IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received June 16, 2020, accepted July 3, 2020, date of publication July 8, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007955

An Intelligent IoT-Based System Design for Controlling and Monitoring Greenhouse Temperature

AHMAD F. SUBAHI[1] **AND KHEIR EDDINE BOUAZZA**[1,2]
[1]Department of Computer Science, University College of Al Jamoum, Umm Al Qura University, Makkah 21421, Saudi Arabia
[2]Laboratoire d'Informatique et des Technologies de l'Information d'Oran (LITIO), University of Oran, Oran 31000, Algeria

Corresponding author: Ahmad F. Subahi (afsubahi@uqu.edu.sa)

This work was supported by the King Abdulaziz City for Science and Technology, Saudi Arabia, under Project 13-ENE2356-10.

ABSTRACT The Kingdom of Saudi Arabia is known for its extreme climate where temperatures can exceed 50 °C, especially in summer. Improving agricultural production can only be achieved using innovative environmentally suitable solutions and modern agricultural technologies. Using Internet of Things (IoT) technologies in greenhouse farming allows reduction of the immediate impact of external climatic conditions. In this paper, a highly scalable intelligent system controlling, and monitoring greenhouse temperature using IoT technologies is introduced. The first objective of this system is to monitor the greenhouse environment and control the internal temperature to reduce consumed energy while maintaining good conditions that improve productivity. A Petri Nets (PN) model is used to achieve both monitoring of the greenhouse environment and generating the suitable reference temperature which is sent later to a temperature regulation block. The second objective is to provide an Energy-Efficient (EE) scalable system design that handles massive amounts of IoT big data captured from sensors using a dynamic graph data model to be used for future analysis and prediction of production, crop growth rate, energy consumption and other related issues. The design tries to organize various possible unstructured formats of raw data, collected from different kinds of IoT devices, unified and technology-independent fashion using the benefit of model transformations and model-driven architecture to transform data in structured form.

INDEX TERMS Intelligent greenhouse agriculture, temperature control system, the Internet of Things (IoT), Petri nets (PNs), graph database, model transformations, model-driven architecture (MDA).

I. INTRODUCTION

Agriculture in the Kingdom of Saudi Arabia (KSA) faces several constraints, including extreme temperatures, water scarcity, sea water desalination costs, and non-fertile soil. To overcome this hostile environment and ensure agricultural self-sufficiency, multiple government agricultural programs were launched to ensure food security [1]. Indeed, agricultural self-sufficiency is a sign of a country's stability and strength [2]. Agricultural self-sufficiency can only be achieved by introducing innovative environmentally suitable solutions and modern agricultural technologies necessary for improving productivity and decreasing production costs. Greenhouse farming is interesting in the sense that it succeeds in isolating the yield of nature, and allowing the protection of plants against the immediate impact of external climatic conditions [3].

In the desert climate where the summer lasts over half of the year, as in Saudi Arabia. The average temperature in July is around 43 °C, and the average one in January is about 14 °C. It is impossible to enhance the production of vegetables and fruits like tomatoes, Cucumbers, Sweet peppers and strawberries, as the optimum temperature for their growth falls in the range between 11°C to 28°C.

From that, it can be realized that there is a necessity of providing an appropriate controlled microclimate, for various kinds of crops, that requires caring of four main environmental parameters, namely, temperature, humidity, CO_2 level and light intensity. This makes greenhouses is an appropriate economical solution for farming because climate variables can be manipulated and controlled to achieve optimal growth rate of crops. Greenhouse allows producing crops, especially

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

fruits and vegetables production that requires cold weather to grow fast in a quality manner, all year round and meet consumer demand for out-of-season fruit and vegetables [4].

The Internet of Things (IoT) concept allows the system, using electronic circuits, sensors and programming, to detect and control other devices remotely, creating a good interaction between the physical and computer world in order to improve efficiency and accuracy while achieving financial benefits [5]. The use of IoT in the development of smart homes has received increasing interest. The studies presented focus mainly on energy management through the control of electrical units [6]–[10]. The results are very interesting, and are being used more and more in everyday life.

The use of IoT in greenhouse agriculture contributes to its development [11]. Thus, the information collected from the sensors, inside and outside of the greenhouse, can be analyzed and stored on a central cloud data storage for archiving long term analysis and data mining tasks, as well as stored on cloud edge points for faster processing. End-users can access these data from any active internet device and gain benefits of the generated knowledge regarding their greenhouse crops production, energy consumption, and other related issues associated to this business [12]–[14]. The use of IoT in greenhouses is receiving increasing attention and many interesting results have been achieved [15]–[17].

In this paper, based on the IoT new advances in using sensors equipment, we propose to build an intelligent Energy-Efficient (EE) system which monitors and controls internal greenhouse temperature. The proposed system will allow increased and improved productivity. The main study objective is not only to build a consistent growing environment, but also to automate the whole system and make it smart to save energy and production costs. The proposed approach focuses on monitoring and controlling greenhouse internal temperature, but it can be extended to other kinds of properties, e.g. Carbon dioxide (CO_2) and humidity.

The proposed system is considered as smart because it is able, autonomously, to monitor the outside temperature and the energy consumption rush hours, in order to accurately generate the suitable reference temperature, and ensure that the greenhouse temperature reaches this reference temperature. In addition, this system can identify the angle of the Sun rays in order to control the opening and closing of the awnings, which results in reducing the effects of high-temperatures.

All these captured parameters related to temperature and energy are recorded for future analysis and prediction in a dynamic graph data model used in designing the backend storage of the system. The proposed system design supports handling IoT data in a unified and technology-independent fashion, using a suitable strategy of model transformations, a principle of model-driven engineering methodology of software system development. The designed graph-based schema accepts multiple formats of IoT data and parameters that come from various sensor brands.

The rest of this paper is organized as follows. Section II represents a brief highlight about the related technologies to the proposed work, including the concepts of big data, graph data model, principles of model transformations and model-driven architecture (MDA). Section III, discusses in depth the overall architectural design of the proposed greenhouse system, including its core components: temperature control & monitoring subsystem with simulation results, data conversion subsystem, and greenhouse management information system. Section IV, presents the contribution summary of the proposed greenhouse system. Finally, in the conclusion (section V), we summarize the ideas and innovations presented in this paper and outline some future perspectives and applications.

II. RELATED TECHNOLOGIES

A. IoT SENSORS

A broad variety of sensors that provides a mature sensing technology for greenhouse monitoring applications is available in markets These sensors can be utilized for collecting, automatically, important information including microclimate data in the greenhouse, control actions, crops growth rate and characteristics of the crops and more related data. The availability of these various kinds of advanced sensing technologies in the markets makes the implementation of the proposed IoT-based system is achievable.

Temperature sensors, for instance, can be used to measure, periodically the temperature inside the greenhouse. There are some common models utilized for this purpose, such as LP PYRA 02 [4], E + E Elektronik EE160 [18], DHT11 [15], [19], and LM35 model [3]. On the other hand, Humidity sensors can be utilized for sensing the amount of vapours in the air, such as LP PYRA 02 [4], E + E Elektronik EE160 [18], DHT11 [15], [19], and HSM20G [3]. Additionally, there is another important sensing technology need to be utilized inside the greenhouse for measuring the level of CO_2. It is available in markets with different models such as Vaisala GMP220 [4]. Sensors for sensing the density of Sun rays is also another kind of critical and sensitive technology required in the greenhouse visible, such as LP PYRA 02 [4], and Apogee Instruments Inc. SP110 [18].

B. ENERGY EFFICIENCY (EE)

Energy efficiency (EE) has always been a research hotspot in the field of Internet of Things [20], [21]. According to [20], for instance, an integrated structure is introduced for both wireless and wired parts to optimize the energy efficiency (EE) performance of the fifth generation (5G) Internet of Things systems. A cellular partition zooming (CPZ) mechanism and a precaching mechanism were utilized for the wireless part and the wired one respectively. The integration of both mechanism, in the proposed comprehensive solution (structure), provided better deployment of the select-and-sleep mechanism in the introduced component of the unified control center. In order to cover wider outdoor area, the single antenna RRHs was replaced by massive MIMO array instead. As a consequence, This solution was regarded economical as it used a fewer number of RRHs. Additionally, it was considered energy-efficient as it can satisfy transmission requests using the information stored in the router directly in less distance [20]. Besides, IoT devices for agriculture and aquaponics that are based on nRF52840 microcontroller and Bluetooth 5 have been introduced and tested, in [21], for energy harvesting. It showed nodes ability to harvest energy several times, during their lifetime, more than their amount of energy consumption. Thus there is no maintenance needs for these nodes.

