IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 13, 2018, accepted January 8, 2019, date of publication January 15, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892905

Distributed Continuous-Time Fault Estimation Control for Multiple Devices in IoT Networks

ROBERTO CASADO-VARA[1], (Student Member, IEEE), PAULO NOVAIS[2], (Senior Member, IEEE), ANA BELÉN GIL[1], (Member, IEEE), JAVIER PRIETO[1], (Member, IEEE), AND JUAN MANUEL CORCHADO[1,3,4,5], (Member, IEEE)

[1]Bisite Research Group, University of Salamanca, 37007 Salamanca, Spain
[2]ALGORITMI Centre/Department of Informatics, University of Minho, 4710-070 Braga, Portugal
[3]Department of Electronics, Information and Communication, Faculty of Engineering, Osaka Institute of Technology, Osaka 535-8585, Japan
[4]Air Institute, IoT Digital Innovation Hub (Spain), 37188 Salamanca, Spain
[5]Pusat Komputeran dan Informatik, Universiti Malaysia Kelantan, Kota Bharu 16100, Malaysia

Corresponding authors: Roberto Casado-Vara (rober@usal.es) and Juan Manuel Corchado (corchado@usal.es)

This work was supported in part by the Salamanca Ciudad de Cultura y Saberes Foundation through the Talent Attraction Programme (CHROMOSOME Project).

ABSTRACT This paper investigates distributed continuous-time fault estimation for multiple devices in the Internet-of-Things (IoT) networks by using a hybrid between cooperative control and state prediction techniques. First, a mode-dependent intermediate temperature matrix is designed, which constructs an intermediate estimator to estimate faulty temperature values obtained by the IoT network. Second, the continuous-time Markov chains transition matrix and output temperatures and the sufficient conditions of stability for auto-correct error of the IoT network temperatures are considered. Moreover, faulty devices are replaced by virtual devices to ensure continuous and robust monitoring of the IoT network, preventing in this way false data collection. Finally, the efficiency of the presented approach is verified with the results obtained in the conducted case study.

INDEX TERMS Linear feedback control systems, data handling, algorithm design and analysis, predictive maintenance, nonlinear control systems, fault tolerant systems, IoT.

I. INTRODUCTION

Due to their wide-ranging application, Networked Control Systems (NCSs) have received considerable attention form the scientific community in the last decades [1]. The advances in communications techniques, network topologies and control methods, have all greatly increased the possibilities of NCS. Since multiple sensor, controller and actuator nodes can be flexibly added to an Internet of Things (IoT) network, it is possible for one network to collect data from a wide variety of buildings. However, when the accuracy of sensors decreases, the data they collect are faulty, this leads to inappropriate decisions. Therefore, it is important to improve the IoT network's ability to detect the sensors that do not function correctly [2]. This paper presents a new predictive temperature control algorithm for the predictive management of a large amount of IoT nodes, achieving efficient temperature control. By implementing a system that controls and monitors the accuracy states of the IoT nodes it will be possible to ensure confidence in the data collected by the IoT network. Discrete-time control studies the performance of the system in a discrete-time interval rather than a continuous time interval. Discrete-time control problems in linear systems have been investigated [3]–[5]. Meanwhile, the studies on the discrete-time control of a nonlinear system have also been carried out for triangular systems [6], nonlinear dynamical networks [7], etc. Discrete-time control techniques have been applied for many practical applications, for instance, multi-agent systems [8] and secure communications [5]. Feedback nonlinear systems that represent a class of nonlinear control systems have been widely considered [9], [10]. We address the problem of predictive maintenance of IoT networks in continuous-time, with the aim of increasing the monitoring and control reliability of IoT networks [14]. By using continuous-time Markov chains to predict the future accuracy states of sensors, IoT networks will ensure quality data because their nodes will always work in an optimal state [16], [22].

Motivated by the above observation, this paper proposes a new feedback control algorithm for improved predictive maintenance of the IoT networks. The algorithm finds the IoT nodes that do not function correctly and identifies false data. To optimize the monitoring and control processes of the IoT network, a novel application of the continuous-time Markov chains is used. We predict the future accuracy states of the IoT nodes and in case it is predicted that a sensor will become faulty after the time control period has expired, the controller sends a signal that this IoT node has to be replaced. Before a new sensor is deployed in its place, the control algorithm creates a virtual sensor in its position. This virtual sensor estimates the temperature of the faulty sensor based on the temperature of its neighboring nodes. In this way, the IoT network collects data in continuous-time range without any loss of reliability in the data due to malfunctions in the IoT devices.

Although the problem of data quality and false data detection has been widely studied [11]–[13], the aforementioned works on data quality and false data detection are concerned with discrete-time systems, and the results that correspond to continuous-time systems are relatively few. In fact, continuous time control systems have already been used in a wide range of areas, such as feedback control of nonlinear systems [15], [17], time-delay communications [18], control of marine surfaces [19] or neural networks [20]. Control algorithms face the following challenges in the field of temperature data quality and predictive maintenance of IoT networks:

1) For predictive maintenance in continuous time it is necessary to solve complex differential equations with initial conditions and boundaries that change at each iteration.
2) Algorithms that increase data quality and detect false data can produce false positives. It is important to discriminate between a hot (cold) temperature spot and a faulty sensor.

In this paper, we intend to cover the research gaps in the field of monitoring and control of continuous-time networked systems with multiple IoT devices. Our aim is to present an improved control algorithm which will achieve the maximum allowable efficiency in predictive maintenance. A unified continuous-time hybrid control system model is presented together with a data quality and false data detection algorithm and a feedback control algorithm for predicting the accuracy state of the IoT sensor. The output of the data quality algorithm is the input of the predictive feedback control algorithm. The main contribution of this paper can be summarized as follows:

1) To the best of our knowledge, the proposed approach allows to obtain efficient feedback control for the continuous-time system model with respect to the false data detection or malfunction of IoT devices.
2) A novel way of estimating the accuracy states of IoT nodes from the error in the measurements and through the continuous-time Markov chains the algorithm predicts the future accuracy states of IoT nodes in continuous-time.
3) A new predictive maintenance model based on the prediction of precision states that is updated in continuous time. If the algorithm finds malfunctioning sensors, the control algorithm creates a virtual IoT node to keep measuring the area and requests their replacement.
4) A novel control algorithm capable of integrating the above contributions to provide an innovative IoT network temperature control mechanism.

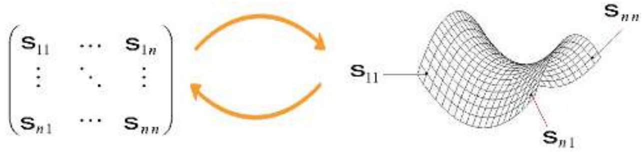

FIGURE 1. Illustrative example of how to create the matrix of IoT nodes from the ordered mesh placed on the map.

The efficiency of the presented approach is illustrated by a numerical case study. Preliminary results on the improvement of data quality and detection of false date in WSNs have been presented in the work of Casado-Vara *et al.* [11].

The rest of the paper is organized as follows. Section II shows the procedure of the control algorithm design in this paper. Case study and simulation results are performed in Section III. Section IV concludes this paper.

II. SYSTEM MODEL

This section presents the control algorithm that we have developed, Fig. 2 shows the model described in this paper. The control algorithm is a hybrid of 2 other algorithms: Cooperative control algorithm and Accuracy state prediction algorithm. The algorithm is hybrid as it combines 2 algorithms designed to solve the same problem; together they work better than individually. But, in this case, we are not simply referring to combining several algorithms to solve the problem; each of the presented algorithms has its own characteristics that significantly improve the functioning of the hybrid algorithm. The cooperative control algorithm detects and auto-corrects erroneous data collected by IoT nodes. So, the control algorithm for predictive system maintenance, which is composed by the following 2 algorithms: 1) Cooperative control algorithm. This algorithm receives the data collected by the IoT network and increases the quality of the data by searching and auto-correcting false data. The output variables of this algorithm are the input variables of the following algorithm 2) Accuracy state prediction algorithm. The predictive algorithm receives data from the cooperative control algorithm and the estimate of the accuracy state of the IoT nodes at time $(t + k)$. Then, the output of this algorithm are the estimated accuracy states. This algorithm implements a predictive maintenance system to make the IoT network more robust. A flowchart of the cooperative control algorithm is shown in Fig. 3 and a flowchart of the accuracy state prediction algorithm is illustrated in Fig. 4.

