With such attributes, this new generation of digital and computerized systems aims to bridge the cyber and physical worlds, closing gaps between these two (distinct) realities and creating opportunities to extend existing knowledge in these fields [1].

The Internet of Things (IoT) and Cyber-Physical Systems (CPS) are two technological revolutions that implement this paradigm shift; IoT is strongly linked with connectivity and inter-networking of physical devices, while CPS are more focused on achieving an optimised management interconnected subsystems, subject to time-varying constraints and objectives.

These technological revolutions raise important challenges yet to be properly addressed and solved by the scientific community in this field: as novel digital systems are composed of distinct and dynamic subsystems in constant interaction, proper modelling and integration mechanisms need to be developed. As large volumes of data can be easily collected nowadays, selective information processing mechanism are also required in order to assure the real-time nature of CPS.

Moreover, *context-awareness* is also expected to play a crucial role in this task, allowing CPS to explore measured data and inferred contextual information. The use of context information has been successfully explored in small and well-defined application domains; however, their complexity challenges its integration in modern IoT and CPS case studies.

Human presence must also be accounted for, as it contributes to heterogeneity, dynamics and uncertainty, challenging the management of these systems. CPS that take such human interactions into consideration have been classified in the literature as Human-in-the-Loop Cyber-Physical Systems (HiTLCPSs) [2].

The proposed research intends to make contributions to these challenges, by addressing the design of CPS, integration of dynamic and heterogeneous entities, and improving their overall optimised behaviour under changes and uncertainties in perceived and inferred context information. To this end an Adaptive Supervisory Framework is intended to be developed.

The proposed problem emerged in the sequel of prior works of the author [3] where an optimised management of underground conduits in existing water drainage systems was investigated and performed. This problem was further studied, leading to its publication and presentation in an international conference [4], being distinguished with the *Best Paper Award*.

This past work highlighted the need for the development of intelligent and fully automated systems, capable of managing environments of this nature, motivating a more comprehensive study comprising other resources besides water. The current document was prepared based on the thesis proposal document produced by the *PhD* candidate Joaquim Leitão [5].

The remainder of this document is organised as follows: Section II introduces a series of important concepts adopted throughout the document. A literature review of the fields more directly related with this research work is conducted in Section III. Section IV presents the research problem and the Adaptive Supervisory Framework, the main expected outcome of this work. Finally, Section V concludes the document.

II. Concepts Definition

The current section focuses on the definition of important concepts related to this work, and used throughout the document.

A. Cyber-Physical Systems

The concept needed to be defined in first place is that of Cyber-Physical Systems (CPS). The authors of [6] have proposed the following formal definition:

Definition 1 (Cyber-Physical Systems) *"Cyber-Physical Systems (CPS) are physical and engineered systems whose operations are*

Joaquim Leitão wishes to acknowledge the Portuguese funding institution Foundation for Science and Technology (FCT) for supporting this research under the Ph.D. grant SFRH/BD/122103/2016.

monitored, coordinated, controlled and integrated by a computing and communication core."

Based on the presented definition, a CPS can be considered as a system that monitors, coordinates and controls a given physical process, which usually has a distributed nature. CPS bridge the cyber and physical worlds, by integrating computing capabilities into modern physical systems with the ultimate goal of monitoring and controlling the dynamics of these systems in an automated way.

Recent technological trends and advances have highlighted the need for a new generation of engineered and intelligent systems. This new generation improved on previous ones by not only growing in dimension, but also by becoming composed of several interconnected heterogeneous and dynamic subsystems [1], [6], [7].

The proliferation of low-cost sensors with high accuracy and low power consumptions allowed existing digital systems to grow in dimension and complexity, requiring the development of novel algorithms that enabled an adaptive, efficient, reliable and real-time management of larger and more complex environments. Human entities also gained importance, strongly conditioning the operation of novel CPS. Figure 1 depicts this scenario.

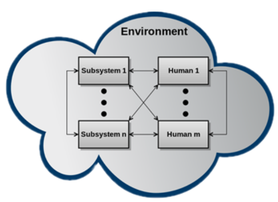


Figure 1: Typical composition of current Cyber-Physical Systems.

In modern CPS proposals, the main characteristics of these systems are [1], [6]: (i) Real-time nature; (ii) *Context-Awareness*, (the management of the environment is conditioned by perceived and inferred state information); (iii) Distributed nature; (iv) Ability to deal with uncertainty and unexpected events; (v) Adaptation capabilities, Resilience and Security.

Applications of CPS can be found in a vast range of domains, including the automotive industry, environment monitoring, critical infrastructures' management, healthcare, social networking and military applications, just to name a few [1]. Examples of such applications include, for instance, power management and supply systems such as smart grids; water resource management and supply systems; disaster response systems, such as emergency evacuation planners; autonomous vehicles and automatic pilot avionics; process control systems; medical monitoring systems; and smart homes and smart environments [6].

B. Internet of things

According to [8], the first formal definition of the term *"Internet of Things"* (IoT) dates back to 1998, and was made by Kevin Ashton. As this field matured and grew in popularity and dimension, IoT became a complex and vast area of research on its own. As a result of recent technological advances IoT and CPS are aligned with a new observed paradigm shift, where the need for the development of a new generation of systems with the mentioned capabilities is pushed up.

According to [9], a current and commonly accepted definition for this term has been presented by [10]:

Definition 2 (Internet of Things) *The Internet of Things is "a dynamic network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'Things' have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network."*

In IoT, a *'Thing'* is any object of the physical or cyber worlds (that is, physical or virtual devices) capable of being identified and integrated into communication networks [9], [11].

From a research perspective, IoT systems should have certain properties and characteristics. These include *intelligence*, by means of proper inference techniques to reason and extract new information from previously collected data; communication capabilities with existing and new 'Things'; ability to track the different 'Things' deployed in a given environment; real-time nature and scalability characteristics, by exploring resources such as cloud computing and the *everything-as-a-service* model [12].

C. Human-in-the-Loop

Human-in-the-Loop is a term often applied to scenarios that require an active presence and intervention of human entities. The authors of [2] propose the application of term to CPS, where human presence and behaviour is no longer considered an external factor, but instead as a strong source of dynamic behaviour, heterogeneity and uncertainty becoming a key component of the systems and environment under monitoring and optimisation.