C. BIG DATA

Nowadays, with rapid technology advancement in the information era, data has become available and accessible everywhere as it is collected in massive quantities with different formats and from various sources in an unstructured or semi-structured way. This has led to the domain of big data to emerge and receive great attention from computer, data scientists and software system architects.

With respect to IoT applications, vast amounts of data is periodically captured and collected from a wide variety of sensor devices, which is considered one common source of big data in general. These data are captured with different formats such as texts, images, documents, sounds, videos, Global Positioning System (GPS) points, routes information and more. At the backend of the system, these various types of data must be indexed somehow, stored, and linked, before being analyzed, processed, and retrieved.

These operations and the other basic CRUD (create, read, update and delete) ones brought different challenges in the domain of big data. According to [22], traditional relation-based data models do not properly serve the needs of storing and processing terabytes or even petabytes of big data. The data model of big data has special characteristics that are not fully supported by relational data models, namely: Volume, Velocity and Variety. NoSQL data model brings key benefits to tackle these issues, such as efficiency, scalability, and availability when storing and processing large amounts of data. By the time, the role of NoSQL databases has emerged in solving many Big Data challenges.

Graph databases are found based on graph theory and can be defined as a form of data store representation that adopts graph structures in expressing data entities, properties, interrelationships and semantic queries. It has been used in recent decade, as a formal technique, for modeling various kinds of distributed and interconnected systems that work with massive amounts of data, such as social networks, cloud-based systems, biological systems and more. It is considered an example of NoSQL databases that has a strong ability to express and manage connected data and their relationships in IoT systems [23], [24]. There are various graph database systems used in the domain of big data and IoT applications, such as Neo4j, TITAN, and OrientDB.

One of the main characteristics of distributed systems is the demand to distribute data storage across different locations, or network nodes. Each part of the overall system is responsible for generating, storing, and manipulating various kinds of independent data. Graph representation is an effective and light way to express and model the distributed data because of its nature. In contrast to traditional relational database systems, the efficient performance of the graph data model outweighs the robustness benefits of the relational one. As such the performance of the relational models might be cost when the number of connected data are increased via JOIN operators. Thus, in the proposed work, the graph data model is considered as a design choice to model the required data for the proposed greenhouse smart system.

Based on the graph theory, the graph can be navigated and traversed through different traversal techniques. In graph database this requires adopting appropriate query language to traverse graph paths and brings answers [25]. Regarding the Neo4j, the suggested design choice, the Cypher query language is used for querying Neo4j graph databases. Cypher has a descriptive, human-readable, textual notation for expressing user queries [25], [26]. Similar to other query languages, Cypher has various widely used operations, namely, filters, aggregate functions, create, delete and match. Going into the syntactical detail about Cypher is out of the scope of goals of this work.

D. SECURITY

Information security is another challenge and a research hotspot in the field of Internet of Things (IoTs). With the recent advancement in technology, blockchain has emerged as one of the promising secure technologies for IoT applications [27]–[30].

The work presented in [27] starts by introducing the basic concept of blockchain and illustrating why a consensus mechanism along with an encryption algorithm plays an indispensable role in a blockchain enabled IoT system strengthening its overall security by design. Three main mechanisms were included in the detailed comparison between their characteristics, advantages and limitations for IoT systems in [27], namely, Proof of Work (PoW), Proof of Stake (PoS) and Direct Acyclic Graph (DAG) based consensus mechanisms (Tangle and Hashgraph approaches).

In addition to this, there are several blockchain-based frameworks introduced to store and audit access control policies. For instance, Sash, a secure and decentralized IoT blockchain-based data-sharing framework for auditing access control policy was introduced in [27]. It is considered secured-by-design that provides two methods for granting access policies via two sharing schemas. The first method is by distribution of prefix decryption Keys, whereas the second one is based on Access Control List (ACL) that contains a list of permissions to manage who can access a data [27].

E. PRINCIPLE OF MODEL TRANSFORMATIONS

Automatic model transformation (MT) is considered a key principle in advanced model-oriented software development methodologies, such as Model-driven Engineering (MDA). Models, as abstracted views of the system, are considered

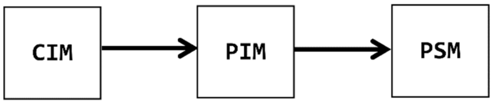

FIGURE 1. Steps of transformations.

primary artefacts used throughout the development lifecycle instead of code. These models express the system from different views and at multiple levels of abstraction to enable the separation of concerns based on developers perspectives. The MDA-based approaches are based on utilizing three kinds of core models, namely, Computation Independent Model (CIM), Platform-Independent Model (PIM), and Platform Specific Model (PSM) [31].

The CIM contains all crucial details about the problem domain, expressed by domain experts or business analysts, without any information system implementation detail. CIM may include business processes and desires of end-users. Besides, the PIM model describes the structure and behavior of a system abstractly without containing concrete technical details about the final system. It is expressed textually or graphically using a general purposes modeling language, such as the Unified Modelling Language (UML), or a suitable domain specific modeling language (DSML). On the other hand, the PSM model describes the structure and behavior of the target system for a particular environment and technology [32], [33].

Some MDA applications focus on deriving one or more target PSMs, associated to implementation detail of a target environment, and generating executable database schema from a comprehensive PIM or conceptual data model. The PIM is translated, by means of executing an appropriate strategy of model transformations, into PSMs and the target schema or code. This work introduces an automatic transformational approach for (1) transforming a platform independent IoT-oriented graph, which contains raw data captured from sensors, into an optimized and simplified physical Neo4j graph data model, (2) generating an executable Neo4j Cypher script from the physical model.

III. SYSTEM DESIGN AND ARCHITECTURE

Basically, it is assumed that the proposed greenhouse system for controlling and monitoring temperature consists of three main subsystems, namely, temperature control & monitoring subsystem, greenhouse management information system, and data conversion subsystem, rather than starting the design description using collaborated classes and responsibilities. This strategy simplifies the complexity of intercommunications and collaborations between the system units. The following UML component diagram (Figure 2) is used to demonstrate the overall architectural design of the proposed system via a number of abstracted units (subsystems).

A. TEMPERATURE CONTROL & MONITORING SUBSYSTEM

In the KSA, maintaining a suitable temperature within a greenhouse environment is very important. Temperature changes can damage plants in only a few hours, and consequently decrease productivity. Keeping healthy plants requires the best possible growing environment. Controllers are the heart in a greenhouse system where different sensors and actuators are connected to build an automated system.

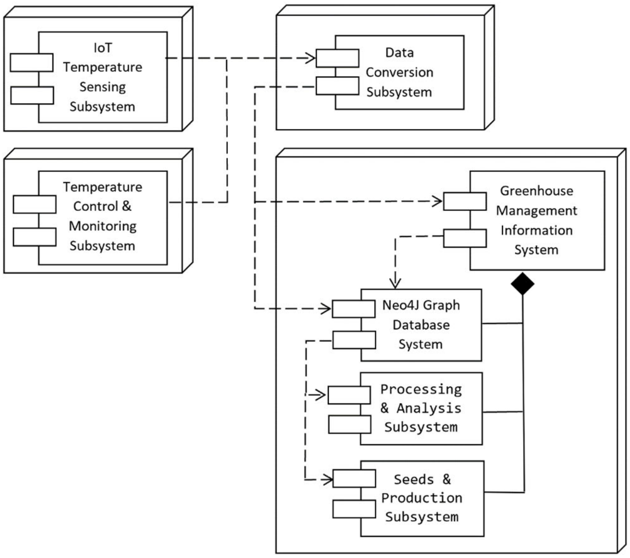

FIGURE 2. Architectural design of the greenhouse system.

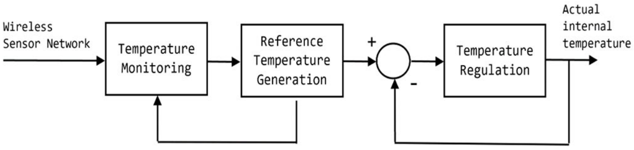

FIGURE 3. Control system scheme.

To reach the objectives of saving energy and production costs, two different schemes are used at the design level. The first scheme is based on the smart use of traditional feedback closed-loop systems to regulate the greenhouse temperature. The second scheme, for controlling greenhouse lighting, uses a Petri Nets (PN) model to control the smart spread of shade cloth awnings according to the angle of the Sun rays. In Figure 3, the proposed system consists of three stages, each one is responsible for a particular task.