The algorithm proposed in this paper controls the temperature of a smart supermarket. For this purpose, data collected in time t from the IoT network is the input of the algorithm

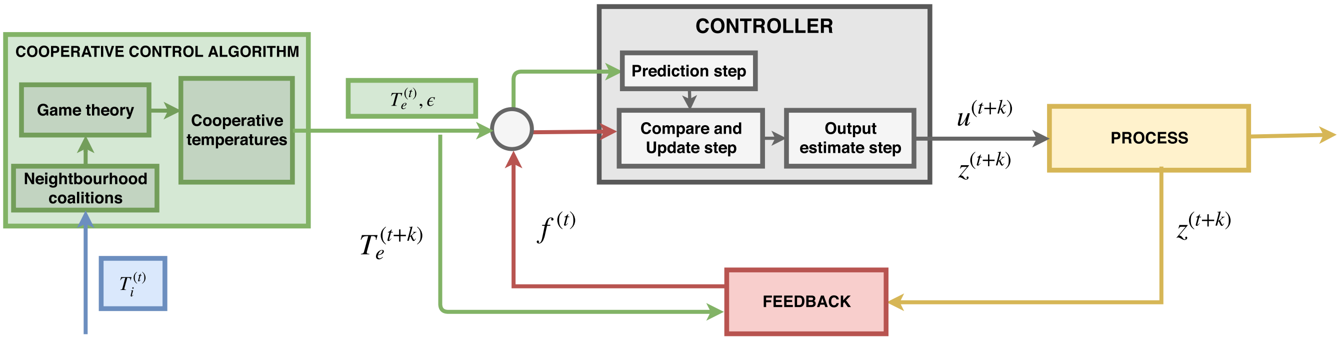

FIGURE 2. This algorithm predicts the accuracy state of the sensors via the feedback control algorithm in the time interval $(t, t+k)$ where k is the control time interval. The input of the algorithm are data from the IoT nodes and the output of the algorithm are controlled parameters of the IoT network. This algorithm can improve the control of the IoT network monitoring and control by its accuracy state prediction step.

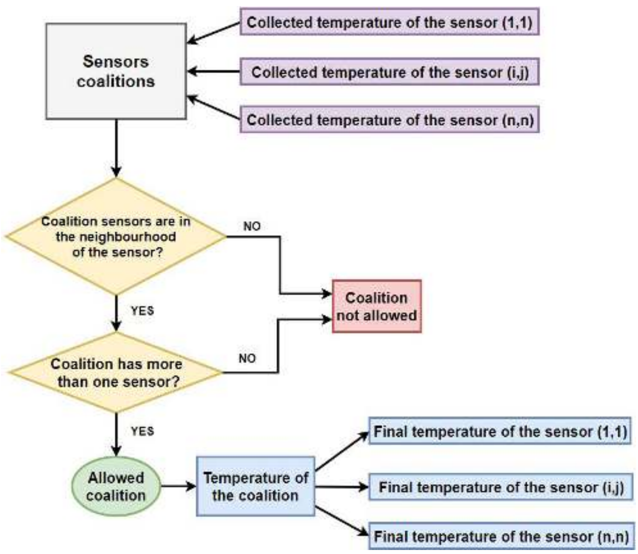

FIGURE 3. Cooperative control algorithm flowchart.

(i.e., $T_i^{(t)}$ in blue block). The cooperative control algorithm forms coalitions of neighboring IoT nodes, these coalitions will detect incorrect temperature values and correct them. This first part of the proposed algorithm calculates the difference between the temperature collected by the IoT network and the optimal output temperature of the cooperative control algorithm. Then, the calculated error (i.e., $T_e^{(t)}$) in time t is sent to the controller as input of the prediction step. The prediction step resolves the Markov strings in continuous-time resulting in the probability that the IoT nodes have the same error that in time t or this error will change. Forecasts of the accuracy state of the IoT nodes are sent to the actuator (i.e., thermostats) to set the process (i.e., smart building) temperature. Two signals are sent from the controller: 1) Since t is the current time in the current loop, k is the time interval that must be determined; $z^{(t+k)}$ predicts the accuracy of the IoT nodes at the end of the time interval $t+k$. 2) The second signal that comes out of the controller $u^{(t+k)}$, determines which IoT nodes need to be repaired and which operate correctly. The process sends the final temperature coming out of the

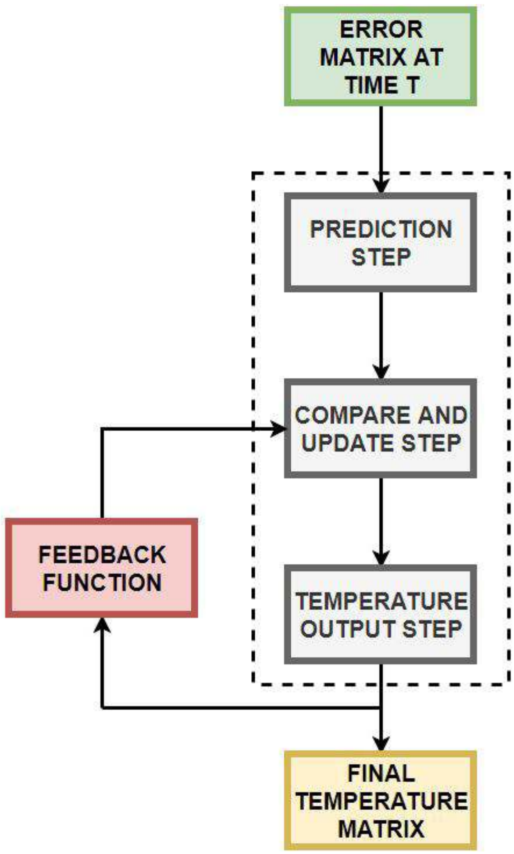

FIGURE 4. Accuracy state control algorithm flowchart.

algorithm to the feedback function that compares the prediction of the accuracy states with the new temperature inputs of the algorithm and corrects the errors in the predictions for the next step of the algorithm.

A. COOPERATIVE CONTROL ALGORITHM

The cooperative control algorithm is located in the reference input. The input of the cooperative control algorithm must be data in a matrix. The IoT nodes placed in Fig. 5 collect data from their environment. Then, we place an ordered mesh in

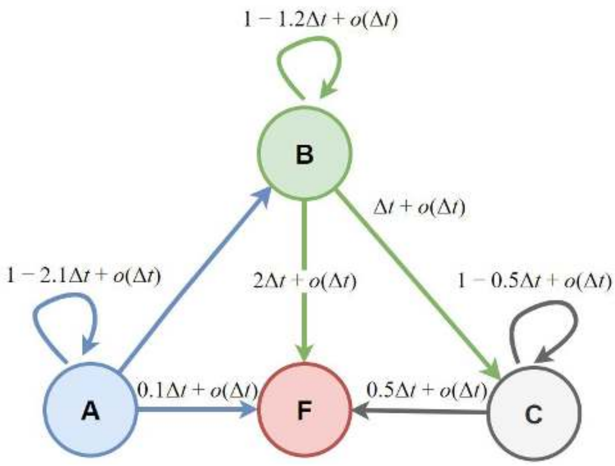

FIGURE 5. Graphical representation of the Markov chain of the solution of the Kolmogorov differential equations of the proposed simulation.

this IoT network in such a way that the IoT nodes match the vertices of the mesh. The mesh is ordered from (1, 1) to (n, n) considering the distance between them (rows and columns). It is easy to create a matrix from the mesh and apply the cooperative control algorithm to it (see Fig. 1). If we have a mesh with n sensors ordered from (1,1) to (n,n), the matrix shown in (1), is created without loss of generality, so the temperature matrix at time t is as follows:

$$T_{n,n} = \begin{pmatrix} t_{s_{1,1}} & \dots & t_{s_{1,n}} \\ \vdots & \ddots & \vdots \\ t_{s_{n,1}} & \dots & t_{s_{n,n}} \end{pmatrix} \tag{1}$$

1) MATHEMATICAL DESCRIPTION OF THE ALGORITHM

Let $n \geq 2$ be the number of players in the game, numbered from 1 to n, and let N = {1, 2, ..., n} be the set of players. A coalition, S, is defined to be a subset of N, $S \subseteq N$, and the set of all coalitions is denoted by $\mathbb{S}$. A cooperative game in N is a function u (characteristic feature of the game) that assigns to each coalition $S_i \subseteq \mathbb{S}$ a real number $u(S_i)$. In addition, one of the conditions is that $u(\emptyset) = 0$. In our case, the characteristic function is non-negative (the values of the characteristic function are always positive), monotonous (if more players are added to the coalition the value of the expected characteristic function does not change), simple and 0-normalized (players are obliged to cooperate with each other since individually they will obtain zero benefit).