Intervention of human entities in CPS can be performed at different levels, as highlighted in [2], [13]: (i) Human control, where humans have a direct influence on the optimisation decisions; (ii) Human monitoring, where existing systems observe and analyse human activities in order to take proper actions, but humans do not have a direct influence over their management decisions. In the scope of CPS these can either be *open-loop* or *closed-loop* systems[1].; (iii) Hybrid system, where human-related information is used as feedback in control-loops, while also accepting direct human inputs.

D. Context-Awareness

The final concept to be defined in this section is that of *context-awareness*. Because of the strong relationship existing between the concepts of context and context-awareness, a clarification of both these terms will be conducted.

Over the years, accepted definitions for both these concepts have substantially changed in the literature. This has been more notorious for the definition of context information. Early definitions considered context as a set of values that characterised the situation of an entity (usually by answering the following three questions: "Where you are? Who you are with? And What resources are nearby?" [14]).

In more recent definitions context is regarded as a collection of measured and inferred knowledge, which arises from the general activity of a context-aware system. Context is also present in the absence of interactions between users and applications.

Defining context-awareness has not experienced deep changes over time. The main updates relied on the definition of context, whereas advances in the context-awareness definition usually occur as a collateral adjustment to the context concept.

In the current discussion we proposed the following definitions for these two concepts, extending the definitions proposed by [15], [16]:

Definition 3 (Context) *We can define context as a collection of measured and inferred information, potentially containing uncertain, ambiguous and unknown segments, obtained from a highly dynamic*

[1]Open-loop systems monitor and analyse human-related information and report computed results, while closed-loop systems use perceived human information to actively contribute towards the optimisation goal. However, in these closed-loop systems the human only has an indirect influence over optimisation decisions.

and heterogeneous environment, which characterises its current status.

Definition 4 (Context-Aware System) *A Context-Aware System (C-AS) is a system capable of adjusting its operation based on perceived, processed and inferred context information, obtained in highly dynamic, heterogeneous and uncertain environments.*

III. Related Work

The current section presents a literature review on four main topics related with the research work covered in this document. Starting with Context-Awareness, a study on context-aware software systems is carried out in III-A, complementing presented context and context-awareness definitions.

Covered literature on this topic is extended in III-B, which addresses the topic of self-adaptive software systems. Recommended approaches and models concerning the development and design of such systems are also discussed in this section.

The next topic to be discussed is concerned with CPS behaviour optimisation under multiple objectives, constraints and uncertainties in information relevant to this task. This discussion is carried out in III-C.

The last topic to be covered in the present literature review is related to the identification of recurrent patterns of behaviour (section III-D).

This section ends with a final overview and summary of the discussed topics, presented in III-E.

A. Context-Awareness

In Section II-D a definition of *context* and *context-awareness* concepts was carried out, with a very short overview of the evolution of these two definitions. Even though no single definition for each of these concepts has been unanimously accepted in the literature, common ideas are still shared among distinct definitions, motivating the proposal of our own clarification of these concepts.

In this section a survey on context-aware applications is performed, providing insights on the characteristics, design and overall development process of such applications.

We start this survey of context-aware applications by extending previous discussions on the definitions of these terms, presenting the state-of-the-art in their conceptualization. The main topics to be covered include current context types and categorisation schemes, main features of context-aware applications, and interactions between C-AS and human entities.

Context Categorisation Scheme

Considering that no consensus regarding the definition of context has been formally achieved in the literature, it comes as no surprise that different context categorisation schemes have also been proposed based on different perspectives on this concept.

Categorisation schemes proposed by sounding names in this field have been distinguishing context information types based on its importance to context-aware applications and dependence on other context sources. Overall, researchers have supported the adoption of two main categories of context - *primary* vs *secondary* context - differing on the distinction between them.

Perera *et al.* [12], propose as a revision of previous categorisation schemes from an operational perspective. Because this categorisation has a strong dependency on what and how information was acquired, the same data can be considered as primary context information in one scenario and as secondary in another:

- **Primary Context**, comprising any information retrieved without using existing context information and without performing any kind of sensor data fusion (e.g., GPS readings as location information).
- **Secondary Context**, comprising any information that can be computed using primary context (e.g., computing the distance between two entities based on their location coordinates) and information retrieved based on primary context (e.g., phone number, email address or birth date of a personal identity).

Interaction with Context-Aware Systems

In earlier stages of the development of such systems human entities could be considered as external factors that had little to no influence in the overall operation. This is no longer the scenario, as these entities are becoming more and more involved in the activities of modern digital systems, being considered an integral component of these systems.

As a result, one can find a plurality of vectors along which human interactions may occur in modern digital systems. With respect to Context-Aware Systems (C-AS), [17] identified three levels of interactivity between C-AS and their human entities: (i) Personalisation; (ii) Passive Context-Awareness; and (iii) Active Context-Awareness.

In a later work, Alegre *et al.* [18] proposed an alternative categorisation of interactions with C-AS that improved on the previously cited work. Classification of interactions with C-AS was performed in two different modalities - *Execution* and *Configuration* - each supporting *Active* or *Passive* behaviours: (i) Active Execution; (ii) Active Configuration; (iii) Passive Execution; and (iv) Passive Configuration.

Features of Context-Aware Applications

During their study of *context* and *context-awareness*, besides the proposal of valuable insights to the definition of these two terms, Abowd *et al.* also clarified what features a C-AS should support. In the view of the authors, context-aware applications and systems should provide support for proper (i) *Presentation*, (ii) *Execution* and (iii) *Tagging* features.

The first feature is related with the ability to filter context information presented to the users, avoiding presenting unrelated or irrelevant information. The second feature is concerned with the automatic execution of services. Based on currently perceived and inferred context information, C-AS must be capable of identifying new goals and needs that must be satisfied. The last feature, which the authors identify as being equal to the term *contextual augmentation* defined by Jason Pascoe [19], addresses the need to tag context information together with sensor data for later processing and understanding.

