Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

A cyber-physical system for building automation and control based on a distributed MPC with an efficient method for communication[☆]

Ali Karbasi, Alireza Farhadi[*]

Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 21 December 2018
Revised 23 April 2021
Accepted 30 April 2021
Available online xxx

Recommended by Prof. T. Parisini

Keywords:
Building automation and control system
Cyber-physical system
Building temperature regulation
Large-scale system

ABSTRACT

This paper introduces a cyber-physical system for building automation and control that is developed based on a distributed model predictive control. The implemented distributed method significantly reduces computation overhead with respect to the centralized methods. However, continuous data transfer between subsystems, which are often far from each other, is required when using this method. Information transmission between subsystems is very often subject to the limitations of transmission bandwidth and/or short communication range resulting in significant communication overhead. This causes significant time latency between making measurements and applying control commands, which adversely affects the control performance. Therefore, the distributed method used in this paper implements a two-level communication architecture to reduce the communication overhead. In order to avoid collision in communication inside neighborhood using this method, the TDMA-OFDMA scheme is used for wireless communication between distributed devices. Under these assumptions, the communication overhead is calculated. Then, a novel algorithm for finding the size of neighborhoods resulting in the lowest time latency between making measurements and applying control commands is presented for a typical office building. Finally, the satisfactory performance of the proposed cyber-physical system for the temperature control of a typical office building in the presence of disturbance and model inaccuracy is illustrated using computer simulations.

1. Introduction

1.1. Motivation and background

Large-scale Building Automation and Control Systems (BACSs) are used for controlling the environment of buildings, such as temperature by means of Fan Coil Units (FCUs), air conditioning, and ventilation. About half of the energy consumed in commercial buildings is directly related to space heating, cooling, and ventilation [9]. There are some commercial products available that are based on wired communication technologies for setting up BACSs. However, large-scale wireless BACSs, which are easy to deploy and cost-effective, have not been well studied yet. The advent of easily controllable smart thermostats and smart meters (with wireless communication and computation capabilities) and new wireless communication infrastructures has made it possible to reduce building energy consumption by designing a wireless cyber-physical system to operate the heating, cooling,and ventilation systems in a more efficient way to address the global energy and environmental concerns.

In recent years, advances in communication as well as sensor and actuator technologies enable us to develop cyber-physical systems. These systems are the new generation of Industrial Internet of Things (IIoT) systems. IIoT systems comprise of three different layers: Field, communication,and computation. The field layer consists of distributed sensors, actuators,and other networked devices. The communication layer consists of low cost and low power consumption Machine to Machine (M2M) IoT wireless communication modules, such as Digi XBee, LoRa, Sigfox. Each distributed field device is equipped with at least one M2M IoT module in an IIoT system. This module transmits data from the attached field device to a gateway that connects the M2M module to the Internet. The computation layer includes a computer server connected to the Internet. Data gathered from the field is received by this remote computer in real time; and subsequently, it generates proper commands to be sent to field devices via the communication layer by running the optimization, estimation, data mining,or the machine learning algorithms in real time. The ease of interconnection of distributed sensors and actuators and generally speaking the field

[☆] This work was supported by the research office of Sharif University of Technology.

[*] Corresponding author.
E-mail address: afarhadi@sharif.edu (A. Farhadi).

https://doi.org/10.1016/j.ejcon.2021.04.008
0947-3580/© 2021 European Control Association. Published by Elsevier Ltd. All rights reserved.