The first stage is designed to monitor the external temperature and energy consumption rush hours. A wireless sensor network continually measures temperature. As soon as there are changes in the outside temperature this stage should inform the next stage to take the appropriates tasks. Temperature measurements are stored in the backend graph database. The stored data can be very useful in analyzing system properties and in predicting future values of properties via the data processing and analysis subsystem, which is another core component of the greenhouse system.

In addition to monitoring outside temperature, the proposed PN will also monitor energy consumption in the national power grid to take appropriate action to reduce energy consumption. Indeed, the PN model will send a moderate temperature a little bit higher that the ideal one during the energy consumption rush hours to reduce the greenhouse consumption and then to participate in reducing energy consumption pressure on the kingdom power grid.

After the end of the energy consumption rush hour the smart system can return back to the ideal temperature. The higher temperature sent during energy consumption rush hours should not have any negative effect on the plants inside the greenhouse. In the second stage, based on the information obtained from the first stage, the necessary temperature reference signal is generated and sent to the next stage. The reference signal transmission lasts for a period chosen by the system manager. At the end of this period the system will go back to the first stage to check if there are any changes in the environmental proprieties that require changes in the reference signals. It is worth mentioning that the introduced control system design can be work the recent energy-saving technologies. As the targeted implementation platform, which is out of the scope of this paper, is based on 5G wireless communication technology with MIMO strategy in transmitting and receiving captured IoT big data.

The third stage is dedicated to the system control and regulation. Based on the reference signals sent in its input, this stage will regulate the greenhouse internal temperature. Because of its recognized proprieties such as power of modeling and implementation, possible real-time implementation, and supervision qualities, we propose to use Petri Networks (PNs) in the first and second stage. Indeed, using a PN model, the user has the ability to monitor the system and to take appropriate actions when needed.

The main role of the third stage is to ensure that the actual greenhouse temperature reaches the temperature reference signal sent by the previous stage. For that, a traditional feedback control system is used. Each user can use, at this stage, the controller he deems most appropriate.

The innovation of the control part of the presented work is to have associated different processes; monitoring and supervising the greenhouse through PN, reference temperature generation, and temperature regulation using the PID controller for obtaining an intelligent system that reduces energy consumption while maintaining an adequate temperature to grow the plants in good conditions.

Indeed, associating a Petri Nets Model, responsible for "listening" to the outside world (first block of Figure 3), generating an adequate reference temperature signal (second block of Figure 3), and then ordering the PID controller to reach a reference temperature (third block of Figure 3), to ensure that the whole control system is smart with a high degree of interaction with the outside world.

All these tools, used in the proposed system, are well-known and have given good results each in its corresponding field. We considered the advantages of each tool when using it in the control system. Therefore, the proposed system integrates all the advantages of these components: feasibility, ease of implementation, and great flexibility.

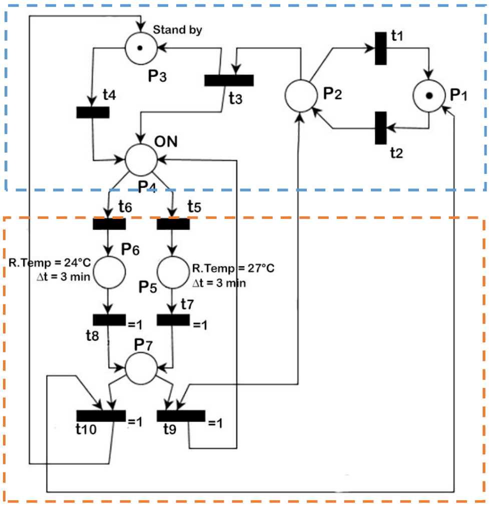

FIGURE 4. Temperature PN supervisor.

1) GREENHOUSE MONITORING AND REFERENCE SIGNAL GENERATION

The supervisor PN that monitors the greenhouse system and generates the suitable reference temperature is shown in Figure 4. As we can see on the figure, the supervisor PN is divided into master PN (represented in blue) and slave PN (represented in red).

a: MASTER PN

The master PN is used to identify the system state. It is composed of four states. The Standby mode (P3) orders the system to stop generating and send a reference temperature, but still monitor the systems. It happens when the outside temperature is between 17°C and 30°C. The second system mode is the ON mode (P4). This mode is activated when the outside temperature is below 17°C or above 30°C. P1 and P2 are two states representing the outside temperature.

b: SLAVE PN

The slave PN works only when the master PN is in On mode. Depending on daytime, the slave PN sends the suitable references temperatures. Indeed, if it is the energy consumption rush hour time, generally between 11:00 and 17:00 [34], then the PN model to reduce energy consumption will send a relatively high reference temperature equal to 27°C. Apart from energy consumption rush hour, a more adequate reference temperature equal to 24°C is send to the temperature regulation stage. The reference temperature sending stage can still be activated for a duration chosen by the system manager. In this case study the PN model keeps this state activated for 3 minutes.

At the end of this duration, two situations are possible. Either the outside temperature is still above 30°C or below 17°C, and a new cycle of identifying daytime and sending reference temperature is accomplished. Or the outside temperature is between 17°C and 30°C and then the system goes back to the standby mode.

TABLE 1. Temperature control PN places and transitions.

Places		Transitions	
P1	17 C <= Temp <= 30 C	t1	17C < Temp < 30C change detection
P2	Temp < 17 or Temp > 30	t2	Temp < 17 or Temp > 30 change detection
P3	Standby mode	t3	Working status changes from Standby to ON
P4	On mode	t4	Working status changes from ON to Standby
P5	Sending ref temp = 27 C. to stage3 during t = 3 min.	t5	Energy consumption rush hours detection
P6	Sending ref temp = 24 C. to stage3 during t = 3 min.	t6	Not energy consumption rush hours detection
P7	Intermediary state	t7, t8, t9, t10	= 1

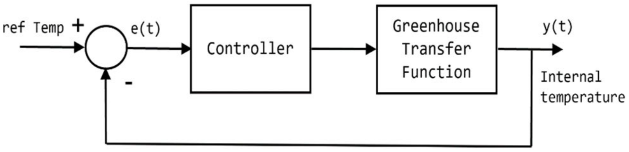

FIGURE 5. Greenhouse closed loop temperature regulation system.

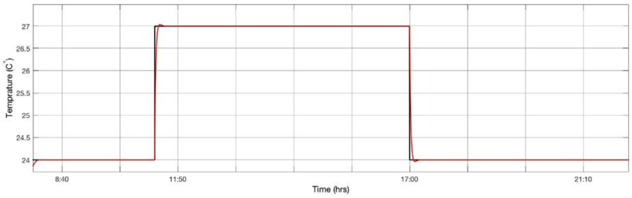

FIGURE 6. Greenhouse temperature response with PID controller.

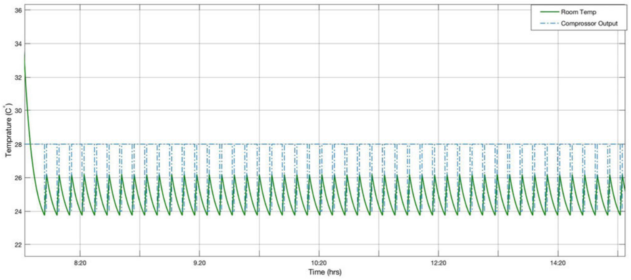

FIGURE 7. Greenhouse temperature response with PID controller.

2) TEMPERATURE REGULATION

In this section, some of the main control algorithms used in greenhouse control literature are cited. The closed loop system for the greenhouse temperature control is represented in Figure 5.

A significant number of control algorithms have been used for temperature control inside a greenhouse. We can cite basic controls such as Proportional-Integral-Derive (PID) control, controls based on fuzzy logic or neural networks, robust controls, non-linear type controls, as well as hybrid controls [35]. Overall, these algorithms could be classified into three types of algorithms, conventional controls, advanced controls, and finally intelligent controls [36]–[39].

Because of its high-quality performance, simplicity and flexibility, the Proportional-Integral-Derivative (PID) control is one of the most used in greenhouse system control [40]. The PID controller starts by reading sensor values, then it computes the corresponding actuator output by adjusting proportional, integral, and derivative factors [41]. Adjusting PID controller factors can be achieved using different approaches [42].