Similarly to what was verified in the categorisation of interactions with C-AS, Alegre *et al.* also proposed to extend this set of features, taking into account their proposed improvements concerning the categorisation of interactions with C-AS. The following four features are considered: (i) Presentation of information to stakeholders; (ii) Active or Passive execution of a service; (iii) Active or Passive Configuration of a service; and (iv) Tagging context information.

Context Life Cycle

The field of CPS has a wide range of areas of study and application of interest to the scientific community. Because of this scenario, one can verify the existence of a large array of context-aware CPS proposed in the literature, on scientific journals of different topics [12], [18], [20].

Such diversity implies the presence of applications focusing on fields of very distinct natures, namely mobile networks and communications, energy and water supply optimisation, industrial environments, human-in-the-loop and human-in-the system, just to name a few.

Despite the diversity in this field, surveying literature regarding this topic it is possible to find similarities in most approaches proposed by researchers. Four main common steps, performed in sequence and in loop, have been identified: (1) Information Collection; (2) Information Storage; (3) Context Inference; (4) Context Dissemination. Some authors and researchers also consider the definition of middleware and abstraction layers to enable the communication and management of data between the presented modules.

1) Information Collection: An initial step, applied in all scientific works of this nature, consists in the acquisition of information from the system and its environment, typically using sensor networks or other available means. The goal of this step is to transform sensed raw data into low-level context information which will serve as input to the next module.

Information collection in C-AS and CPS comprehends more than a simple data acquisition from multiple and heterogeneous sources. In some scenarios one is interested in working with information that results from the aggregation of data from distinct sources. In this sense, pre-processing tasks such as categorisation or data fusion and aggregation are often applied. Makris *et al.* [16] propose a division of tasks performed during information collection into four main sub-functionalities:

- *Monitoring* of context information, supporting self-adaptation and self-reconfiguration mechanisms.
- *Gathering* tasks, mainly consisting in sensor data fusion and aggregation, but also contemplating some classification and labelling tasks.
- *Prediction* and *Learning* tasks, employed not only as a means of replacing missing or invalid measured data, but also to adapt collection and monitoring tasks to changes in users' behaviour and in the surrounding environment. Some researchers also propose the inference of new information from previously collected and perceived information [21].

2) Information Storage: Once the information is collected, and pre-processed, it must be modelled and stored. In the literature, context models are often used as a representation technique that defines how context data is structured and highlights interactions, properties and relations between different concepts and entities that compose the system in question.

A context model allows for a high-level description of the context by defining and characterising entities and their relationships, which can be either static or dynamic [12], [22]. Static models only support a predefined set of context information with fixed attributes and relationships with other entities, while dynamic models allow for these properties to change over time.

Extensive enumerations and study of context modelling techniques have been presented in the literature [12], [16] with databases and ontologies appearing to be the most popular and used approaches for this purpose.

Regarding databases, their most attractive properties involve the ability to store massive amounts of data with some support for more complex queries. One of the main drawbacks of such techniques has to do with the difficulty in changing the data structure, although recent trends and insights in NoSQL databases appear to have addressed some of these issues [12].

When dealing with large volumes of data, context retrieval in ontologies is still a computationally expensive task. Despite this disadvantage their support for reasoning engines has been one of the deciding characteristics responsible for making this one of the preferred mechanisms for managing and modelling context information [12]. Furthermore, ontologies were developed targeting information and knowledge sharing, which is also an interesting and attractive characteristic when heterogeneous information sources need to be considered and integrated.

3) Context Inference: The third step considered in this listing is commonly referred to as context inference or reasoning. Its main goal is to deduce new information based on perceived (and stored) information. Within context inference, techniques capable of predicting and learning the dynamics of the subsystems and entities that compose the CPS are also considered.

A vast number of reasoning techniques have been proposed and studied in the literature which, according to Perera *et al.* [12], can be grouped into six categories: (i) *Supervised Learning*; (ii) *Unsupervised Learning*; (iii) *Rule-Based*; (iv) *Fuzzy Logic*; (v) *Ontology-Based*; and (vi) *Probabilistic*.

Clearly, each individual modelling and reasoning techniques possesses their own strengths and weaknesses, with n0 single approach outperforming the others in all possible scenarios. Therefore, multiple models and reasoning techniques adjusted to the problems being tackled must be combined in order to explore the strengths while mitigating the weaknesses of selected solutions.

4) Context Dissemination: Depending on the CPS, different approaches can be followed at this point. After information has been collected, modelled, stored and reasoned, the literature in this topic considers a rather general *context dissemination* task, responsible for delivering collected and inferred context information to consumers (which can be a context-aware application, the end user, among other possibilities).

Since many CPS seek to improve the management of a given environment with respect to certain metrics, this step usually consists in the formulation and solution of an optimisation problem. At this point CPS detect context changes and the need to adapt their behaviour (reflected by the formulation of an optimisation problem), and compute the required adaptations (by solving the optimisation problem).

B. Self Adaptive Software Systems

In this discussion Context-Aware Cyber-Physical Systems (C-A CPS) were presented as systems capable of managing contextual information originated in their subsystems and in the environment that surrounds them, being able to determine operational goals and constraints, as well as adjusting them to observed changes in context information.

According to Oreizy *et al.* [23], software systems that observe and analyse their components and surrounding environment, identifying and implemented required changes in their behaviour in order to meet their execution goals, are considered to possess *self-adaptive* characteristics.

Authors of the cited work investigated the adoption of a general-purpose approach to self-adaptive software systems relying on feedback-loops. The main goal behind the application of feedback-loops to self-adaptive systems is to keep the goals and operational objectives controlled in the target system.

Indeed, this theory provides systems with proper mechanisms to manage adaptation and system evolution, driving the state of the art in this field. Heterogeneity, dynamism and uncertainty can also be addressed exploring this theoretical background, providing the scientific foundations to achieve robustness, stability and overall optimisation of the system's operations [24]–[26].

Recent published works that explore the application of feedback-loop theory in self-adaptive software systems share a similar core of components that compose the well-known MAPE-K loop (or simply MAPE-K) [26], [27].

1) MAPE-K Loop: Inspired by the feedback-loop model, IBM researchers defined a model for the development of self-adaptive and self-managing systems composed of four main components that name the model: Monitoring, Analysis, Planning, and Execution [28]. Figure 2 presents the control-loop.