In our case, the set of players is the set of ordered sensors S and the characteristic function u is defined as:

$$u : 2^n \longrightarrow \{0, 1\} \tag{2}$$

such that, for each coalition of sensors, u = 1 or 0 depending on whether a particular coalition can vote or not, respectively (see (2, 3)).

$$\mathbb{S} \ni S_i \longrightarrow u(S_i) = \{0, 1\} \in \mathbb{N} \tag{3}$$

where $\mathbb{N}$ are the Natural numbers.

2) COOPERATIVE SENSOR COALITIONS

The sensors will be limited to form coalition only with the sensors in their surrounding environment; that is with neighboring sensors. Let's consider the matrix of the sensors and a pair of sensors $s_{i,j}$ and $s_{k,m}$ will be in the same neighborhood, if and only if:

$$\| (i-k)^2 - (j-m)^2 \| \leq 1 \tag{4}$$

that is, if each sensor to which the game is applied, is the center of a Von Neumann neighborhood, its neighbors are those lying within a Manhattan distance (in the matrix) equal to one. In addition, the following conditions have to be fulfilled by the allowed coalitions:

1) Coalition sensors have to be in the same neighborhood as defined in eq. (4).
2) Coalitions cannot be formed by a single sensor.

3) A CHARACTERISTIC FUNCTION TO FIND *COOPERATIVE* TEMPERATURES

In the proposed game, we want to democratically decide on the temperature of the main sensor (i.e., the sensor currently selected by the algorithm). To do this, the sensors will form coalitions that will decide on the final temperature of the sensor, which will be determined by whether they can vote or not in the process. From the characteristic function defined in eq.(2), if the value is 1(0), the coalition can vote (not vote) respectively. Assume that s_i is the main sensor with its associated temperature t_{s_i}, the characteristic function is built in the following way:

1) First, the average temperature of all the sensors is calculated:

$$T^k_{s_i} = \frac{1}{V}\sum_i^V t_{s_i} \tag{5}$$

here $T^1_{s_i}$ represents the average temperature of the IoT nodes' neighborhood s_i (including it) in the first iteration of the game and V is the number of neighbors in the coalition.

2) The next step is to compute an absolute value for the temperature difference between the temperatures of each sensor and the average temperature:

$$\overline{T}^k_{s_i} = \left(\frac{1}{V}\sum_i^V \mid t_{s_i} - T^k_{s_i} \mid^2\right)^{\frac{1}{2}} \tag{6}$$

3) Using the differences in temperature values with regards to the average temperature $\overline{T}^k_{s_i}$ (see eq.(6)) a confidence interval is created and defined as follows:

$$I^k_{s_i} = \left(T^k_{s_i} \pm t_{(V-1,\frac{\alpha}{2})}\frac{\overline{T}^k_{s_i}}{\sqrt{V}}\right) \tag{7}$$

in eq. (7) we use the Student's-t distribution with an error of 1%.

4) In this step, we use a hypothesis test. If the temperature of the sensor lies in the interval $I^k_{s_i}$, it belongs to

the voting coalition, otherwise, it is not in the voting coalition:

$$u^k(s_1,\ldots,s_n) = \begin{cases} 1 & \text{if } t_{s_i} \in I^k_{s_i} \\ 0 & \text{if } t_{s_i} \notin I^k_{s_i} \end{cases} \quad (8)$$

5) The characteristic function will repeat this process iteratively (k is the number of the iteration) until all the sensors in that iteration belong to the voting coalition. At each iteration k, the following payoff vector of the coalition is available S_j (with $1 \leq j \leq n$ where n is the number of sensors in the coalition) in step k ($PV(S_j^k)$):

$$PV(S_j^k) = (u^k(s_1),\ldots,u^k(s_n)) \text{ where } \sum_i^n u^k(s_i) \leq n \quad (9)$$

The stop condition of the game iterations is $PV(S_j^k) = PV(S_j^{k+1})$. That is, let $PV(S_j^k) = (u^k(s_1),\ldots,u^k(s_n))$ and let $PV(S_j^{k+1}) = (u^{k+1}(s_1),\ldots,u^{k+1}(s_n))$ the iteration process ends when both payoff vectors contain the same elements. This process is shown in the following equation:

$$\begin{cases} u^k(s_1) = u^{k+1}(s_1) \\ \vdots \\ u^k(s_n) = u^{k+1}(s_n) \end{cases} \quad (10)$$

Then, the game finds the solution that is defined in the following subsection.

4) SOLUTION OF THE COOPERATIVE GAME

Once the characteristic function has been applied to all sensors involved in this step of the game, a payoff vector is available in step k (see eq.(9)). Since the proposed game is cooperative, the solution concept is a coalition of players that we have called game equilibrium (GE). The GE of the proposed game is defined as the minimal coalition with more than half of the votes cast, so this voting coalition becomes the winning coalition. Let n be the number of players involved in this step of the game. The winning coalition must satisfy the following conditions:

1) The sum of the elements of the coalition PV must be higher than half plus 1 of the votes cast:

$$\sum_i^n u^k(s_i) \geq \frac{n}{2} + 1 \quad (11)$$

2) The coalition is maximal (i.e., coalition with the greatest number of elements, different from 0, in its payoff vector $PV(S_j^k)$).

Therefore, the solution to the proposed game is the coalition, from among all possible coalitions that are formed at each step k of the game, that satisfies both conditions.

5) TEMPERATURES OF THE WINNING COALITION

Once the characteristic function decides which is the winning coalition, it is possible to calculate the temperature of the main sensor. Let $\{s_1,\ldots,s_j\}$ be the winning coalition's sensors and $\{t_{s_1},\ldots,t_{s_j}\}$ be their associated temperature.

The temperature that the game has voted to be the main sensor's temperature (MST) is calculated as follows:

$$MST = \max_{j \in |S_{winner}|} \{j \cdot t_{s_i}\}_{s_i \in S_{winner}} \quad (12)$$

where $|\mathbb{S}_{winner}|$ is the number of elements in the winning coalition. Therefore, the MST will be the maximum temperature that has the highest relative frequency. In the case of a draw, it is resolved by the Lagrange criterion.

6) DIFFUSE CONVERGENCE

There is a temperature matrix at each game iteration, (see (1)). Hence, we define a sequence of arrays $\{M_n\}_{n\in\mathbb{N}}$ where the M_i element corresponds to the temperature matrix in step i of the game. Therefore, it can be said that the sequence of matrices is convergent if:

$$\forall \epsilon > 0, \text{ there is } N \in \mathbb{N} \text{ such that } |M_{i-1} - M_i| \leq \epsilon \quad \forall i \in \mathbb{N}. \quad (13)$$

That is, if the element $m^{i-1}_{n,m} \in M_{i-1}$ and the element $m^i_{n,m} \in M_i$ are set and the convergence criterion is applied, we have:

$$\forall \epsilon_{n,m} > 0 \text{ there is } N \in \mathbb{N} \text{ such that } |m^{i-1}_{n,m} - m^i_{n,m}| \leq \epsilon_{n,m} \quad \forall i \in \mathbb{N}, \forall i \geq N \text{ and } m^{i-1}_{n,m} \in M_{i-1}, m^i_{n,m} \in M_i \quad (14)$$

Therefore, by applying the criterion of convergence in (14) for each of the elements, a new matrix is obtained; by calculating the difference in the temperatures obtained in the game's previous step and those obtained in the current step.

$$\begin{pmatrix} |m^{i-1}_{1,1} - m^i_{1,1}| & \ldots & |m^{i-1}_{1,m} - m^i_{1,m}| \\ \vdots & \ddots & \vdots \\ |m^{i-1}_{n,1} - m^i_{n,1}| & \ldots & |m^{i-1}_{n,m} - m^i_{n,m}| \end{pmatrix} \quad (15)$$

For the succession of matrices to be convergent, each of the sequences of elements that are formed with the $|m^{i-1}_{n,m} - m^i_{n,m}|$ must be less than the fixed $\epsilon > 0$. In this work, it is established that $\epsilon = 0.01$. The game reaches the equilibrium if at least 80 % of the elements of the matrix are convergent

B. CONTINUOUS-TIME MARKOV CHAINS MATHEMATICAL BACKGROUND

In this subsection we present a practical mathematical background of continuous-time Markov chains and demonstrate how to find the solution of continuous-time Markov chains.

1) CONTINUOUS-TIME MARKOV CHAINS

Let $\{X_t\}_{t\geq 0}$ be a stochastic process defined over a probabilistic space (Ω, A, P), and let $\{F_t\}_{t\geq 0}$ be a family of growing subσ-something of A such that $\forall t \geq 0$

$$X_t : (\Omega, F_t) \to (\mathbb{R}, \mathbb{B}) \quad (16)$$

where $\mathbb{B}$ is the Borel algebra, is measurable [24].