More advanced controllers that require accurate mathematic was also applied to control greenhouse systems. Model Predictive Control was largely used in controlling greenhouse system proprieties, we refer the reader to [43]–[46]. Feedback and feedforward control was also applied to control greenhouse systems [47]–[49]. To handle unknown model proprieties, some authors used adaptive control such as in [50]–[53]. In the presence of system uncertainties and disturbances, robust control is the most adapted. It was also largely used and gave interesting results [54]–[57].

Depending on the budget that can be invested, for the main desired objective and the used model, the greenhouse manager can choose one of the control algorithms mentioned above.

3) SIMULATION RESULTS

In order to show the effectiveness of the proposed approach in this work, a number of simulations are performed using a greenhouse temperature transfer function proposed by authors in [58].

$$G(s) = \frac{1.03 \times 10^{-4}}{s^2 + 0.015s + (2.78 \times 10^{-5})}$$

The greenhouse is supervised using the PN model developed in Figure 4. The regulation model, applied in stage 3 of the proposed system, is implemented using a PID controller according to the model represented in Figure 5.

The reference temperature signal sent for regulation is typical for a summer day (From May to September). The reference temperature is equal to 27°C from 11:00 to 17:00 (energy consumption rush hour). Apart from energy consumption rush hour, from 8:00 am to 11:00 am and 17:00 pm to 22:00 pm, a reference temperature equal to 24°C is sent to the regulation block. The greenhouse temperature simulation using the proposed smart system is shown in Figure 6 below. Furthermore, Figure 7 demonstrates the temperature in the greenhouse when the On/Off controller is applied around a temperature of 24°C.

Additionally, Figure 8 shows the energy consumed when the On/Off control was applied, while Figure 9 illustrates the energy consumed after applying the PID controller to the smart system.

The comparison between the energy consumed by the traditional temperature control system, based on a reference temperature signal of On/Off type, is shown in Figure 8.

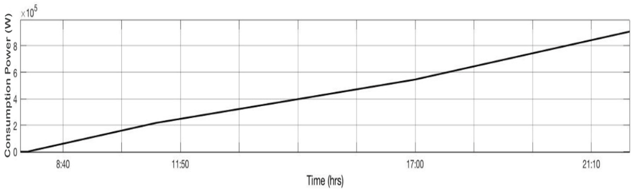

FIGURE 8. The power consumption of ON-OFF controller.

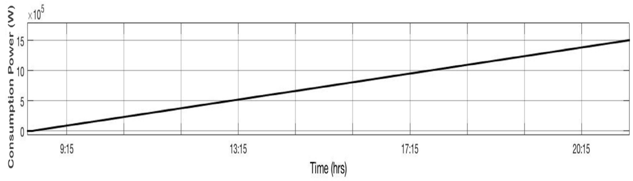

FIGURE 9. The power consumption after applying PID controller.

The proposed smart system, illustrated in Figure 9, shows a significant reduction in the consumed energy by the latter compared to the first one. We can notice on the simulations that using the proposed smart system, the reduction in energy consumption is approximately of **38.45%** compared to the On/Off controller. Indeed, when using the On/Off controller to maintain the temperature around a predefined temperature of 24 °C for 14 hours, from 8:00 am to 10:00 pm, the total consumed energy was **1514.76 kW** as it demonstrated in Figure 8. This is remarkably declined to **932.36 kW** when using the proposed smart system, as it shown in Figure 9.

4) LIGHTING: AWNINGS

The proposed greenhouse system is designed to be used in KSA. A region of the world known for its extreme climate. In fact, temperatures can exceed 50 °C, especially in summer [28]. In addition, full sunshine periods, generally, last a large part of the day. Thus, temperature control in greenhouses in KSA is very difficult especially during the part of the day when the Sun rays are at their maximum. It is therefore important to try to reduce the period of sunshine in order to decrease temperature and to not dry out the plants. In the second scheme of the proposed system, controlled awnings, thick enough to reduce and soften the Sun rays are a solution that achieves this objective. Indeed, they help to reduce the energy consumed, because they reduce the temperature inside the greenhouse and consequently reduce the energy necessary for the regulation of the temperature. It is very important that the awnings are made of a material resistant to the sun, wind, and dust.

The awning system opens and closes automatically according to the angle of the Sun rays. The more the Sun rises, the more the awnings open, until they open completely when the Sun rays are perpendicular to the earth (Figure 10). The amount of sunshine received during the rest of the day is enough to give plants the sunlight they need to create their food. Thus, we use the following PN model to control the opening and closing of the awnings (Figure 11).

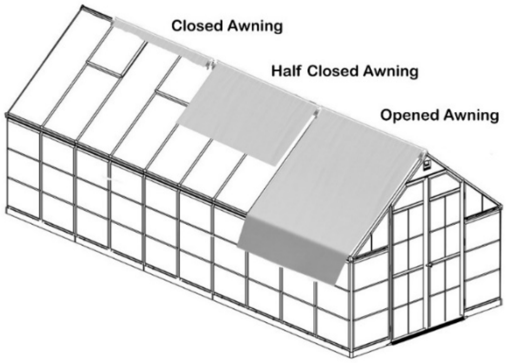

FIGURE 10. Greenhouse awnings.

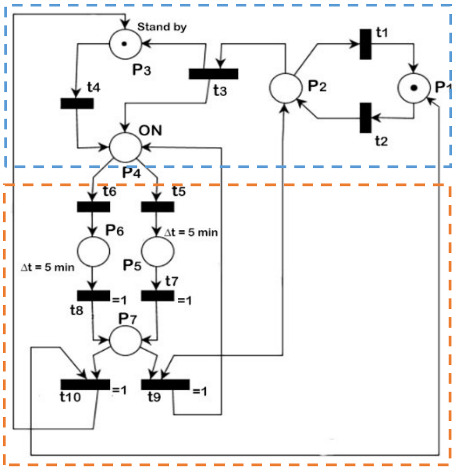

FIGURE 11. Awnings PN controller.

In the same way as the previous temperature control PN, the awnings control PN is divided into two parts. First, the master PN (in red) that shows the main awnings control modes. Second, the slave PN (in blue), responsible for detecting the angle of Sun rays and controlling the opening and closing of the awnings.

The awnings control system goes into standby when the Sun rays reach an angle of 150° in the afternoon to an angle of 45° in the morning. When the angle of the Sun rays is between 45° to 70° and between 130° to 150°, the awnings only open halfway. Between an angle of 70° to 130°, the temperature is very high and consequently, awnings are completely opened.

B. LAYERED DATA TRANSFORMATIONAL SUBSYSTEM

In this section, a strategy of constructing a structured backend graph data model of the greenhouse management system is introduced via a layered architecture, illustrated in Figure 12. It aims at translating unstructured, or even semi-structured, IoT raw data into executable Neo4j cyber statements that create nodes and relationships in the graph database or mod-

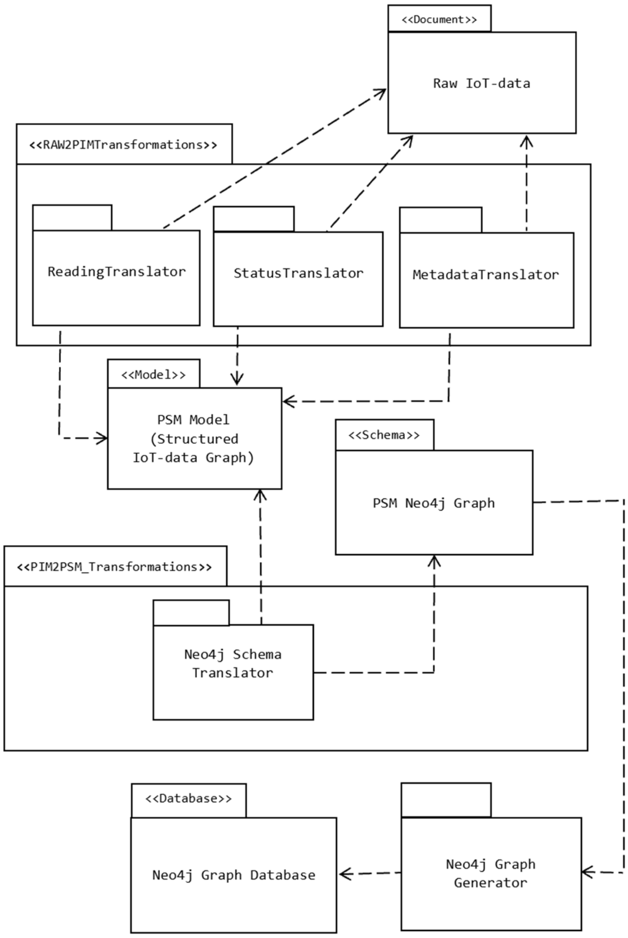

FIGURE 12. A transformations chain of unstructured IoT data into Neo4j Cypher.