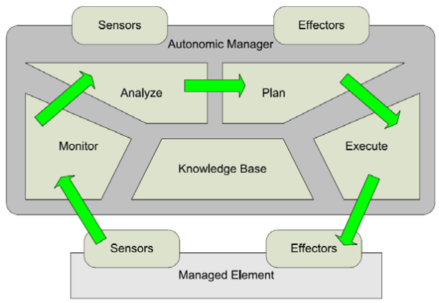


Figure 2: The MAPE-K loop (Image from [29]).

In this loop, the **Monitoring** component is responsible for all sensing and information collection tasks. Different monitoring approaches can be carried out, based on the type of information being collected (and available sensor equipment). Within the scope of the AMADEOS Project [29] *Hardware*, *Software* and *Hybrid* monitoring approaches were discussed.

Collected and sensed information must be further processed at this stage so that relevant context events can be identified. By receiving and processing information from monitoring components located in different points of the target system, the **Analyser** determines if an adaptation of the system's operations, goals or constraints is, or not, required.

After identifying the need to perform some sort of adaptation to the target system's operation, the Analyser must issue a notification so that the **Planner** can compute and define a strategy that best fulfils the new requirements. The Planner then issues sequences of discrete operations (instead of continuous signals issued in common feedback-loop controls) that compose the reconfiguration plan [25].

Finally, the **Executer** component is responsible for handling all the interactions with physical devices and software components in order to carry out the previously computed reconfiguration plan.

In Figure 2 an additional component to the MAPE-K loop can be identified, corresponding to the **Knowledge Base**. The purpose of this component is to store relevant knowledge about the CPS being monitored and adapted. The Knowledge Base has important contributions to the activity of the Analyser and Planner components.

2) MAPE-K Patterns: One major characteristic property of current C-A CPS is their distributed nature. With respect to the aforementioned MAPE-K loop, this not only means that physically distant data sources need to be considered at the Monitor component, but also that the different components can be deployed and replicated over physically distant subsystems.

In the scope of the AMADEOS Project a *MAPE Pattern* is defined as a way of distributing the MAPE elements through the system's components. Two main groups of patterns have also been identified in this project: *Formal* and *Non-Formal Hierarchy*.

Regarding Formal Hierarchy patterns, *Hierarchical Control* (Figure 3a), *Master/Slave* (Figure 3b) and *Regional Planner* (Figure 3c) are identified.

The *Hierarchical Control* pattern finds its motivation in the distributed nature of many digital systems. A layered architecture is considered, where higher layers have a more global overview of the entire system and work at longer time scales, assigning lower layers to specific parts of the system.

The *Master/Slave* considers a centralised subsystem that is responsible for orchestrating all the activities of a series of subordinates. Therefore, the centralised subsystem (the *Master*) implements the Analyser and Planner components, leaving its multiple subordinates responsible for the Monitoring and Executer components.

In the *Regional Planner* one subsystem implements the Controller component of the MAPE-K, delegating to other subsystems the implementation of the remaining components. The subsystem that implements the Planner is referred to as the *Regional Planner* and is responsible for collecting necessary information from all its subordinates in order to plan adaptations and reconfiguration actions.

Concerning Non-Formal Hierarchy patterns, authors of the cited work distinguish *Coordinated Control* (Figure 3d) from *Information Sharing* (Figure 3e) patterns.

The *Coordinated Control* Pattern is implemented when it is not feasible to implement a centralised controller. Each component of the MAPE-K loop is distributed across the different subsystems, meaning that all these components must coordinate their operations with their peers in all the other subsystems.

In contrasts, the *Information Sharing* Pattern presents similarities to the *Coordinate Control* with the main difference being that only interactions between Monitors are allowed.

3) Alternatives to the MAPE-K Loop: Despite the popularity of the MAPE-K model, attempts to extend and improve this model have been made and reported in the literature. Among the studied works in this scope, the *DYNAMICO* reference model proposed by Norha Villegas [26], [30] (Figure 4) is highlighted.

Besides addressing the hidden state of control-loop components, the Villegas claims that self-adaptation is characterised by three levels of dynamics which have not been fully addressed in past works: (i) control objectives; (ii) adaptation of the target system; and (iii) dynamic monitoring.

4) Emergence and Evolution: In highly dynamic environments changes and evolution in the behaviour and constitution of CPS are more than frequent, occurring in a continuous fashion.

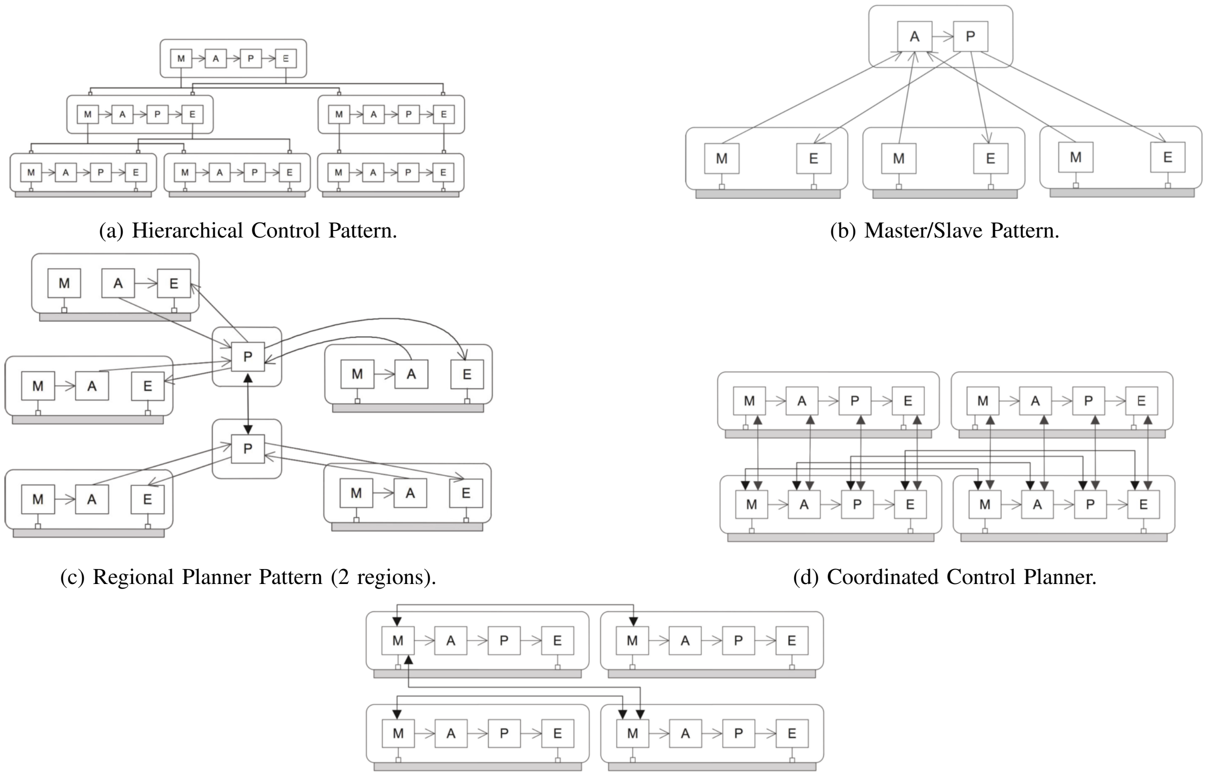
New and unpredicted behaviours must be properly identified and addressed by CPS. In a broad view, two main scenarios appear as the subject of interest of the research community in this topic [29]: Evolution of CPS and Emergent properties.