Definition 1: It is said that $\{X_T\}_{t\geq 0}$ is a Markov process regarding

$$\{F_t\}_{t\geq 0} \quad if\ \forall B \in \mathbb{B} \quad \text{and}\ \forall s < t$$
$$P\{X_t \in B|F_s\} = P\{X_t \in B|X_s\} \tag{17}$$

Actually, it is most common to take $F_t = \sigma(X_s, s \leq t)$, that is, the minimum σ-algebra in Ω that makes all applications measurable X_s for all $s \leq t$.

If all random variables X_t take values within a measurable space $(E, \mathcal{E})$, this space is called the process status space, where $E \subset \mathbb{R}$ and $\mathcal{E} = \mathbb{B}_E$.

Definition 2: Given a Markov process of $\{X_t\}_{t\geq 0}$ (for any family of σ-algebras) it is called a transition function to the $P(s, x, t, B)$ function defined for

$$s \leq t \in [0, \infty), \quad x \in \ \text{and}\ B \in \mathcal{E} \tag{18}$$

in order to

$$P(s, x, t, B) = P\{X_t \in B|X_s = x\}. \tag{19}$$

The transition function has to fulfill the following properties [24]:

1) $\forall s \leq t \in [0, \infty), \forall B \in \mathcal{E} \quad P(s, \cdot, t, B)$ is a measurable function of $(E, \mathcal{E})$ in $(\mathbb{R}, \mathbb{B})$.
2) $\forall s \leq t \in [0, \infty), \forall x \in E \quad P(s, x, t, \cdot)$ is a measure of probability of $\mathcal{E}$.
3) Chapman-Kolmogorov equation: $\forall s \leq u \leq t \in [0, \infty), \forall x \in E, \forall B \in \mathcal{E}$,
$$P(s, x, t, B) = \int_{\mathcal{E}} P(s, x, u, dy)P(u, y, t, B) \tag{20}$$
4) $\forall s \in [0, \infty), \forall x \in E, P(s, x, s, E - \{x\}) = 0$

2) TRANSITION FUNCTION AND PROCESS DISTRIBUTION

Since the status space E is discreet, the probability of transition $P\{X_{s+1} \in B/X_s = x\}$, can be expressed by the probability function $P\{X_{s+1} = j/X_s = i\}$, which is called the probability of transition from state i to j between s and $s + t$. Let's assume that the transition probabilities are stationary (i.e., $P\{X_{s+1} = j/X_s = i\}$ is not dependent on s but is only a function of t). In this case, the transition probability from i to j in any length of t time can be represented by $p_{ij}(t)$ [24].

General properties of the transition functions are:

1)
$$p_{ij}(t) \geq 0 \quad \forall i, j \in E \quad \forall t \geq 0 \text{ and}$$
$$\sum_{j\in E} p_{ij}(t) = 1 \quad \forall i \in E \quad \forall t \geq 0. \tag{21}$$

2)
$$p_{ij}(s + 1) = \sum_{k\in E} p_{ik}(s)p_{kj}(t) \quad \forall i, j \in E \quad \forall s, t \geq 0. \tag{22}$$

which is the discreet version of the Chapman-Kolmogorov equation.

3)
$$p_{ij}(0) = \delta_{ij} = \begin{cases} 1 & if\ i = j \\ 0 & if\ i \neq j \end{cases} \tag{23}$$

The transition probabilities that correspond to the same time span t, are usually ordered in the form of a matrix

$$P(t) = (p_{ij}(t))_{i,j\in E} \tag{24}$$

in which the row sub-index represents the primary state and the column sub-index represents the final state.

Distribution of a Markov process in continuous time, with discrete state space and stationary transition probabilities, is determined by:

- The space of states E (finite or numberable).
- The family of transition matrices $\{P(t)\}_{t\geq 0}$ that obey Chapman-Kolmogorov's equation.
$$P(s + t) = P(s)P(t) \tag{25}$$
- The initial distribution
$$P\{X_0 = i\} = p_i(0) \quad \forall i \in E. \tag{26}$$
which is usually expressed as a row vector $p(0)$.

We now find the explicit formulas that allow us to express the distribution of the process. First of all the marginal distributions of the process variables are:

$$P\{X_t = i\} = p_j(t) = \sum_{k\in E} p_k(0)p_{ki}(t) \quad \forall i \in E, \tag{27}$$

thus, expressed in the form of a vector

$$p(t) = p(0)P(t) \tag{28}$$

Since time is continuous, there is no unit of time that represents the minimum time lapse between two consecutive instants, and according to which the probabilities of transition can be expressed in more than one stage. For this reason, it doesn't come with a single matrix transaction, but you need one for every $t \geq 0$. However, since the transition matrices must fulfill the equation $P(s + t) = P(s)P(t)$, which stresses that it is not necessary to know all these matrices a priori, but that they can be calculated from each other. In particular, since we know the value of $P(t)$ for $t \in [0, \varepsilon)$ any $P(t)$ can be calculated $t > 0, \forall \varepsilon > 0$.

It is necessary to introduce a restriction to the model we propose in this paper, it is a quite natural restriction that is verified in most practical situations, it consists in supposing that:

$$\forall i, j \in E, \quad p_{ij}(t) \text{ converges on } \delta_{ij} \text{ when } t \downarrow 0; \tag{29}$$

or in matrix form:

$$P(t) \underset{t\downarrow 0}{\longrightarrow} I \tag{30}$$

when I is the identity matrix.

A continuous time Markov process, with discrete state space and stationary transition probabilities, is said to be

TABLE 1. Accuracy state of sensors.

X_n	Sensor accuracy state	Error (%)
A	High accuracy	e $\leq$ 10
B	Accurate	10 < e $\leq$ 20
C	Low accuracy	20< e $\leq$ 35
F	Failure	e $\geq$ 35

stochastically continuous or standard when it fulfills the above condition of continuity in the source of transition probabilities. In this manuscript, we will assume that this condition is always fulfilled [24].

C. ACCURACY STATE PREDICTION ALGORITHM

In this subsection, we propose a new feedback control algorithm for predictive maintenance and improve the monitoring and control of the IoT networks. By using the above mathematical background, we can build a continuous-time Markov chain model called Kolmogorov's differential equation which is a particular case of a Markov chain. We outline the states and the allowed actions in the accuracy states changes, in Table. 1. The basic parameters of the continuous time markov chains remain stable throughout the process, however, the probabilities change according to Eq. (39) (i.e., in continuous time Markov processes the probabilities of state change at each time t).

1) INITIAL ACCURACY STATE

Initially, it is necessary to define a scale of accuracy degradation expressed in percentages. This is done according to the data obtained by the algorithm that we had developed in a previous research [11]. This scale will be the discussion universe of the random variable X_n that defines the current state of precision of the system related to the error of the sensors. Therefore, the sensors' possible states are $X_n = \{A = high\ accuracy,\ B = accurate,\ C = low\ accuracy,\ F = failure\}$. Below, table 1 contains the selection made for each variable.

Let $T_i^{(t)}$ be the matrix of initial temperatures at time t collected by the WSN, and let $T_f^{(t)}$ be the final temperatures, obtained after applying the data quality algorithm. Then, the accuracy error matrix of the sensors, according to the data quality algorithm, is given by the following equation:

$$T_e^{(t)} = |T_f^{(t)} - T_i^{(t)}| \tag{31}$$

where the coefficients e_{ij} of the matrix $T_e^{(t)}$ are the differences between the initial and final temperature in absolute value for each sensor.

Given the $T_e^{(t)}$ matrix, we now apply the error correction given by the allowed error margin ϵ, and adjust the error matrix:

$$T_\epsilon^{(t)} = |T_e^{(t)} - Id \cdot \epsilon| \tag{32}$$

Now, let's centralize these measures to calculate the states of the sensors. For this purpose, we calculate the average of the elements in the array m_ϵ and the maximum of the array $T_\epsilon^{(t)}$ that we call max_ϵ. Therefore, the centralizing measure is defined as:

$$\delta = m_\epsilon + max_\epsilon \tag{33}$$

this measure is applied to the matrix $T_\epsilon^{(t)}$ to calculate the percentages associated with each error and therefore calculate the states of each sensor:

$$T_\delta^{(t)} = \begin{pmatrix} t_{1,1}^\delta = \dfrac{(t_{1,1}x100)}{\delta} & \dots & t_{1,n}^\delta = \dfrac{(t_{1,n}x100)}{\delta} \\ \vdots & \ddots & \vdots \\ t_{n,1}^\delta = \dfrac{(t_{n,1}x100)}{\delta} & \dots & t_{n,n}^\delta = \dfrac{(t_{n,n}x100)}{\delta} \end{pmatrix} \tag{34}$$

Then, one can define the following function in order to estimate the accuracy state of the IoT nodes in time t. For this purpose we use the Solution of Kolmogorov's differential equations to design this function:

$$g^{(t)} : M_{n,n}(\mathbb{R}) \longrightarrow M_{n,n}(\{X_n\}) = T^{g(t)} \tag{35}$$

defined as follows:

$$g^{(t)}(t_{i,j}^\delta) = \begin{cases} A & if\ t_{i,j}^\delta \leq 10\% \\ B & if\ 10\% < t_{i,j}^\delta \leq 20\% \\ C & if\ 20\% < t_{i,j}^\delta \leq 35\% \\ F & if\ t_{i,j}^\delta \geq 35\% \end{cases} \tag{36}$$

where $t_{i,j} \in T_\delta^{(t)}$, and let $T^{g(t)}$ be the matrix with the accuracy states of the sensors at time t.