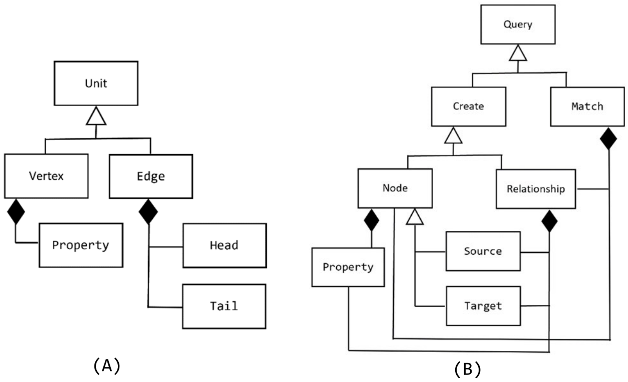

FIGURE 13. Concepts of PIM & PSM graphs at the metamodel level.

TABLE 2. Proposed PN places and transitions.

Places		Transitions	
P1	Sun rays angles < 55 or > 150	t1	Sun rays < 55 or > 150 changes detection
P2	55 < Sun rays < 150	t2	55 < Sun rays < 150 changes detection
P3	Standby mode	t3	Working status changes from On to Standby
P4	On mode	t4	Working status changes from Standby to ON
P5	Partially open awnings	t5	Sun rays < 75 or > 130 changes detection
P6	Completely open awnings	t6	75 < Sun rays < 130 changes detection
P7	Intermediary place	t7, t8, t9, t10	= 1

ify the existing graph. This raw data is captured from various kinds of IoT sensors/ devices with different formats, including actuators, as well as associated information about weather, geographical location and energy consumption. One of the benefits of the proposed layered transformations of data is forming a unified and technology-independent PIM view that contains the collected raw data of interests without bothering with the type of devices that produces these data. This strategy establishes a common ground of a unified representation of PIM model that can be used by different transformations steps.

It is worth mentioning that this layered architecture is considered the detailed design of the data conversion component illustrated previously in Figure 2. Following the MDA principle, the proposed process of data conversion is distributed across three main layers. The first layer aims at constructing a platform-independent model (PIM) from raw, unstructured IoT device data. The second transformation layer intends to derive the platform-specific model (PSM) from the resulting intermediate PIM. Additionally, the final layer is generating Neo4j Cypher statements to construct the Neo4j graph database of the greenhouse.

C. DATA TRANSFORMATIONS & UNDERLYING REPRESENTATIONS

As various formats of streamed data stored in some .csv files, which is captured from different types of IoT devices/ sensors, are used as input to the first layer. These files consist of several kinds of unstructured/ semi-structured critical data, such as reading data, metadata of sensor/ device, energy consumption data in voltage, geo data, weather and location data. Three main translators are responsible for unifying the concepts and representations of input files to generate technology-independent PIM elements, namely, *ReadingTranslator*, *MetadataTranslator* and *StatusTranslator*. This approach of constructing the PIM model leads to enable the reuse and composition of required transformations from PIM to PSM and from PSM to code in technology-independent fashion.

1) FROM IoT RAW DATA TO STRUCTURED IoT DATA GRAPH

It is assumed that suitable serialization techniques are applied to various.csv files, which contains captured data from sensors, for restructuring some of the attributes to construct

$$
\begin{aligned}
&tr{:}\, PIM \to PSM\\
&\quad r1{:}\, Vertex \to Node\\
&\qquad \forall v{:}\, Vertex, \forall a{:}\, Attribute \cdot ((v \in PIM \wedge a \in PIM)\\
&\qquad\qquad\qquad \wedge\, AttributeOf(a, v))\\
&\qquad \to\\
&\qquad \exists!\, n{:}\, Node, \exists!\, p{:}\, Property \cdot ((n \in PSM \wedge p \in PSM) \wedge\\
&\qquad\quad (CreateNode(n, v) \wedge CreateProperty(p, a)\\
&\qquad\qquad\qquad \wedge\, PropertyOf(p, n)))\\
&\quad r2{:}\, Edge \to Relationship\\
&\qquad \forall v1, v2{:}\, Vertex \cdot ((v1 \in PIM \wedge v2 \in PIM) \wedge\\
&\qquad\quad \forall e{:}\, Edge, \forall a{:}\, Attribute \cdot ((e \in PIM \wedge a \in PIM)\\
&\qquad\qquad\qquad \wedge\, AttributeOf(a, e))\\
&\qquad \to\\
&\qquad \exists!\, n1, n2{:}\, Node \cdot ((n1 \in PSM \wedge n2 \in PSM) \wedge\ (ReferTo(n1, v1)\\
&\qquad\qquad\qquad \wedge\, ReferTo(n2, v2))\\
&\qquad \exists!\, r{:}\, Relationship, \exists!\, p{:}\, Property \cdot ((r \in PSM \wedge p \in PSM)\\
&\qquad\qquad \wedge \big(CreateRelationship(r, e) \wedge CreateProperty(p, a)\\
&\qquad\qquad\qquad \wedge\, PropertyOf(p, r)\big) \wedge RelateTo(n1, n2))
\end{aligned}
$$

LISTING 1. **Transformation rules from the PIM graph into PSM graph.**

the PIM graph model. As such some attributes might be eliminated or added, and others may be replaced, split up or merged. The resulting technology-independent PIM is defined using generic terminologies/ concepts expressed in the following metamodel (Figure 13a). A UML class diagram is used for representing PIM concepts at the metamodel level and to show how they relate to each other using *generalization* and *composition* relationships.

It can be seen that the PIM metamodel contains general terms related to graph theory that are more readable and organized with a balance between normalizing and denormalizing graph elements than streamed data stored in.csv files. On the other hand, the metamodel of PSM (Figure 13b) uses a language closed to the implementation of Neo4j query Cyphers for manipulating the graph database.

2) FROM STRUCTURED IoT-DATA GRAPH INTO Neo4j CYPHER GRAPH SCHEMA

At the second phase of transformations, which is a core step aims at deriving final representation of the Neo4j graph query schema (PSM) from the intermediate PIM graph. The following First-Order Predicate Logic (FOPL) rules, Listing 1, express the mapping between PIM elements and PSM elements. These rules might be implemented using the declarative approach and languages of transformations like Atlas Transformation Language (ATL) and Epsilon or imperative one using Java at the implementation phase, which is out of the scope of this article.

```
<Vertex type=”ActuatorOrder“ id=”ao8833”>
    <Attribute name=”actionType“ value=”CLOSE“ />
    <Attribute name=”priority“ value=”1“ />
    <Attribute name=”reqTimestamp“ value=”13:53:28PM“ />
</Vertex>
<Vertex type=” Actuator“ id=” ac1002”>
    <Attribute name=”job“ value=”partialOpen“ />
    <Attribute name=”voltageReq“  value=”73.89V” />
    <Attribute name=”job“ value=”partialOpen“ />
    <Attribute name=”size“  value=”1.2k” />
</Vertex >
<Edge id=”r1” label=”ACTIVATES“>
    <Head ref=”ao8833“ />
    <Tail ref=”ac1002“ />
</Edge >
```

LISTING 2. **Underlying representation of the intermediate PIM graph.**

```
<Query action=”create”>
    <Node name=”node1” type=”ActuatorOrder“ aoId=”ao8833”>
        <Property name=”actionType“ value=”CLOSE“ />
        <Property name=”priority“ value=”1“ />
        <Property name=”reqTimestamp“ value=”13:53:28PM“ />
    </Node>
    <Return ref=”aoId” />
</Query>
<Query action=”create”>
    <Node name=”node2” type=”Actuator“ aid=”ac1002”>
        <Property name=”job“ value=”partialOpen“ />
        <Property name=”voltageReq“  value=”73.89V” />
        <Property name=”job“ value=”partialOpen“ />
        <Property name=”size“  value=”1.2k” />
    </Node>
    <Return ref=”aId” />
</Query>
<Query action=”create”>
    <Relationship id=”r1” label=”ACTIVATES“>
        <Source ref=”ao8833“ />
        <Target ref=”ac1002“ />
    </Relationship >
    <Return ref=”id” />
</Query>
```

LISTING 3. **Underlying representation of the intermediate PSM graph.**

The widely used Extensible Markup Language (XML) format is adopted to express the underlying representation of both PIM and PSM models, due to its ability to represent data in a hierarchal structure. The following listings (Listing 2 and 3) demonstrates a snapshot of a possible representation of PIM model and its equivalent PSM model respectively.