CPS evolution is concerned with design modifications on different components, often triggered by changes in the environment, resulting from its dynamic nature or due to the introduction or replacement of some of its components, services, entities, etc.

On their turn, emergent properties are phenomena manifested at macro-level carrying novelty with respect to the non-relational phenomena of any of its proper components at micro level [31], [32].

According to the cited authors emergent behaviours can be classified in four distinct categories, depending on their impact on the overall operation of the CPS (beneficial or detrimental) and prediction at micro level (expected or unexpected): (i) *Expected and beneficial emergent behaviour*, which correspond to the normal case; (ii) *Unexpected and beneficial emergent behaviour*, which have also been addressed as "positive surprises"; (iii) *Expected and detrimental emergent behaviour*, the least problematic case due to its expected nature; and (iv) *Unexpected and detrimental emergent behaviour*, which correspond to the problematic case due to its unexpected nature.

According to [29], detecting something conceptually new at macro-level is a very challenging research topic. The ability to accommodate unpredicted phenomena not accounted for in current models is



(a) Hierarchical Control Pattern.

(b) Master/Slave Pattern.

(c) Regional Planner Pattern (2 regions).

(d) Coordinated Control Planner.

(e) Information Sharing Pattern.

Figure 3: Formal and Non-Formal MAPE-K Patterns (Images from [27]).

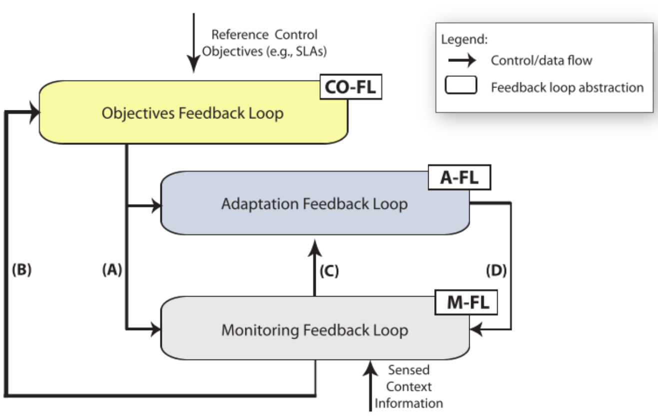


Figure 4: Block diagram illustrating the three feedback-loops proposed in the DYNAMICO reference model to address the levels of dynamics of self-adaptive, context-aware, software systems (Image from [26]).

another demanding task that this new generation of digital systems must be able to address.

Emergent properties can be associated with regularities in CPS behaviour at macro-level that are not predicted nor explained by current models of such entities. This is an important characteristic that can be explored to improve the detection and identification of such phenomena at macro-level.

In this sense, [33] proposed a library of signatures of emergence phenomena observed in digital systems. By analysing the characteristics of perceived behaviour at macro-level, regularities and common properties can be identified, resulting in the mentioned signatures of emergence.

Approaches of this nature share a common concern: an intelligent decision procedure capable of detecting anomalies and classifying them as emergent must be developed. This type of exercise lies within the task of *Unsupervised Learning*, where approaches such as clustering techniques and self-organising maps (among others) can have an important role [29].

C. Cyber-Physical System Optimisation

In Section III-A a life-cycle for context information in C-A CPS was presented, comprehending four distinct stages. With respect to the last stage, a rather general context dissemination step was considered. In many C-A CPS an optimisation of the system's behaviour is often considered, being accomplished by the formulation and solution of an optimisation problem.

In the context of the CPS being studied more than one objective and more than one constraint will usually be considered in such optimisation procedure, rendering this a *Constrained Multi-Objective Optimisation Problem*. Due to the characteristics of constituent systems, interacting entities and surrounding environments, both the operation objectives and the respective constraints will most certainly be varying over time.

A typical formulation of a Multi-objective Optimisation Problem (MOP) is as presented in Equation 1:

$$\begin{aligned} &\underset{X}{\text{Minimize}}\ \text{F(X)} = [F_1(X), F_2(X), \cdots, F_n(X)]^T \\ &\text{subject to}\ \ g_i(X) \leq 0,\ \ i = 1, 2, \cdots, m \end{aligned} \tag{1}$$

For non-trivial problems of this nature, no single solution exists simultaneously optimising each objective. In that case a plurality of solutions to the problem exist which form a set of *Pareto optimal solutions*.

Many methods to solve MOP have been studied in the literature, motivating the proposal of classification schemes to group similar approaches together. The classification scheme adopted by Yousef Sardahi in his PhD Thesis [34] is presented below: (i) *Scalarisation Methods*, which require the transformation of the MOP into a single optimisation problem by weighting the objectives individually; (ii) *Pareto Methods*, which do not aggregate the objectives in any way, searching for Pareto Optimal solutions; and (iii) *Non-Scalarisation and Non-Pareto Methods*.