Nomenclature

BACSs	Building Automation and Control Systems
FCU	Fan Coil Unit
IIoT	Industrial Internet of Things
M2M	Machine to Machine
IoT	Internet of Things
TDMA	Time Division Multiple Access
OFDMA	Orthogonal Frequency Division Multiple Access
t	Time (s)
T_i	The temperature of node i (K)
Q	Thermal energy (J)
C_i	The thermal capacitance of node i ($\frac{\mathrm{J}}{\mathrm{K}}$)
R_{ij}	Thermal resistance between zone i and j ($\frac{\mathrm{k}}{\mathrm{W}}$)
h	Heat transfer coefficient ($\frac{\mathrm{W}}{\mathrm{m^2 k}}$)
A	Area (m^2)
L	Wall/roof thickness (m)
$\dot{Q}^i_{in}$	The source heat transfer rate to room i (W)
$\dot{m}_s{}^i$	The mass flow rate from the heater to room i ($\frac{\mathrm{kg}}{\mathrm{s}}$)
c_p	The specific heat capacity of air ($\frac{\mathrm{J}}{\mathrm{kg\ \ k}}$)
T^i_s	The supply air temperature of room i (k)
ρ	Density of air ($\frac{\mathrm{kg}}{\mathrm{m^3}}$)
$\dot{V}$	Volumetric flow rate ($\frac{\mathrm{m^3}}{\mathrm{s}}$)
L	Control horizon length
Q, R	Weighting matrices
N_v	The total number of building rooms
x_i	The state variable of the ith subsystem
u_i	The decision variable of the ith subsystem
ν_i	Interacting variable
LQ	Linear Quadratic
X_i	The closed convex subset of $\mathbb{R}^{n_i}$
$\mathcal{G}_i$	The closed convex subset of $\mathbb{R}^{m_i}$
x^d_i	The desired value for the ith state variable
S_i	The ith subsystem
$\mathcal{N}_i$	The ith neighborhood
n_j	The number of subsystems of the jth neighborhood
MAC	Medium Access Control
FFD	Full Function Device
RFD	Reduced Function Device
PAN	Personal Area Network
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
GTS	Guaranteed Time Slot
BI	Beacon Interval
SD	Superframe Duration
BO	Beacon Order
SO	Superframe Order
CAP	Contention Access Period
CFP	Contention Free Period
ZC	ZigBee Coordinator
ZR	ZigBee Router
ZED	ZigBee End Device
NSDU	Network Service Data Unit
NHR	Network Header
NPDU	Network Protocol Data Unit
MSDU	MAC Service Data Unit
MHR	MAC Header
MFR	MAC Footer
MPDU	MAC Protocol Data Unit
PSDU	Physical Service Data Unit
PHR	Physical Header
SHR	Synchronization Header
PPDU	Physical Protocol Data Unit
LIFS	Long Inter-Frame Spacing
SIFS	Short Inter-Frame Spacing
$\mathcal{B}$	The beacon frame transfer time
$\mathcal{D}$	The data frame transfer time without considering subsystems data
$\mathcal{C}$	The command frame transfer time
ACK	The acknowledgment frame transfer time
m_j	The size of the jth neighborhood
m_{max}	The size of the biggest neighborhood
m^*	The optimal size of neighborhood
n^j	The number of persons in the jth room

devices to IIoT systems results in larger and larger IIoT systems. As the size of IIoT systems increases, the available centralized algorithms, such as the active set method, the interior point method, the Kalman filter, etc., which the centralized computer server runs, are not able to complete their computation on time due to the huge computational complexity associated with large-scale IIoT systems. One way to overcome this computational scalability problem is to use the cloud computing concepts and exploit the available computational resources of the field. In IIoT systems, each distributed sensor, actuator,or networked device is equipped with at least one M2M IoT module, which is an embedded system equipped with a microprocessor/microcontroller. These distributed embedded systems provide us with one of the computational resources of the field. Having this computational resource, one way to overcome the aforementioned drawback is to break down the computational load of the centralized computer server to the distributed computational resources of the field and use parallel computation and consensus to complete the required centralized computation on time. In this type of IIoT systems, the computation layer is integrated to the field layer forming a new type of IIoT systems, which is known as cyber-physical systems. The main advantages of cyber-physical systems over IIoT systems are the scalability, higher reliability (due to its distributed nature), the ease of implementation, the lower cost of implementation and maintenance, etc. Therefore, the applications domain of cyber-physical systems is now very vast. It ranges from agricultural irrigation networks [19] to smart buildings [4,27,30,35] and power systems.