3) GENERATING EXECUTABLE Neo4j CYPHER STATEMENTS FROM Neo4j CYPHER GRAPH

In this phase, the intermediate Neo4j Cypher graph is used as a source form of conceptual schema for generating the final executable Neo4j Cypher queries. The granularity level of the model supports the design of the generator to be simpler, in which many of direct one-to-one mapping rules can be applied to construct the final queries from this low-level PSM model.

The output of this phase is a Cypher script file that contains all query statements for constructing the final Neo4j greenhouse graph database. The following listing

TABLE 3. Description of entities.

Graph Entity	Description	Main Properties
Organisation	A public of private company that manages any activity related to the food production or smart farming and used the proposed system.	regNo, activities, name, ownerId, establishDate, email
Distributor	Details of individual or crop distribution company	disId, email, name, ownerID, regNo, type
Farm	Details of a farm that may consist of some greenhouse units	gpsLoc, length, width, noOfFarmers, noOfGreenHouses
GreenHouse	Details of a greenhouse units including its size and dimensions	gpsLoc, length, width, height, noOfBlocks, shapeType
Block	Details of a specific area within a greenhouse (each greenhouse has a number of bocks)	bNo, length, width, noOfRaisedBed, shapeType
RaisedBed	Details of raised bed in each block	length, rbNo, seedsType, shapeType, soilType, width
Seeds	Details of seeds	seedId, colour, maxHeight, maxWidth, species, stemShape, stemType, surfaceType
Sensor	kind of devices that record measures of parameters (temperature, humidity, pressure, degree of lights and more) that is used in greenhouses	coverage, voltageReq, mUnit, outputParameterType, readingPeriod, senId, type
Actuator	Kind of devices that consumes energy to perform some mechanical or electrical action for controlling the greenhouse.	aId, voltageReq, height, job, length, size, type, width
CropOrders	Maintaining information about customer or distributor orders	amount, coDate, delDate, totalPrice, itemId
CropProduction	Recording all details about the crop production and its prices	amount, cpDate, cpId, cpType, expDate, uPrice, unitType
ActuatorOrders	Maintaining information about actuator orders that are requested by the system for controlling the greenhouse.	actionType, aoId, priority, reqTimeStamp
GrowthRate	Recording the change in crop growth	colour, comment, dayNo, height, measurmentNo, timestamp, width
CompletedActions	Maintaining archival information about all completed actions performed by actuators	actionType, caId, voltageCon, finishingTimeStamp, result
SensingValues	Maintaining all details about raw data captured by a sensor (sensor readings)	voltageCon, svId, timestamp, value

```
CREATE (node1:ActuatorOrder { aoId: 'ao8833',
actionType: 'CLOSE', priority: '1', reqTimestamp:
'13:53:28PM'})
RETURN node1.aoId
CREATE (node2:Actuator { type: 'Awning', job:
'partialOpen', voltageReq: '73.89V', size: '1.2K', aId:
'ac1002'})
RETURN node2.aId
MATCH (node1:ActuatorOrder),(node2:Actuator)
WHERE node1.aoId = 'ao8833' AND node2.aId = 'ac1002'
CREATE (node1)-[r1:ACTIVATES]->(node2)
RETURN type(r1)
```

LISTING 4. A snapshot of the generated Neo4j Cypher script.

(Listing 4) demonstrates an example of Cypher statements for creating two new nodes, *ActuatorOrder* and *Actuator* and a relationship (*Activates*) between them.

D. CONCEPTUAL DESIGN OF GREENHOUSE MANAGEMENT INFORMATION SYSTEM

This section mainly discusses a core component of the greenhouse system, which is the data model. Introducing a design of a conceptual data model that includes all the distinct data managed by the intelligent greenhouse and transferred through the middleware of the system is one of the main objectives of this work. As mentioned earlier, the intelligent greenhouse system can be considered a form of distributed sensor-based data-intensive system, which its connected data can be represented easily using a graph data model. From this perspective, each node in the graph represents an entity, in which it can be a user, thing, or actual data, whereas each edge represents a direct relationship between some entities. Attributes of each entity and relationship are represented by graph properties.

1) GRAPH DATA MODEL

During the analysis phase, it is found that there are a variety of core processes that can be grouped into some collaborated and independent subsystems, namely, greenhouse management information system, temperature control & monitoring subsystem, seeds and production subsystem and data processing and analysis subsystem. Figure 14 depicts a demonstration of the meta graph data model designed for the proposed smart greenhouse management information system. It consists of several nodes connected by some directed relationships.

Each subsystem works intensively with some parts (entities) of the graph data model. Consequently, nodes in the data model are classified into different categories, namely, (1) physical components such as sensor devices and actuators machines (blue nodes), (2) actual monitoring data about sensor readings (values), actuator action logs, seed growth rate (red nodes), (3) structural information about logical entities such as farm, organization, greenhouse and distributor (light blue nodes), and (4) Archival information about Crop Orders, Completed Actions, Actuator Orders and crop production level (pink nodes).

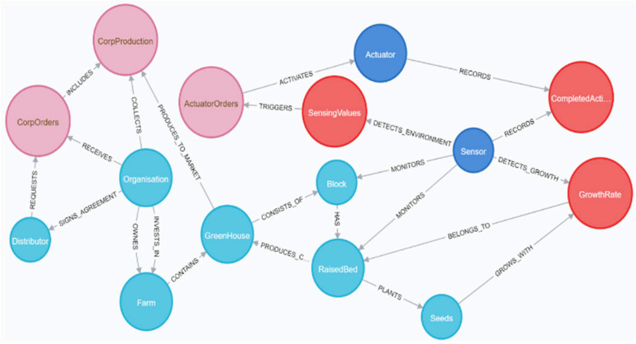

FIGURE 14. The complete graph data model of the greenhouse management information system.

```
MATCH(node1:Seeds),(node2:GrowthRate),(node3:CropProdu
ction)
WHERE node1.seedId = 'seeds9090'
RETURN (node1, node2, node3)
```

LISTING 5. A Neo4j Cypher query for representing the relationship between the kind of seeds and the production speed.

```
MATCH(node1:Seeds),(node2:CompletedActions),(node3:Sen
singValues)
WHERE node1.seedId = 'seeds500'
RETURN (node1, node2, node3)
```

LISTING 6. A Neo4j Cypher query for representing the relationship between the kind of seeds and the overall energy consumption.

2) DESCRIPTION OF GRAPH ENTITIES

In the proposed graph data model 15 entities are identified (Table 3). Regarding the actual monitoring data that are stored in these entities, it is worth mentioning that there is various periodical information used in the proposed subsystems that are collected automatically from different IoT devices and sensors installed in the smart greenhouse.

This data is categorized into environmental, control and seed growth parameters. On one hand, the environmental parameters include greenhouse temperature, humidity and CO_2 level, external temperature, Sun radiation density, soil moisture, chemical level in irrigation water and soil, and more. On the other hand, control parameters include actuator orders, actuator actions and energy consumed. Whereas, the crop growth parameters include the growth rate and characteristics of seeds. In addition to this, the overall farm production information can also be modeled in the graph data model. The information includes organization departments, employees, salaries, production line information, and more.

Examples of Neo4j Cypher Query: Recording the captured microclimate and environmental parameters and storing them in the database is beneficial for future analysis and decision making. This subsection presents two examples of possible running queries. For instance, in order to extract useful information or knowledge related to a specific kind of seeds imported from such a place, a simple Neo4j Cypher query can be executed as:

Another example, when examining the total amount of consumed energy in the greenhouse while producing a specific kind of seeds, another simple Neo4j Cypher query is executed as:

IV. CONTRIBUTION SUMMARY

a) Using PN, through its high-quality supervising properties, allows us to achieve the two first stages, system monitoring and reference temperature generation. These properties justify the choice we made of using this interesting and flexible tool.

b) The fact that awnings are completely or halfway open during the day's high-temperature period, decreases internal greenhouse temperature, and therefore reduces the necessary energy for temperature regulation.

c) The angles of the Sun rays and reference temperatures used in this work are only illustrative values obtained from discussions we had with local greenhouse owners. These values can be modified to fit the specific conditions of each greenhouse.

d) Using an economic mode (higher reference temperature) during the energy consumption rush hour has three advantages.