2) TRANSITION MATRIX

Let λ_A be the time the IoT node remains in state A (exponential distribution). λ_B and λ_C are defined in a similar way. And let ξ_A be the time the IoT node remains in state A. Let μ_A (μ_B, μ_C) be the probability that an IoT node in state A (B,C) at time t shift to state F in the time interval $(t, \Delta t + t)$. Thus, if the IoT node was in state A at time t, the probability that the IoT node remain in state A at time $t + \Delta t$ knowing it was previously in state A (conditional probability), is given by the following equation:

$$\begin{aligned} p_{AA} &= P(\xi_A > t + \Delta t | \xi_A > t) = \frac{e^{-\lambda_A(t+\Delta t)}}{e^{-\lambda_A t}} \\ &= e^{-\lambda_A \Delta t} = 1 - \lambda_A \Delta t + o(\Delta t) \end{aligned} \tag{37}$$

the second equality follows from the simple fact that $P(A \cap B \mid A) = P(B|A)$ where we let $A = \{X(u) = i \text{ for } 0 \leq u \leq s\}$ and $B = \{X(u) = i \text{ for } s < u \leq s + t\}$.

Similarly, the probability that a sensor in state A at the beginning, will shift to state B, is given by the following equation

$$\begin{aligned} p_{AB} &= P(\xi_B > t + \Delta t | \xi_A > t) \\ &= 1 - ((1 - \lambda_A \Delta t + o(\Delta t)) - (\mu_A \Delta t + o(\Delta t))) \\ &= (\lambda_A - \mu_A)\Delta t + o(\Delta t) \end{aligned} \tag{38}$$

In this way, we can build the transition matrix between t and $t + \Delta t$, where the coefficients of the transition matrix are the probabilities of the sensors' switching states (e.g., p_{AF} is the probability that a sensor in state A at the beginning, will eventually shift to state F in the interval $(t, \Delta t + t)$).

In this way, the transition matrix $P(t)$ is:

$$P(t) = \begin{pmatrix} P(\xi_A > t+\Delta t | \xi_A > t) = p_{AA} & \dots & p_{AF} \\ \vdots & \ddots & \vdots \\ P(\xi_A > t+\Delta t | \xi_F > t) = p_{FA} & \dots & p_{FF} \end{pmatrix} \quad (39)$$

3) PREDICTIVE CONTROL ALGORITHM

Here we describe how the control algorithm works. This algorithm is used by the sensor control system to monitor and control the accuracy of the sensors. In Fig. 2, the set point (green arrow) with the reference inputs contain the following variables: 1) The accuracy error matrix, T_e (see Eq. (31)). This matrix has the precision errors of the mesh of sensors. Each step of the algorithm at every time t, this matrix is introduced to update the data of the algorithm. 2) The allowed error ϵ. This parameter enters the flow in each of the steps of the algorithm.

a: CONTROLLER

The first action performed by the controller is the prediction step. In this stage of the algorithm, the transition matrix of the developed model is used (see (Eq. 39)). Let $z^{(t)} : T^{g(t)} \longrightarrow z^{(t)}(T^{g(t)}) = T^{z(t+k)}$ be the prediction function of accuracy states (i.e., Prediction step) for each time t and let $t+k$ where $k \in \{1, 2, \cdots\}$ be the predicted time. Given that $t_{i,j}^{\delta} \in T^{\delta}$, the controller function u is defined as follows:

$$z_{ij}^{(t+k)}(t_{i,j}^{g}) = max\{\mathbb{P}_{t_{i,j}^{g(t+k)}A}, \mathbb{P}_{t_{i,j}^{g(t+k)}B}, \mathbb{P}_{t_{i,j}^{g(t+k)}C}, \mathbb{P}_{t_{i,j}^{g(t+k)}F}\} \quad (40)$$

Let $z^{(t)}(T^{g}) = T^{z(t+k)}$ be the matrix of the states of accuracy given by the prediction function. The output of this function is the accuracy state of the sensors at time t.

The next step of the algorithm is to compare the measurements of the IoT nodes with those of the feedback function in order to update them. Let $x^{(t)} : T^{z(t)}xT^{f(t-k)} \longrightarrow x^{(t)}(T^{z(t)}) = T^{x(t)}$ be the comparison function defined by the following numerical values $\{A = 1, B = 2, C = 3, F = 4\}$ as follows:

$$x^{(t)}(t_{i,j}^{z(t)}, t_{i,j}^{f(t-k)}) = w_{x_1(t)} t_{i,j}^{z(t)} + w_{x_2(t)} t_{i,j}^{f(t-k)} \quad (41)$$

where $w_{x_n(t)}$ with $n \in \{1, 2\}$ are the weights given for each of the coordinates of the function x.

Let $y^{(t)} : T^{x(t)} \longrightarrow y^{(t)}(T^{x(t)}) = T^{y(t)}$ be the update function defined as follows:

$$y^{(t)}(T^{z(t)}, T^{f(t)}) = \begin{cases} 1 & if\ 0 \le t_{i,j}^{h(t)} \le 1.5 \\ 2 & if\ 1.5 < t_{i,j}^{h(t)} \le 2.5 \\ 3 & if\ 2.5 < t_{i,j}^{h(t)} \le 3.5 \\ 4 & if\ t_{i,j}^{h(t)} \ge 3.5 \end{cases} \quad (42)$$

The update function refreshes the accuracy states of the prediction function with the results obtained from the comparison function.

Let $u : T^{y(t)} \longrightarrow u^{(t)}(T^{y(t)}) = T^{u(t)}$ be the controller function (i.e., output estimate step) and let $T^{u(t)}$ be the system controller matrix at time t. Then, this function finds sensors that are in a state of failure (F). In this way, the system creates a virtual sensor to maintain system monitoring. In addition, it will identify the faulty sensors, optimize the positioning of the sensors or it will send a request to the service staff to replace the malfunctioning sensor. Given that $t_{i,j}^{y(t)} \in T^{y(t)}$, u is defined as follows:

$$u(t_{i,j}^{y(t)}) = \begin{cases} 1 & if\ t_{i,j}^{y(t)} = F \\ -1 & if\ t_{i,j}^{y(t)} \neq F \end{cases} \quad (43)$$

thus, if $u(y^{(t)}) = 1$, the system creates a virtual sensor in the position (i, j) and requests maintenance.

b: FEEDBACK

Let $h^{(t)} : T^{g(t)}xT^{g(t+k)}xT^{z(t+k)} \longrightarrow h^{(t)}(T^{z(t+k)}) = T^{h(t)}$ be the auxiliary feedback function. Since $k \in \{1, 2, \cdots\}$ and the accuracy states in numerical values are $\{A = 1, B = 2, C = 3, F = 4\}$, h is defined as follows:

$$\begin{aligned} h^{(t)}(t_{i,j}^{g(t)}, t_{i,j}^{g(t+k)}, t_{i,j}^{z(t+k)}) &= w_{h_1(t)} t_{i,j}^{g(t)} + w_{h_2(t)} t_{i,j}^{g(t+k)} \\ &\quad + w_{h_3(t)} t_{i,j}^{z(t+k)} \end{aligned} \quad (44)$$

where $w_{h_n(t)}$ with $n \in \{1, 2, 3\}$ are the weights given for each of the coordinates of the function h.