Defining and solving a mathematical optimisation problem is not the only approach that can be considered when attempting to optimise the behaviour of a given system with respect to some objectives and constraints. As stated in III-B1, machine learning techniques can also be applied. Besides the ANN structures suggested in [29], approaches exploring the use of fuzzy logic for this purpose can be found in the literature applied to tasks such as energy management [35], [36].

Another important idea that cannot be omitted in this discussion is related with the presence of uncertainty in the perceived and inferred information used during the optimisation process. *Optimisation under uncertainty* is a field of study concerned with optimisation problems featuring the mentioned characteristics [37].

This field of study has also been referred to as *Stochastic Programming*; however, techniques of this nature must not be mistaken with probabilistic techniques such as Genetic Algorithms or Simulated Annealing, applied to some discrete optimisation problems. As mentioned in [38], optimisation under uncertainty involves making optimal decisions under uncertainties.

Uncertainties in optimisation problems need to be represented and modelled so that they can have a proper effect in the decision-making processes. Rockafellar distinguishes three modelling techniques: (i) Stochastic, according to which sources of uncertainties are identified and modelled as random variables, each with its respective probability distribution; (ii) Deterministic, which push uncertainties out of the problem's scope; and (iii) Range Modelling, which defines *ranges of uncertainty* for uncertainty sources, restricting quantities associated to them to particular intervals of values.

Concerning the solution of the optimisation problem, Rockafellar presents a relevant discussion topic centred on solving the problem before or after making an observation from the problem's environment (*decision first* vs *observation first*). As argued by the cited author, there is no approach that clearly outperforms the other; both cases can arise leading to scenarios where partial decisions and partial observations are made in stages and interspersed with one another.

D. Information Categorisation

Within the scope of the work being presented and discussed in this document, information categorisation is related with the identification of recurrent patterns in the behaviour of the subsystems and their surrounding environment, supported by the collection and proper analysis of information from such sources.

This is an important task for self-adaptive context-aware systems, since it allows not only for a categorisation of their behaviour but also for the identification of unexpected and anomalous situations which require an intervention and adjustment in their operation.

Considering self-adaptive systems implementing the MAPE-K reference model presented in Section III-B1, recurrent patterns of behaviour can be stored in their Knowledge Base components, which will be compared against collected sensor information so that the Analyser can determine if behavioural adjustments are required or not.

From a Pattern Recognition point of view, the identification of such behavioural patterns consists in a clustering or unsupervised learning problem. The goal of such problems is to group a set of observations in such a way that information more similar to each other is assigned to the same group (also called a cluster).

Distinct clustering techniques have been proposed in the literature, being grouped in a distinct way by different authors. Han *et al.* [39] proposes that clustering algorithms be compared based on the following properties: (i) *Separation*, where cluster overlapping may or may not be allowed; (ii) *Partitioning Criteria*; (iii) *Similarity measures*; and (iv) *Clustering Space*.

Based on these properties, the authors propose a division of clustering techniques in four main groups: (i) *Partitioning*, which iteratively builds the clusters by assigning patterns in the dataset to a cluster; (ii) *Hierarchical*, which seek to create clusters of different levels, as in a hierarchy; (iii) *Density-Based*, which considers clusters as high-density regions of the feature space, separated by sparse regions; and (iv) *Grid-Based*.

E. Summary

The current section discussed the most relevant ideas addressed in the literature concerning four main topics covered in this document.

For starters a study on context-aware software systems was carried out, extending the discussion presented in II-D. Currently, the scientific community in this topic considers two main categories of context information: *Primary* - retrieved without performing any additional interaction, computation or fusion with data from other sources - and *Secondary* information - obtained based on primary context.

Different interaction levels between human entities and C-AS are also possible, concerning how these systems react when in the presence of specific situations, and how they adjust their future operations based on past experience. In both cases human entities can either have an *Active* or *Passive* contribution to the configuration or operation of C-AS.

Browsing the literature on C-A CPS, the flow of context information in such systems appears to follow four main stages, comprising its collection, storage, inference and overall dissemination, with this last stage often considered as an optimisation exercise.

Self-Adaptive Software Systems are also related to C-A CPS in the sense that the latter ones must be capable of adapting their behaviour to observed changes in their subsystems and environment while also adjusting operation goals and constraints.

Different reference models aiming to guide the development of systems with such characteristics have been proposed in the literature, of which the *MAPE-K Loop* has received notorious attention.

The ability to work with uncertainty in collected and inferred information is also very important for this new generation of digital systems. This is by no means an easy and straightforward process. As presented in Section III-C, this is a very extensive field of research. Popular approaches to deal with mentioned uncertainty involve the identification and modelling of its sources by means of random variables described by a probability distribution. When few insights are available regarding the probabilistic nature of the uncertainties it can be useful to consider *ranges of uncertainty*.

The identification of recurrent patterns of activity and behaviour within C-A CPS' operations is also important to guarantee the self-adaptability of these systems. Clustering techniques are suited for such a task. *Partitioning* and *Hierarchical* clustering approaches are among the most popular adopted solutions, with K-Means and Hierarchical clustering being the most referenced techniques, often applied together [40]–[44].

Alternative solutions in the field of Pattern Recognition can still be identified, such as the use of Self-Organising Maps, Decision

Trees and Support Vector Clustering. Furthermore, the application of techniques from other fields, such as Expectation Maximization and Renyi entropy has also been reported [45]–[51].

Considering all that has been presented and discussed throughout the current chapter, there still remain questions to be further investigated and answered.

Regarding the use of context information within the scope of Self-Adaptive, C-A CPS, determining which context information is relevant to the current scenario observed in the system is not a trivial task, nor is adjusting the operation of a system according to perceived and inferred context information. As supported by Bauer and Dey, "anticipating relevant context - ahead of the actual situation being reality - is a key challenge" [52].

IV. Adaptive Supervisory Framework

The current section is reserved for presenting and discussing the adaptive supervisory framework intended to be developed as the main outcome of our work.

A. Research Problem

The main expected outcome of this work is to develop an Adaptive Supervisory Framework for Cyber-Physical Systems. The primary focus of such a framework is to continuously monitor and optimise a given system, usually large-scale and with a distributed nature, composed of several heterogeneous and interacting subsystems and entities.