  - Reducing the consumed energy by sending a relatively higher reference temperature than usual that needs less energy.
  - Reducing better energy performance when it runs using 5G wireless communication technology and MIMO method.
  - Reducing pressure on the public electricity supply during energy consumption rush hours.
  - Reducing energy costs by consuming less energy during the daytime rush hours, because it is well known that energy costs are higher during rush hours.

e) The proposed greenhouse system design is scalable; it can be extended to capture other kinds of properties such as CO_2, humidity, dust level and more associated parameters by attaching additional sensors (with) control components.

f) The proposed system is considered intelligent as it archives all related data for future analysis and prediction.

g) The system supports different brands of sensors as the transformation units will unify all reading formats of the captured data to be structured and consistent with the backend data model.

h) Supporting distribution of system tasks, in which the amount of processing data will be split up across the network edge points.

V. CONCLUSION

In order to overcome the very restrictive climatic conditions in the KSA, a highly scalable intelligent system that monitors the greenhouse environment, generates the reference temperature, and regulates the internal temperature was developed.

The use of a PN model allows us to monitor the greenhouse environment, to generate suitable reference temperatures, and to supervise the whole system.

A controlled awning that reduces the effects of the Sun rays was also introduced.

The proposed system is autonomously able to: monitor the outside temperature, monitor the energy consumption rush hours, monitor the angles of the Sun rays, generate the suitable temperature, send this temperature as a reference signal for temperature regulation, guarantee that the ambient greenhouse temperature reaches this reference temperature, and finally, goes into standby state in the absence of tasks to accomplish.

The main innovative point of this work is to build a smart system through associating different techniques; greenhouse monitoring and supervising through PN, temperature regulation using a closed loop system and awnings control to obtain a smart framework that reduces energy consumption and ensuring an appropriate greenhouse growing environment. This is considered very beneficial while achieving energy savings and reducing production costs, it keeps an appropriate environment for plants to grow. The effectiveness of the proposed approach is demonstrated via a number of simulations that are performed using a greenhouse temperature transfer function.

Additionally, all captured information that is related to energy consumption, internal greenhouse temperature and other agricultural attributes are structured using a proper strategy of model transformations, then stored and archived in the dynamic Neo4j graph-based database system. This is introduced and discussed as a scalable design of a supporting management information system in the second part of this work.

REFERENCES

[1] M. B. Baig and G. S. Straquadine, "Sustainable agriculture and rural development in the Kingdom of Saudi Arabia: Implications for agricultural extension and education," in *Vulnerability of Agriculture, Water and Fisheries to Climate Change*. Dordrecht, The Netherlands: Springer, 2014, pp. 101–116.

[2] S. Fiaz, M. A. Noor, and F. O. Aldosri, "Achieving food security in the kingdom of saudi arabia through innovation: Potential role of agricultural extension," *J. Saudi Soc. Agricult. Sci.*, vol. 17, no. 4, pp. 365–375, Oct. 2018.

[3] V. Keerthi and G. N. Kodandaramaiah, "Cloud IoT based greenhouse monitoring system," *Int. J. Eng. Res. Appl.*, vol. 5, no. 10, pp. 35–41, 2015.

[4] J. A. Sánchez-Molina, N. Pérez, F. Rodríguez, J. L. Guzmán, and J. C. López, "Support system for decision making in the management of the greenhouse environmental based on growth model for sweet pepper," *Agricult. Syst.*, vol. 139, pp. 144–152, Oct. 2015.

[5] M. A. Akkaş and R. Sokullu, "An IoT-based greenhouse monitoring system with micaz motes," *Procedia Comput. Sci.*, vol. 113, pp. 603–608, 2017.

[6] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, "A novel smart energy theft system (SETS) for IoT-based smart home," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5531–5539, Jun. 2019.

[7] K. E. Bouazza and W. Deabes, "Smart Petri nets temperature control framework for reducing building energy consumption," *Sensors*, vol. 19, no. 11, p. 2441, May 2019.

[8] K.-L. Tsai, F.-Y. Leu, and I. You, "Residence energy control system based on wireless smart socket and IoT," *IEEE Access*, vol. 4, pp. 2885–2894, 2016.

[9] R. J. Tom, S. Sankaranarayanan, and J. J. P. C. Rodrigues, "Smart energy management and demand reduction by consumers and utilities in an IoT-fog-based power distribution system," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7386–7394, Oct. 2019.

[10] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K.-C. Wang, "Review of Internet of Things (IoT) in electric power and energy systems," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 847–870, Apr. 2018.

[11] T. Guo and W. Zhong, "Design and implementation of the span greenhouse agriculture Internet of Things system," in *Proc. Int. Conf. Fluid Power Mechatronics (FPM)*, Aug. 2015, pp. 398–401.

[12] L. Dan, C. Xin, H. Chongwei, and J. Liangliang, "Intelligent agriculture greenhouse environment monitoring system based on IoT technology," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City*, Dec. 2015, pp. 487–490.

[13] I. Mat, M. R. Mohd Kassim, A. N. Harun, and I. Mat Yusoff, "IoT in precision agriculture applications using wireless moisture sensor network," in *Proc. IEEE Conf. Open Syst. (ICOS)*, Oct. 2016, pp. 24–29.

[14] T. Gomes, J. Brito, H. Abreu, H. Gomes, and J. Cabral, "GreenMon: An efficient wireless sensor network monitoring solution for greenhouses," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2015, pp. 2192–2197.

[15] M. Danita, B. Mathew, N. Shereen, N. Sharon, and J. J. Paul, "IoT based automated greenhouse monitoring system," in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Madurai, India, Jun. 2018, pp. 1933–1937.

[16] L. K. S. Tolentino, E. O. Fernandez, R. L. Jorda, S. N. D. Amora, D. K. T. Bartolata, J. R. V. Sarucam, J. C. L. Sobrepena, and K. Y. P. Sombol, "Development of an IoT-based aquaponics monitoring and correction system with temperature-controlled greenhouse," in *Proc. Int. SoC Design Conf. (ISOCC)*, Oct. 2019, pp. 261–262.

[17] D. Shinde and N. Siddiqui, "IOT Based environment change monitoring & controlling in greenhouse using WSN," in *Proc. Int. Conf. Inf., Commun., Eng. Technol. (ICICET)*, Pune, India, Aug. 2018, pp. 1–5.

[18] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," *Biosyst. Eng.*, vol. 177, pp. 4–17, Jan. 2019.

[19] P. Rajalakshmi and S. Devi Mahalakshmi, "IOT based crop-field monitoring and irrigation automation," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–6.

[20] D. Zhang, Z. Zhou, S. Mumtaz, J. Rodriguez, and T. Sato, "One integrated energy efficiency proposal for 5G IoT communications," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1346–1354, Dec. 2016.

[21] R. A. Kjellby, L. R. Cenkeramaddi, A. Frøytlog, B. B. Lozano, J. Soumya, and M. Bhange, "Long-range & self-powered IoT devices for agriculture & aquaponics based on multi-hop topology," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Apr. 2019, pp. 545–549.

[22] A. F. Subahi, "Edge-based IoT medical record system: Requirements, recommendations and conceptual design," *IEEE Access*, vol. 7, pp. 94150–94159, 2019.

[23] M. Younas, "Research challenges of big data," *Service Oriented Comput. Appl.*, vol. 13, pp. 105–107, Jun. 2019.

[24] I. Robinson, W. Jim, and E. Emil, *Graph Databases: New Opportunities for Connected Data*. Newton, MA, USA: O'Reilly Media, 2015.

[25] E. Latronico, E. A. Lee, M. Lohstroh, C. Shaver, A. Wasicek, and M. Weber, "A vision of swarmlets," *IEEE Internet Comput.*, vol. 19, no. 2, pp. 20–28, Mar. 2015.

[26] (2019). *Cypher Query Language*. neo4j. Accessed: Dec. 27, 2019. [Online]. Available: https://neo4j.com/developer/cypher-query-language/

[27] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When Internet of Things meets blockchain: Challenges in distributed consensus," *IEEE Netw.*, vol. 33, no. 6, pp. 133–139, Nov. 2019.

[28] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards secure and decentralized sharing of IoT data," IN *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 176–183.