To illustrate the application of the cyber-physical system described above, let us consider controlling the temperature of an office with several rooms and hallways. One easy way for controlling the environment of this office as used in many places is as follows: In each room and hallway (referred to as subsystem), at least a thermostat which includes a sensor, a tuning volume, and an electrical relay is deployed to measure temperature, and a fan coil unit (actuator) is also installed to maintain the temperature close to the desired value. Moreover, the temperature control policy in each room and hallway is designed in a fully decentralized fashion without considering the temperature interaction between rooms and hallways, which from now on are referred to as subsystems. However, as internal doors are opened, the distribution of subsystems temperature changes, and hence, the implemented control policy becomes inefficient as it has been designed in a fully decentralized way. Now, suppose actuators and sensors in the setup proposed in this paper are equipped with wireless communication modules. Therefore; they can collaborate with each other for the temperature regulation of the entire building. To achieve this goal, a performance criterion for the entire building that penalizes both the energy consumption and the deviation of each subsystem temperature from the desired temperature is defined. For this setup, each decision-maker is defined as the collection of the thermostat and fan coil of each subsystem equipped with at least one M2M wireless communication module. In this setup, the decision-makers can be designed to minimize this performance criterion subject to the thermal dynamics of the building and operational constraints forming a wireless cyber-physical system for building automation and control. Two main methodologies have been developed to address this optimal control problem: One is based on the centralized optimization methodology, and the other one is based on the distributed optimization methodology. The conventional and widely used methodology is the centralized methodology, e.g., [31]. However, for large-scale buildings with many subsystems, the implementation of centralized methods requires high-performance computing devices with very fast processors and very large memory, which are beyond the available computational power and memory of each distributed decision-maker. Therefore, the distributed optimization methods have been introduced in the literature for developing cyber-physical systems by distributing the computational load to distributed decision-makers [19,32,36,45]. It has been shown that distributed methods are successful and superior with respect to centralized methods in many aspects, especially in terms of computational complexity [16,18] and reliability. Because of the above reasons, we implement the distributed optimization methods for distributing the computational load to the distributed decision-makers of the proposed cyber-physical system to tune actuators settings properly.

Wireless cyber-physical systems are very often subject to communication delays. Therefore, many research papers in this area are concerned with the analysis and compensation of communication delay effects in distributed networks, e.g., [12,17,50]. A hierarchical (two-level) communication architecture and a three-step algorithm including an extra outer iterate step are proposed in [19,44] to provide scope for managing the communication overhead, which is associated with the distributed optimization methods. In this distributed optimization method, distributed decision-makers are grouped into disjoint neighborhoods of nearby decision-makers; and the number of decision-makers within a neighborhood denotes the size of the neighborhood. The exchange of information between decision-makers within a neighborhood, which is subject to low communication overhead, frequently occurs after each decision variable update. In contrast, the exchange of information between neighborhoods, which is subject to high communication overhead, is limited to be less frequent. Hence, as most of the communication between distributed decision-makers is within neighborhoods (referred to as inside neighborhood communication), the communication overhead of the method of [19,44] is much less than that of the other distributed methods, e.g., [46], that uses a single – level communication architecture. This is shown in Section 3 by calculating the communication cost for inside neighborhood communication denoted by C_{com}^{in} and between neighborhood communication (referred to as all to all communication) denoted by C_{com}^{out}. In large-scale systems with many geographically distributed decision-makers, C_{com}^{out} is much larger than C_{com}^{in}. Therefore, in the distributed methods that use single-level communication architecture for communication, the communication overhead is extremely high as all the time, all to all communication takes place. Hence, in this paper, we use the distributed method of [19] that uses the two-level communication architecture; because from the communication point of view, it is efficient. In this method, each decision-maker within a neighborhood frequently updates its local component of the overall decision variable by solving an optimization problem of reduced size. The updated value is then communicated to all other neighboring decision-makers. This inside neighborhood update or communication is referred to as an inner iterate. In addition to inner iterates, updates of decision variables from other neighborhoods are received periodically. They are referred to as outer iterates. Between outer iterates, distributed decision-makers continue to compute and refine the local approximation of the optimal solution with fixed values for decision variables from outside the neighborhood. These inner-outer iterates continue until an ϵ-convergence stopping criterion is satisfied.

Due to the superiority of distributed optimization methods with respect to the conventional centralized methods, In the proposed cyber-physical system we use the distributed optimization method introduced in [19,44] for the temperature control of buildings by considering the effects of computational latency, which is the summation of the computation overhead and communication overhead. Note that the proofs for the feasibility, convergence, and the optimality of the distributed method used in this paper for the temperature control were previously presented in [19]. In the aforementioned reference, a novel distributed optimization method has been presented; and then a novel cyber-physical system, which is based on this optimization method, has been introduced for improving the performance of Australia's automated irrigation network. In sum, in the wireless cyber-physical system proposed in this paper for the temperature regulation of large-scale buildings, the field layer consists of distributed decision-makers. Each subsystem (room or hallway) is equipped with a decision-maker and a decision-maker is the collection of at least one thermostat and one fan coil unit equipped with one M2M IoT module. In this cyber-physical system, distributed M2M wireless IoT modules form the communication layer and the computation layer is integrated with the field layer using the distributed optimization method of [19].