Building on the topics discussed so far in this document, concepts such as context-awareness and self-adaptiveness are perfectly framed within this scenario. Insights from these fields of study are expected to contribute towards a better and more adequate adaptation to its target systems and surrounding environment. We can describe our research problem as:

In modern cyber-physical systems different (physical and cyber) subsystems and human entities are becoming more and more interconnected, continuously challenging their monitoring and overall optimisation. Thus, proper context-aware and self-adaptive mechanisms must be explored in order to achieve an optimised management of current cyber-physical systems, supported by context information.

B. Framework Architecture

The subject of attention and study of the current work are CPS, namely large-scale CPS composed of several interconnected subsystems and entities. Reinforcing what has already been referred in previous sections, each of these composite subsystems can have its own operational mode, objectives and constraints, which need to be properly integrated and optimised at macro-level.

In this sense, it is important to perform an adequate monitoring of the different subsystems and entities being managed, so that a global optimised management of the entire CPS can be achieved.

Given CPS' distributed nature, this subsystems' monitoring must be performed in a distributed manner, being responsible for collecting context information in physically distant areas, where the considered subsystems are located. Such mechanisms must also enable the sharing of context information collected in the different sources.

In order to achieve a global optimisation of the entire CPS, local optimisations of existing subsystems must be performed, triggered based on perceived and inferred context information at both local and global levels. To this end, a central entity capable of analysing and processing information at global scale is required.

To this end a framework with adaptive supervision capabilities is intended to be developed, and is currently being projected. As a starting point, the combination of principles from the DYNAMICO reference model and the hierarchical architectural pattern of the MAPE-K loop is being studied, seeking to develop and propose a preliminary architecture for this framework.

As is common with hierarchical control structures, entities at different hierarchical levels must be considered, with entities at higher levels responsible for monitoring and controlling the operation of entities at lower levels. A higher-level entity can control the operation of a lower-level entity by, for example, assigning to it certain operational objectives.

Perceived changes in the systems being monitored and controlled, as well as in the environment in which they are inserted, can also condition the operation of the CPS at a local and/or global level. Therefore, proper monitoring mechanisms must be implemented at lower-level entities. In addition, the framework to be proposed must allow the CPS to evaluate its own behaviour in real-time and reconfigure its components (subsystems) whenever their operational objectives and constraints are no longer satisfied.

V. Conclusion

The current document is framed within the PhD work developed by *Joaquim Leitão* in the *Doctoral Program in Information Science and Technology*, introducing the research work to be developed by this candidate. The proposal for this research work was developed based on previous research works conducted by the candidate, which highlighted the need to develop intelligent systems in charge of supervising other systems and environments in an autonomous way.

Following this line of thought, in this research work an Adpative Supervisory Framework for Cyber-Physical Systems is intended to be developed, which implements supervision and optimisation mechanisms for systems and entities operating under the IoT and CPS paradigms. Application case studies will also be considered for validation of this Framework.

The proposed Framework will integrate a novel generation of digital systems, in which the cyber and physical worlds are bridged, closing gaps between these two (distinct) realities. An autonomous management of a given environment is sought, being challenged by the characteristics of the systems and entities that often compose the environment in question.

Indeed, from the literature review, systems that share characteristics and properties with the Framework proposed in this document can be identified. Distinct reference models that aim to guide the development of systems of this nature have been identified. Among their most important and relevant characteristics, the ability to work with uncertain information and the identification of recurrent patterns of behaviour are highlighted.

One model that clearly stands out is the MAPE-K loop, proposed by IBM researchers, being the basis and support of other reference models, such as the DYNAMICO model. Inspired by the feedback-loop model, extensively used in control theory, the MAPE-K defines a model for the development of self-adaptive systems by means of a loop with the following steps: continuously monitoring of a given environment, analysis of its state, planning of any interventions and execution of planned actions.

Taking the study and analysis of the available literature, contributions towards the development of an architecture for the desired Adaptive Supervisory Framework were performed, supported by principles adopted in the MAPE-K loop and the DYNAMICO reference models.

References

[1] T. Sanislav and L. Miclea, "Cyber-physical systems-concept, challenges and research areas," *Journal of Control Engineering and Applied Informatics*, vol. 14, no. 2, pp. 28–33, 2012.