[29] M. A. Islam and S. Madria, "A permissioned blockchain based access control system for IOT," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 469–476.

[30] H. Smidt, M. Thornton, and R. Ghorbani, "Smart application development for IoT asset management using graph database modeling and high-availability Web services," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 1–10.

[31] A. F. Subahi, S. A. Department of Computer ScienceUniversity Collage of Al-JamoumUmm Al-Qura UniversityMakkah, and Y. Alotaibi, "A new framework for classifying information systems modelling languages," *J. Softw.*, vol. 12, no. 12, pp. 18–42, Jan. 2018.

[32] M. Kardoš and M. Drozdová, "Analytical method of CIM to PIM transformation in model driven architecture (MDA)," *J. Inf. Organizational Sci.*, vol. 34, no. 1, pp. 89–99, 2010.

[33] J. Alshahrani and P. Boait, "Reducing high energy demand associated with air-conditioning needs in saudi arabia," *Energies*, vol. 12, no. 1, p. 87, Dec. 2018.

[34] I. L. López-Cruz, E. Fitz-Rodríguez, J. C. Torres-Monsivais, E. C. Trejo-Zúñiga, A. Ruíz-García, and A. Ramírez-Arias, "Control strategies of greenhouse climate for vegetables production," in *Biosystems Engineering: Biofactories for Food Production in the Century XXI*. Cham, Switzerland: Springer, 2014, pp. 401–421.

[35] F. Behrooz, N. Mariun, M. Marhaban, R. M. Mohd, and A. Ramli, "Review of control techniques for HVAC systems—Nonlinearity approaches based on Fuzzy cognitive maps," *Energies*, vol. 11, no. 3, p. 495, 2018.

[36] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment—A review," *Renew. Sustain. Energy Rev.*, vol. 13, nos. 6–7, pp. 1246–1261, 2009.

[37] H. Mirinejad, S. H. Sadati, M. Ghasemian, and H. Torab, "Control techniques in heating, ventilating and air conditioning (HVAC) systems," *J. Comput. Sci.*, vol. 4, no. 9, pp. 777–783, 2008.

[38] T. Blevins, W. WK, and M. Nixon, *Advanced Control Foundation: Tools, Techniques and Applications*. Pittsburgh, PA, USA: International Society of Automation, 2013.

[39] Y. Li, Y. Ding, D. Li, and Z. Miao, "Automatic carbon dioxide enrichment strategies in the greenhouse: A review," *Biosystems Eng.*, vol. 171, pp. 101–119, Jul. 2018.

[40] Instruments National. (2018). *PID Theory Explained*. Accessed: Sep. 23, 2018. [Online]. Available: http://www.ni.com/white-paper/3782/en#top

[41] M. Irshad and A. Ali, "A review on PID tuning rules for SOPTD inverse response processes," in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, Kannur, India, Jul. 2017, pp. 6–7.

[42] J. B. Oliveira, J. Boaventura-Cunha, and P. B. Moura Oliveira, "A feasibility study of sliding mode predictive control for greenhouses," *Optim. Control Appl. Methods*, vol. 37, no. 4, pp. 730–748, Jul. 2016.

[43] M. Ramdani, A. Hamza, and M. Boughamsa, "Multiscale fuzzy model-based short term predictive control of greenhouse microclimate," in *Proc. IEEE 13th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2015, pp. 1348–1353.

[44] M. Guoqi, Q. Linlin, L. Xinghua, and W. Gang, "Modeling and predictive control of greenhouse temperature-humidity system based on MLD and time-series," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 2234–2239.

[45] J. K. Gruber, J. L. Guzmán, F. Rodríguez, C. Bordons, M. Berenguel, and J. A. Sánchez, "Nonlinear MPC based on a volterra series model for greenhouse temperature control using natural ventilation," *Control Eng. Pract.*, vol. 19, no. 4, pp. 354–366, Apr. 2011.

[46] N. T. Haas, M. Ierapetritou, and R. Singh, "Advanced model predictive feedforward/feedback control of a tablet press," *J. Pharmaceutical Innov.*, vol. 12, no. 2, pp. 110–123, 2017.

[47] N. Sigrimis, K. G. Arvanitis, K. P. Ferentinos, and A. Anastasiou, "An intelligent noninteracting technique for climate control of greenhouses," in *Proc. 15th Triennial World Congr.*, 2002, pp. 6–21.

[48] A. Setiawan, L. D. Albright, and R. M. Phelan, "Application of pseudo-derivative-feedback algorithm in greenhouse air temperature control," *Comput. Electron. Agricult.*, vol. 26, no. 3, pp. 283–302, May 2000.

[49] D. M. Atia and H. T. El-madany, "Analysis and design of greenhouse temperature control using adaptive neuro-fuzzy inference system," *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 1, pp. 34–48, May 2017.

[50] M. Berenguel, L. J. Yebra, and F. Rodriguez, "Adaptive control strategies for greenhouse temperature control," in *Proc. Eur. Control Conf. (ECC)*, Sep. 2003, pp. 2747–2752.

[51] K. G. Arvanitis, P. N. Paraskevopoulos, and A. A. Vernardos, "Multirate adaptive temperature control of greenhouses," *Comput. Electron. Agricult.*, vol. 26, no. 3, pp. 303–320, May 2000.

[52] F. Rodríguez, J. L. Guzmán, M. Berenguel, and M. R. Arahal, "Adaptive hierarchical control of greenhouse crop production," *Int. J. Adapt. Control Signal Process.*, vol. 22, no. 2, pp. 180–197, Mar. 2008.

[53] S. L. Speetjens, J. D. Stigter, and G. van Straten, "Towards an adaptive model for greenhouse control," *Comput. Electron. Agricult.*, vol. 67, nos. 1–2, pp. 1–8, Jun. 2009.

[54] M. J. Willis and M. T. Tham, *Advanced Process Control*. Ph.D. dissertation, Dept. Chem. Process Eng., Univ. Newcastle Upon Tyne, London, UK, 1994.

[55] S. Van Straten and H. Van Henten, "Optimal greenhouse cultivation control: Survey and perspectives," *IFAC Proc. Volumes*, vol. 43, no. 26, pp. 18–33, 2010.

[56] N. Bennis, J. Duplaix, G. Enéa, M. Haloua, and H. Youlal, "Greenhouse climate modelling and robust control," *Comput. Electron. Agricult.*, vol. 61, no. 2, pp. 96–107, May 2008.

[57] R. Linker, P. O. Gutman, and I. Seginer, "Robust controllers for simultaneous control of temperature and CO2 concentration in greenhouses," *Control Eng. Pract.*, vol. 7, no. 7, pp. 851–862, Jul. 1999.

[58] H. Oubehar, "High-order sliding mode control of greenhouse temperature," *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 4, no. 3, pp. 548–554, 2016.

AHMAD F. SUBAHI received the B.Sc. degree in computer science from King Abdulaziz University (KAU), Jeddah, Saudi Arabia, in 2002, the M.Sc. degree in information technology from the Queensland University of Technology, Brisbane, QLD, Australia, in 2008, and the M.Sc. degree in advanced computer science and the Ph.D. degree in computer science from The University of Sheffield, U.K., in 2010 and 2015, respectively. Since 2015, he has been an Assistant Professor in computer science with the Computer Science Department, University College of Al Jamoum (JUC), Umm Al-Qura University (UQU), Makkah, Saudi Arabia. He was the former Head of the Department, from 2017 to 2018. He was also the former Vice Dean of the Development and Entrepreneurship, JUC, from 2018 to 2020, where he is currently the Vice Dean of academic affairs. His research interests include advanced software engineering, computer science field, model-driven engineering, domain specific (modeling) languages, automatic code generation and programming languages design, software systems architecture and design, database systems, robotics, and the secure IoT systems engineering.

KHEIR EDDINE BOUAZZA received the B.Eng. degree in computer science from the University of Oran, Algeria, in 1998, and the M.Sc. degree in automatic control and digital signal processing and the Ph.D. degree in automatic control from the University of Lorraine, France, in 2000 and 2004, respectively. From 2004 to 2005, he was a Research Associate (ATER) with the Department of Computer Engineering, IUT de Longwy, University of Lorraine. From 2005 to 2010, he was an Assistant Professor and an Associate Professor with the University of Oran. He is currently an Associate Professor with the University of Umm Al-Qura (UQU), Saudi Arabia. His main research interests include networked control systems, intelligent control systems, non-linear control and estimation, time delay systems, digital signal processing, and wireless sensor networks.

• • •