Let $f^{(t)} : T^{h(t)} \longrightarrow f^{(t)}(T^{h(t)}) = T^{f(t)}$ be the feedback function defined as follows:

$$f^{(t)}(T^{h(t)}) = \begin{cases} A & if\ 0 \le t_{i,j}^{h(t)} \le 1.5 \\ B & if\ 1.5 < t_{i,j}^{h(t)} \le 2.5 \\ C & if\ 2.5 < t_{i,j}^{h(t)} \le 3.5 \\ F & if\ t_{i,j}^{h(t)} \ge 3.5 \end{cases} \quad (45)$$

The feedback function returns the accuracy state of the sensor (i, j) back to the flow. In this way, it is verified that the controller is working correctly and that virtual sensors are not created for the repair of sensors that are working properly.

c: PROCESS

The process matrix $T^{p(t)}$ shows when sensors need maintenance. The process matrix is defined as follows:

$$T^{p(t)} = T^{u(t-1)} + T^{u(t)} \quad (46)$$

Thus, when the coefficient of the matrix corresponds to a particular sensor, it means that it has to be replaced $t_{(i,j)}^{p(t)} \ge 0.5\% t_{max}$ time periods with $t_{(i,j)}^{p(t)} \in T^{p(t)}$ (i.e., assuming that $t_{max} = 5$ years, then a sensor has to be replaced if $t_{(i,j)}^{p(t)} \ge 9$ days).

TABLE 2. Symbol and name of all variables used in the control algorithm.

Symbol	Name of the parameter
t	time
k	time interval that we want to control
PV	Payoff Vector of the cooperative control algorithm
s_i	position of the IoT node in its neighbourhood
$T^k_{s_i}$	average temperature of the IoT nodes' neighbourhood
$I^k_{s_i}$	confidence interval of the IoT nodes' neighbourhood
$u^k(s_1,\ldots,s_n)$	characteristic function of the cooperative control algorithm
MST	Main Sensor Temperature
p_{AB}	probability of beginning in state A and finishing in state B
$T^{(t)}_i$	Initial temperature matrix at time t
$T^{(t)}_f$	Final temperature matrix at time t
$T^{(t)}_e$	Error temperature matrix at time t
δ	Normalized value
m_e	Mean of $T^{(t)}_e$ matrix
max_e	Maximun element of $T^{(t)}_e$ matrix
$T^{(t)}_\delta$	Normalized temperature matrix at time t
$P(t)$	Transition matrix of Markov chain
$g^{(t)}$	Estimation accuracy function (vector with probability of each of the states)
$z^{(t)}$	Accuracy state prediction function
$x^{(t)}$	controller comparison function
$y^{(t)}$	controller update function
$u^{(t)}$	Controller function
$f^{(t)}$	Feedback function
$T^{p(t)}$	Temperature process matrix

Then, the controller function sends a signal to the process which sends back the matrix of final virtual temperatures at time t (i.e., $T^{(t)}_{vf}$). When the controller sends the signal that a sensor is in the state of failure, the process creates a virtual sensor in that position and simulates the temperature so that the monitoring and control of the building does not lose efficiency. Let $\{T^{(t)}_f\}_{t\geq 0}$ be the matrix succession with the final temperatures at time t given by the algorithm described in [11] (see appendix A). Moreover, let $VS^{(t)}_{i,j}$ be the virtual sensor in the position (i,j) at time t. Then, the temperature of the $t^v_{i,j}$ is provided by the temperature $t_{i,j} \in T^{(t)}_f$. to increase the understanding of the proposed model, all the variables used in the algorithm can be seen in Table 2.

III. RESULTS

In this section, we present the case study and the results obtained during the experiment. The control algorithm gets data collected by the IoT nodes and auto-corrects the faulty data. Furthermore, in case the controller predicts that an IoT node will be in a state of malfunction, it will create a virtual temperature sensor in order to keep the reliability of the IoT network. In this way, the monitoring and control efficiency of the IoT network is improved.

A. CASE STUDY EXPERIMENTAL SETUP

This case study supposes that the IoT nodes (i.e., temperature sensor) can undergo 4 accuracy states throughout their useful life (A = high accuracy, B = accurate, C = low accuracy, F = failure). The probability that a sensor in state A at instant t shift to state F in the time interval $(t, t+\Delta t)$ is $0.1\Delta t+o(\Delta t)$, if it is in state B it is $0.2\Delta t + o(\Delta t)$ and if it is in state C it is $0.5\Delta t + o(\Delta t)$. In this simulation we assume that the time during which the sensors remain in state A is an exponential time of $\lambda = 2.1$ in state A and $\mu = 1.2$ in state B.

From A in a time interval $(t, t + \Delta t)$ the sensor can pass to F with probability $0.1\Delta t + o(\Delta t)$. If ξ is the time the sensor stays at A, you have it:

$$P(\xi > t+\Delta t|\xi > t) = \frac{e^{-2.1(t+\Delta t)}}{e^{-2.1t}} = e^{-2.1\Delta t} = 1-2.1\Delta t + o(\Delta t) \quad (47)$$

Therefore, the (47) is the probability of remaining in state A at instant t_{i+1} if it was in A at instant t_i. Then, the probability of shifting to B between t and $t + \Delta t$ is

$$1 - ((1 - 2.1\Delta t + o(\Delta t)) - (0.1\Delta t + o(\Delta t))) = 2\Delta t + o(\Delta t) \quad (48)$$

In the successive stages we finally reach to a calculation in which the transition matrix is between t and $t+\Delta t$, as shown in Table 3.

Thus, the derivative of the matrix in the zero is:

$$P'(0) = \begin{pmatrix} -2.1 & 2 & 0 & 0.1 \\ 0 & -1.2 & 1 & 0.2 \\ 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (49)$$

which may be expressed using the Jordan matrix form for the whole period of time t as follows:

$$P(t) = \frac{1}{0.504}\begin{pmatrix} 1 & 1 & 2 & 1 \\ 1 & 0.8 & 0.9 & 0 \\ 1 & 0.56 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & \\ & e^{-0.5t} & & \\ & & e^{-1.2t} & \\ & & & e^{-2.1t} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0.504 \\ 0 & 0 & 0.9 & -0.9 \\ 0 & 0.56 & -0.8 & 0.24 \\ 0.504 & -1.12 & 0.7 & -0.084 \end{pmatrix} \quad (50)$$

For example, the term $p_{AF}(t)$ represents the probability that a sensor that begins its useful life at stage A, will function incorrectly at time t, so:

$$P(\text{Life span} \leq t) = p_{AF} = 1 - \frac{0.9}{0.504}e^{-0.5t} + \frac{0.48}{0.504}e^{-1.2t} - \frac{0.084}{0.504}e^{-2.1t} \quad (51)$$

Figure 5 shows a graphical representation of the Markov chain. Probabilities of changes in the accuracy states of the sensors are shown in Table 3. The instance simulation presented in this section demonstrates that sensors in any

TABLE 3. In this case study, we have assumed that state F is absorbent. That is, for the sensor to move from F to any other state, it needs to be repaired by a maintenance worker.

	A	B	C	F
A	$1-2.1\Delta t+o(\Delta t)$	$2\Delta t+o(\Delta t)$	$o(\Delta t)$	$0.1\Delta t+o(\Delta t)$
B	0	$1-1.2\Delta t+o(\Delta t)$	$\Delta t+o(\Delta t)$	$0.2\Delta t+o(\Delta t)$
C	0	0	$1-0.5\Delta t+o(\Delta t)$	$0.5\Delta t+o(\Delta t)$
F	0	0	0	1

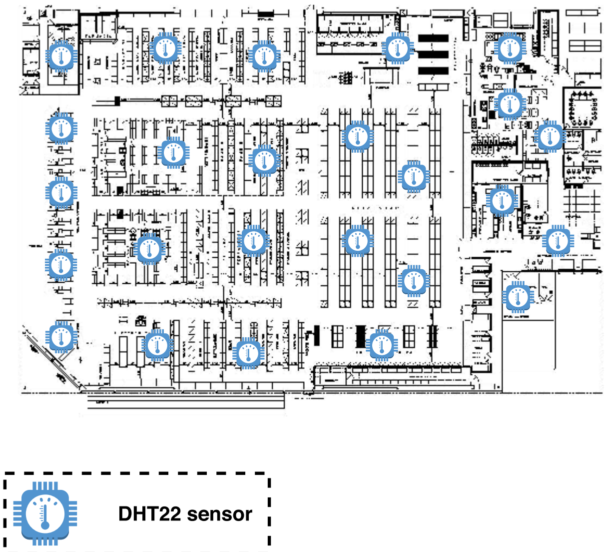

FIGURE 6. Map of the supermarket showing the distribution of the sensors, a sample of an indoor surface for testing our model and a sample of a sensor.

of the precision states (i.e., A,B,C) can move to a state of malfunction(F).In this figure, however, the sensors only degrade by one state; from state A to state B and from state B to state C. This is so, since in this example we assume that the sensor from any of its precision states can fail, while, we assume that a high accuracy sensor (A), has to go through the precise state (B) before moving to the low accuracy state (C).