[2] D. S. Nunes, P. Zhang, and J. S. Silva, "A survey on human-in-the-loop applications towards an internet of all," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 944–965, 2015.
[3] J. Leitão, "Flood management in urban drainage: Contributions for the control of water drainage systems using underground barriers," Master's thesis, University of Coimbra, Portugal, 2016.
[4] J. Leitão, A. Cardoso, J. A. Marques, and N. Simões, "Flood management in urban drainage: Contributions for the control of water drainage systems using underground barriers," in *Experiment@ International Conference (exp. at'17), 2017 4th*, pp. 286–291, IEEE, 2017.
[5] J. Leitão, *Adaptive Supervisory Framework for Cyber-Physical Systems*. PhD thesis proposal, Department of Informatics Engineering, University of Coimbra, September 2017.
[6] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th Design Automation Conference*, pp. 731–736, ACM, 2010.
[7] K.-D. Kim and P. R. Kumar, "Cyber–physical systems: A perspective at the centennial," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.
[8] R. H. Weber, "Internet of things–new security and privacy challenges," *Computer law & security review*, vol. 26, no. 1, pp. 23–30, 2010.
[9] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
[10] R. Van Kranenburg, *The Internet of Things: A critique of ambient technology and the all-seeing network of RFID*. Institute of Network Cultures, 2008.
[11] I. T. Union, "Overview of the internet of things," 2012.
[12] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
[13] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, "Cyber physical system challenges for human-in-the-loop control," in *Presented as part of the 8th International Workshop on Feedback Computing*, (San Jose, CA), USENIX, 2013.
[14] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85–90, IEEE, 1994.
[15] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, *Towards a Better Understanding of Context and Context-Awareness*, pp. 304–307. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.
[16] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: On networking and computing environments' integration," *IEEE communications surveys & tutorials*, vol. 15, no. 1, pp. 362–386, 2013.
[17] L. Barkhuus and A. Dey, "Is context-aware computing taking control away from the user? three levels of interactivity examined," in *International Conference on Ubiquitous Computing*, pp. 149–156, Springer, 2003.
[18] U. Alegre, J. C. Augusto, and T. Clark, "Engineering context-aware systems and applications: A survey," *Journal of Systems and Software*, vol. 117, pp. 55–83, 2016.
[19] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pp. 92–99, IEEE, 1998.
[20] J.-y. Hong, E.-h. Suh, and S.-J. Kim, "Context-aware systems: A literature review and classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.
[21] H. Chen, *An intelligent broker architecture for pervasive context-aware systems*. PhD thesis, University of Maryland, Baltimore County, 2004.
[22] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*, pp. 217–253, Springer, 2011.
[23] P. Oreizy, M. M. Gorlick, R. N. Taylor, D. Heimhigner, G. Johnson, N. Medvidovic, A. Quilici, D. S. Rosenblum, and A. L. Wolf, "An architecture-based approach to self-adaptive software," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 3, pp. 54–62, 1999.
[24] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08. IEEE International Conference on*, pp. 1–9, IEEE, 2008.
[25] G. Tamura, *QoS-CARE: A Reliable System for Preserving QoS Contracts through Dynamic Reconfiguration*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I; Universidad de Los Andes, 2012.
[26] N. M. Villegas, "Context management and self-adaptivity for situation-aware smart software systems," *University of Victoria*, 2013.
[27] D. Weyns, B. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, and K. M. Göschka, "On patterns for decentralized control in self-adaptive systems," in *Software Engineering for Self-Adaptive Systems II*, pp. 76–107, Springer, 2013.
[28] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.
[29] P. AMADEOS, "Deliverable d3.2 "amadeos methodology, mechanisms, and implementations"," 2016.
[30] N. M. Villegas, G. Tamura, H. A. Müller, L. Duchien, and R. Casallas, "Dynamico: A reference model for governing control objectives and context relevance in self-adaptive software systems," in *Software Engineering for Self-Adaptive Systems II*, pp. 265–293, Springer, 2013.
[31] H. Kopetz, O. Höftberger, B. Frömel, F. Brancati, and A. Bondavalli, "Towards an understanding of emergence in systems-of-systems," in *System of Systems Engineering Conference (SoSE), 2015 10th*, pp. 214–219, IEEE, 2015.
[32] H. Kopetz, "Simplification principles in the design of cyber-physical system-of-systems," in *Complex Systems Design & Management*, pp. 39–51, Springer, 2016.
[33] J. C. Mogul, "Emergent (mis)behavior vs. complex software systems," in *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, EuroSys '06, (New York, NY, USA), pp. 293–304, ACM, 2006.
[34] Y. Sardahi, *Multi-objective optimal design of control systems*. PhD thesis, University of California, Merced, 2016.
[35] A. Guillemin and N. Morel, "An innovative lighting controller integrated in a self-adaptive building control system," *Energy and buildings*, vol. 33, no. 5, pp. 477–487, 2001.
[36] H. Zhang, A. Davigny, F. Colas, Y. Poste, and B. Robyns, "Fuzzy logic based energy management strategy for commercial buildings integrating photovoltaic and storage systems," *Energy and Buildings*, vol. 54, pp. 196–206, 2012.
[37] R. T. Rockafellar, "Lecture notes in optimization under uncertainty," 2001.
[38] U. Diwekar, *Introduction to applied optimization*, vol. 22. Springer Science & Business Media, 2008.
[39] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
[40] G. J. Tsekouras, N. D. Hatziargyriou, and E. N. Dialynas, "Two-stage pattern recognition of load curves for classification of electricity customers," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1120–1128, 2007.
[41] J. M. Abreu, F. C. Pereira, and P. Ferrão, "Using pattern recognition to identify habitual behavior in residential electricity consumption," *Energy and buildings*, vol. 49, pp. 479–487, 2012.
[42] G. Chicco, "Overview and performance assessment of the clustering methods for electrical load pattern grouping," *Energy*, vol. 42, no. 1, pp. 68–80, 2012.
[43] J. Kwac, J. Flora, and R. Rajagopal, "Household energy consumption segmentation using hourly data," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 420–430, 2014.
[44] I. Panapakidis, M. Alexiadis, and G. Papagiannis, "Evaluation of the performance of clustering algorithms for a high voltage industrial consumer," *Engineering Applications of Artificial Intelligence*, vol. 38, pp. 1–13, 2015.
[45] F. Rodrigues, J. Duarte, V. Figueiredo, Z. Vale, and M. Cordeiro, "A comparative analysis of clustering algorithms applied to load profiling," *Machine Learning and Data Mining in Pattern Recognition*, pp. 73–85, 2003.
[46] G. Chicco, R. Napoli, F. Piglione, P. Postolache, M. Scutariu, and C. Toader, "Load pattern-based classification of electricity customers," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1232–1239, 2004.
[47] G. Chicco and I.-S. Ilie, "Support vector clustering of electrical load pattern data," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1619–1628, 2009.
[48] Z. Yu, F. Haghighat, B. C. Fung, and H. Yoshino, "A decision tree method for building energy demand modeling," *Energy and Buildings*, vol. 42, no. 10, pp. 1637–1646, 2010.
[49] S. Bidoki, N. Mahmoudi-Kohan, M. Sadreddini, M. Z. Jahromi, and M. Moghaddam, "Evaluating different clustering techniques for electricity customer classification," in *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, pp. 1–5, IEEE, 2010.
[50] G. Chicco and J. S. Akilimali, "Renyi entropy-based classification of daily electrical load patterns," *IET generation, transmission & distribution*, vol. 4, no. 6, pp. 736–745, 2010.

[51] Z. Wang, S. Bian, Y. Liu, and Z. Liu, "The load characteristics classification and synthesis of substations in large area power grid," *International Journal of Electrical Power & Energy Systems*, vol. 48, pp. 71–82, 2013.

[52] C. Bauer and A. K. Dey, "Considering context in the design of intelligent systems: Current practices and suggestions for improvement," *Journal of Systems and Software*, vol. 112, pp. 26–47, 2016.