Given the Markov chain used for this simulation with transition matrix given by the (50), the stationary paths given by the probabilities of change of precision state of the sensors are shown in Fig. 5. This figure illustrates the probability that a sensor's initial accuracy, state A, will shift to a different state in time t. Let's assume that $t_{max} = 5$ years (i.e., lifespan of the sensor is 5 years), then at $t = 0$, the probability that the sensor remain in state A is 1, while at $t \geq 0$ this probability decreases. Thus, the greater the value of t, the greater the probability that a sensor change to state B, C and F respectively. For $t \longrightarrow \infty$, the accuracy state F of the sensor has a probability of 1 (i.e., the sensor is in failure state) [21].

B. GENERAL DESCRIPTION OF THE EXPERIMENT

To test the proposed model, we have chosen a supermarket. At the time the IoT nodes measured the temperature, the thermostat of the supermarket showed 23°C. A mesh was used to place the sensors on the surface of the ground floor (Fig. 6). With the help of laser levels, the IoT nodes were placed vertically one in every section of the supermarket. A total of 25 IoT nodes were deployed.

The type of sensor deployed in the supermarket was a combination of the ESP8266 microcontroller in its commercial version "ESP-01" and a DHT22 temperature and humidity sensor (Fig. 6). The sum of both allows us for greater flexibility when collecting data and adaptability to the case study, since the DHT22 sensor is designed for indoor spaces (it has an operating range of 0°C to 50°C) according to its datasheet [http://www.micropik.com/PDF/dht22.pdf]. The microcontroller obtains data from this sensor through the onewire protocol and communicates it to the environment via Wi-Fi using HTTP standards and GET/POST requests. The ESP-IDF programming environment provided by the manufacturer of the microcontroller, was used to program the device.

The temperature sensors had been collecting data at 15 minute intervals, for an entire day. For the analysis we selected the data collected by the sensors in a time interval that began on 2018-10-17T09:00:00Z and ended on 2018-10-17T21:15:00Z. A specific moment was selected since the game that has been defined is static and not dynamic (i.e., it does not process the data in a temporal evolution). Table 4 provides a statistical summary of the measurements that were collected by the sensors.

TABLE 4. Statistical table of measurements of the IoT nodes.

Timestamp start	Total timestamp	Min temp	Max temp	Mean	Standard deviation
2018-10-17T09:00	11:45:00Z	20.1°C	24.6°C	22.91°C	0.71°C

In this experiment we have considered the next time interval $(t, t+\Delta t)$: $\frac{1}{365 \cdot 5}$ (i.e., a day). To validate the model we applied the accuracy state prediction model to the data collected by the sensors placed in the supermarket.

C. CASE STUDY RESULTS

In this case study, we have tested the proposed model, designed to increase the efficiency of monitoring and control of an IoT network. This is achieved by improving the quality of the data collected by the IoT nodes and the predicted maintenance of these nodes. In this way, the reliability of the data is increased and the energy efficiency of the smart supermarket is greater. The temperature collected by the IoT nodes in the input of the control algorithm.

Fig. 7, shows how the temperature evolves from its initial state (i.e., data collected by the IoT nodes) until the control algorithm sends the data to the process to set the regulators

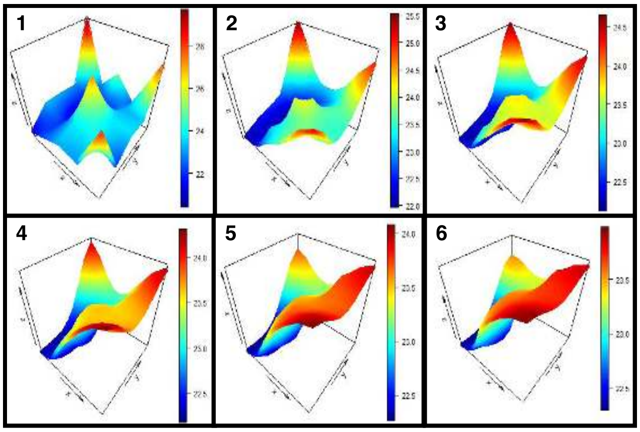

FIGURE 7. Graphic representation of the matrix of initial temperatures, the evolution of the temperature and the final temperatures obtained in this case study. Figure 4.1, gives the temperatures collected by the IoT nodes and the values that the control algorithm considers false data. Final temperatures, obtained after the final execution of the control algorithm, are provided in figure 4.6.

that control the temperatures of the supermarket sections. The supermarket temperature is slightly warmer in areas where there are large temperature differences. The control algorithm finds these zones and, if necessary, self-corrects these temperatures to reach the equilibrium in which the temperature is uniform in the whole supermarket.

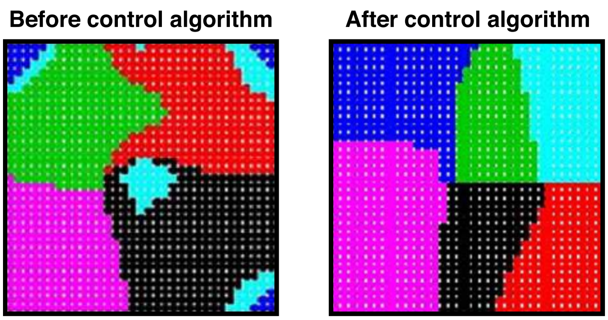

FIGURE 8. Data before applying the algorithm (left) and after applying the algorithm (right). In this figure it can be found that the final data is auto-corrected to eliminate the anomalous temperatures collected by the IoT network.

Fig. 8, shows the study of the temperature clusters that exist before and after the application of the control algorithm. The optimum number of clusters for the data collected by the IoT nodes is 6. On the left-hand side of the figure, it can be seen that the clusters are formed with the initial temperature. In some areas, the clusters are mixed indicating significant temperature variations. In turn, the right-hand side of the figure shows the clusters after the application of the control algorithm, it can be seen that the clusters are very different (i.e., no cluster has regions of other clusters inside it).

IV. CONCLUSIONS

This paper has addressed the problem of predictive control of accuracy in continuous-time NCSs. The feasibility of the proposed approach was verified with a case study in which the closed-loop system was modeled as a continuous-time feedback system with the Kolmogorov differential equations, improving the quality of the data collected by the IoT nodes. Through a newly constructed feedback control-based algorithm, an improved control system has been created. It allows to derive a smart supermarket's maximum allowable energy efficiency such that the resulting closed-loop system improves the control of an IoT network. A numerical case study illustrates the efficiency of our model. However, in many real scenarios, the ability to detect an imprecise or malfunctioning IoT node from a hot (cold) spot is limited. In a future work, we will try to solve this problem with artificial intelligence. Also, we will use blockchain technology and control algorithms to bring security and confidentiality to the data [16], [23].

REFERENCES

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.

[2] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *Proc. 49th IEEE Conf. Decision Control (CDC)*, Dec. 2010, pp. 5967–5972.

[3] F. Amato, M. Ariola, and C. Cosentino, "Finite-time stabilization via dynamic output feedback," *Automatica*, vol. 42, no. 2, pp. 337–342, 2006.

[4] F. Amato, M. Ariola, and C. Cosentino, "Finite-time control of discrete-time linear systems: Analysis and design conditions," *Automatica*, vol. 46, no. 5, pp. 919–924, May 2010.

[5] A. Polyakov, D. Efimov, and W. Perruquetti, "Robust stabilization of MIMO systems in finite/fixed time," *Int. J. Robust Nonlinear Control*, vol. 26, no. 1, pp. 69–90, 2016.

[6] V. I. Korobov, S. S. Pavlichkov, and W. H. Schmidt, "Global positional synthesis and stabilization in finite time of MIMO generalized triangular systems by means of the controllability function method," *J. Math. Sci.*, vol. 189, no. 5, pp. 795–804, 2013.

[7] Q. Hui, W. M. Haddad, and S. P. Bhat, "Finite-time semistability and consensus for nonlinear dynamical networks," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1887–1900, Sep. 2008.

[8] S. Khoo, L. Xie, S. Zhao, and Z. Man, "Multi-surface sliding control for fast finite-time leader–follower consensus with high order SISO uncertain nonlinear agents," *Int. J. Robust Nonlinear Control*, vol. 24, no. 16, pp. 2388–2404, 2014.

[9] Y. Li, S. Tong, and T. Li, "Composite adaptive fuzzy output feedback control design for uncertain nonlinear strict-feedback systems with input saturation," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2299–2308, Oct. 2015.

[10] Y.-X. Li and G.-H. Yang, "Event-triggered adaptive backstepping control for parametric strict-feedback nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 28, no. 3, pp. 976–1000, 2018.

[11] R. Casado-Vara, F. Prieto-Castrillo, and J. M. Corchado, "A game theory approach for cooperative control to improve data quality and false data detection in WSN," *Int. J. Robust Nonlinear Control*, vol. 28, no. 5, pp. 5087–5102, Nov. 2018.

[12] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Commun. ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[13] R. Y. Wang, "A product perspective on total data quality management," *Commun. ACM*, vol. 41, no. 2, pp. 58–65, 1998.

[14] R. Casado-Vara, P. Chamoso, F. De la Prieta, J. Prieto, and J. M. Corchado, "Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management," *Inf. Fusion*, 2019.

[15] K. Liu, A. Seuret, E. Fridman, and Y. Xia, "Improved stability conditions for discrete-time systems under dynamic network protocols," *Int. J. Robust Nonlinear Control*, vol. 28, no. 15, pp. 4479–4499, 2018.

[16] R. Casado-Vara, Z. Vale, J. Prieto, and J. M. Corchado, "Fault-tolerant temperature control algorithm for IoT networks in smart buildings," *Energies*, vol. 11, no. 2, p. 3430, 2018.

[17] X. Zhang and Y. Lin, "Adaptive output feedback tracking for a class of nonlinear systems," *Automatica*, vol. 48, no. 9, pp. 2372–2376, 2012.

[18] X. Zhang and Y. Lin, "Adaptive output feedback control for a class of large-scale nonlinear time-delay systems," *Automatica*, vol. 52, pp. 87–94, Feb. 2015.

[19] J.-X. Zhang and G.-H. Yang, ''Fault-tolerant leader-follower formation control of marine surface vessels with unknown dynamics and actuator faults,'' *Int. J. Robust Nonlinear Control*, vol. 28, no. 14, pp. 4188–4208, 2018.

[20] H. Wang, K. Liu, X. Liu, B. Chen, and C. Lin, ''Neural-based adaptive output-feedback control for a class of nonstrict-feedback stochastic nonlinear systems,'' *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1977–1987, Sep. 2015.

[21] T. Mailund, ''Continuous-time Markov chains,'' in *Domain-Specific Languages in R*. Berkeley, CA, USA: Apress, 2018, pp. 167–182.

[22] J. Kees, C. Berry, S. Burton, and K. Sheehan, ''An analysis of data quality: Professional panels, student subject pools, and Amazon's mechanical Turk,'' *J. Advertising*, vol. 46, no. 1, pp. 141–155, 2017.

[23] R. Casado-Vara, A. González-Briones, J. Prieto, and J. M. Corchado, ''Smart contract for monitoring and control of logistics activities: Pharmaceutical utilities case study,'' in *Proc. 13th Int. Conf. Soft Comput. Models Ind. Environ. Appl.* Cham, Switzerland: Springer, Jun. 2018, pp. 509–517.

[24] J. L. Doob, *Stochastic Processes*, vol. 7, no. 2. New York, NY, USA: Wiley, 1953.

ROBERTO CASADO-VARA (S'17) received the degree in mathematics from the University of Salamanca and the master's degree in big data and visual analytics from the International University of la Rioja. He is currently pursuing the Ph.D. degree in computer engineering with the BISITE Research Group. He is with Viewnext as a Data Scientist and is a Powercenter Consultant for important clients in the pharmaceutical and public administration sectors. He is a member with the BISITE Research Group. As a researcher, his interests include deep learning, advanced mathematical models for intelligent robust and non-linear control and monitoring, blockchain, and knowledge discovery data as well as other fields.

PAULO NOVAIS received the Ph.D. degree in computer sciences and the Habilitation degree in computer science (Agregação ramo do conhecimento em Informática) from the University of Minho, Braga, Portugal, in 2003 and 2011, respectively, where he is currently a Professor of computer science with the Department of Informatics, School of Engineering, and a Researcher with the ALGORITMI Centre, in which he is the Leader of the Research Group ISlab–Synthetic Intelligence, and a Coordinator of the research line Computer Science and Technology. He is the Director of the Ph.D. Program in Informatics and also the Co-Founder and the Deputy Director of the master in law and informatics with the University of Minho.

He started his career developing scientific research in the field of intelligent systems/artificial intelligence (AI), namely, in knowledge representation and reasoning, machine learning, and multi-agent systems. His interest, in the last years, was absorbed by different, yet closely related concepts of ambient intelligence, ambient assisted living, intelligent environments, behavioral analysis, conflict resolution, and the incorporation of AI methods and techniques in these fields. His main research aim is to make systems a little more smart, intelligent, and also reliable. He has led and participated in several research projects sponsored by Portuguese and European public and private institutions and has supervised several Ph.D. and M.Sc. students. He has co-authored over 300 book chapters, journal papers, and conference and workshop papers and books.

He is a member of the International Federation for Information Processing–TC 12 Artificial Intelligence and the Executive Committee of the Iberoamerican Society of Artificial Intelligence (IBERAMIA). He is a Senior Member of the Institute of Electrical and Electronics Engineers. He has been the President of the Portuguese Association for Artificial Intelligence (APPIA), since 2016, and a member of the Executive Committee of IBERAMIA. During the last years, he has served as an Expert/Reviewer for several institutions, such as EU Commission and FCT (Portuguese agency that supports science, technology, and innovation).

ANA BELÉN GIL received the degree in physics and the Ph.D. degree in computer science and automatics from the University of Salamanca, in 1997 and 2011, respectively, where she is currently with the Department of Computer Science and Automatics, and she is the Secretary of the Department. She is the Coordinator of the degree in computer engineering. She currently teaches both undergraduate and graduate courses, specifically the University Master's and Computer Engineering, where she teaches the advanced paradigms of human–computer interaction and information positioning, searches, and retrieval. Her research interests include artificial intelligence, human–computer interaction, recommender systems, and databases.

JAVIER PRIETO (S'11–M'12) received the Telecommunication Engineer degree, the Marketing Research and Techniques degree, and the Ph.D. degree in information and communication technologies from the University of Valladolid, Spain, in 2008, 2010, and 2012, respectively. Since 2007, he has been with different public and private research centers, such as the University of Valladolid and the Massachusetts Institute of Technology. He is currently a member and an Assistant Lecturer with the University of Salamanca. He has published more than 50 papers in international journals, books, and conferences and participated in more than 35 research projects, and is an Author of two Spanish patents. His research interests include social computing and artificial intelligence, for developing smarter and safer cities, localization and navigation technologies, for indoor and outdoor environments, and Bayesian inference techniques, for improving social welfare and environmental sustainability. He received the Extraordinary Performance Award for Doctorate Studies from the University of Valladolid, in 2012, and the Young entrepreneurs of Castilla y León Award from the Youth Institute of Castilla y León, Spain, in 2013. He serves as an Associate Editor for IEEE COMMUNICATIONS LETTERS and the *Wireless Communications and Mobile Computing Journal*, as a Guest Editor for the *International Journal of Distributed Sensor Networks*, and as a Reviewer for numerous international journals. He is a member of the Scientific Committee of the *Advances in Distributed Computing and Artificial Intelligence Journal*. He is a member of the Technical Program Committee of prestigious international conferences (such as IEEE GLOBECOM or IEEE ICC), and a Reviewer for a large number of them.

JUAN MANUEL CORCHADO (M'10) was born in Salamanca, Spain, in 1971. He received the Ph.D. degree in computer sciences from the University of Salamanca and the Ph.D. degree in artificial intelligence from the University of the West of Scotland. He was the Vice President for Research and Technology Transfer, from 2013 to 2017, and the Director of the Science Park with the University of Salamanca, where he was also the Director of the Doctoral School until 2017. He has been elected twice as the Dean of the Faculty of Science with the University of Salamanca. He has been a Visiting Professor with the Osaka Institute of Technology, since 2015, and a Visiting Professor with University Teknologi Malaysia, since 2017. He is currently a Full Professor with the Chair, University of Salamanca. He is a member of the Advisory Group on Online Terrorist Propaganda of the European Counter Terrorism Centre (EUROPOL). He is the Director of the Bioinformatics, Intelligent Systems and Educational Technology (BISITE) Research Group, which he created in 2000. He is the President of the IEEE Systems, Man and Cybernetics Spanish Chapter and the Academic Director of the Institute of Digital Art and Animation, University of Salamanca. He also oversees the master's programs in digital animation, security, mobile technology, community management and management for TIC Enterprises with the University of Salamanca. He is also an Editor and the Editor-in-Chief of specialized journals, such as the *Advances in Distributed Computing and Artificial Intelligence Journal*, the *International Journal of Digital Contents and Applications*, and the *Oriental Journal of Computer Science and Technology*.