1.2. Paper contributions

This paper introduces a cyber-physical system for building automation and control, as described above, which is developed based on the distributed optimization method of [19]. An application of this method for the temperature control of a building is presented. This method requires simultaneous communication within neighborhoods. Therefore, to avoid a collision in inside neighborhood communication using this method, the TDMA (time division multiple access)-OFDMA (orthogonal frequency division multiple access) scheme must be used for wireless communication between the distributed decision-makers. This type of communication can be easily implemented using M2M IoT modules. For example, the XBee module of Digi International provides us with this feature. Each XBee module is equipped with an embedded processor, which can also be used for distributed computation. Each subsystem of the building is equipped with a decision-maker, which includes at least a thermostat as well as a fan coil equipped with an M2M IoT module. Under this setup, the communication overhead is calculated. Then, a novel algorithm for finding the size of neighborhoods resulting in the lowest time latency between making measurements and applying control commands is presented for a typical office building. Finally, the satisfactory performance of the distributed model predictive control method, which is developed based on the distributed optimization method of [19], with minimum time latency for the temperature control of this building in the presence of disturbance and model inaccuracy, is illustrated by computer simulations. Distributed control methods for building automation and control have been investigated previously in [4,27,30,35,40]; however, unlike this paper, none of the available studies are concerned with the effects of communication and computation overheads on building automation performance, and they do not find the best trade-off between control performance, communication overhead and computation overhead for building automation.

1.3. Paper organization

This paper is organized as follows: the introduction was presented in Section 1. Section 2 describes building temperature control system by presenting the optimal control problem to be solved, the distributed optimization method introduced in [19], and the thermal model of a typical office building. Section 3 is devoted to the communication part. In Section 4, the optimal tradeoff between control performance, communication overhead and computation overhead are determined for a typical office building of interest by finding the optimal size of neighborhoods, denoted by m^* for this building. Using computer simulations, the satisfactory performance of the proposed wireless cyber-physical system with the optimal size of neighborhoods for the temperature control of this building in the presence of disturbance is illustrated in Section 5. Finally, the paper is concluded in Section 6.

2. Temperature control of building

This section describes the temperature control aspect of building automation and control system by presenting the thermal model of a typical office building, the optimal control problem to be solved, and the distributed optimization method introduced in [19].

2.1. Thermal model of a typical building

Extensive literature exists on modeling the conductive and convective heat transfer interaction between two rooms through the components separating them. The most common thermal modeling framework to model this interaction is based on a network of resistors and capacitors presented in [5,7,23,49]. For a typical large-scale office building, the number of nodes and edges for this RC network will be in the order of thousands [13,24]. In control applications, the model should be as simple as possible for large-scale systems and detailed enough to be reliable. Hence, most of the available computer-facilitated modeling tools, which are based on this RC modeling, are not suitable in this paper, as they result in complex models that cannot be readily used for large-scale control purposes. In recent years, a particular RC modeling for realistic buildings is proposed and used for designing control policy [6,31,41].

To develop such a simplified yet descriptive RC model, it is assumed that the pressure is constant throughout the building/system, and the amount of air exiting a zone (room or hallway) is equal to the air entering. The model used in this paper has the same dynamic characteristics as the model of [22], as described by the following simple example shown in Fig. 1.

Fig. 1 illustrates heat transfer in a room with two external walls (1 and 2), two internal walls (3 and 4), a floor, and a ceiling (5 and 6). $U_{i,j}$ ($\frac{\text{W}}{\text{K}}$) is thermal transmittance of the components; C_i ($\frac{\text{J}}{\text{K}}$) is the thermal capacitance of air and the components; T_i is the temperature of the components and indoor and T_o (K) is the temperature of outdoor air. The diagram illustrated in Fig. 1 is similar to the RC electrical circuit. The thermal capacity of a component is equivalent to the capacity of a capacitor, thermal resistance equivalent to electrical resistance, and temperatures are equivalent to voltages in an electrical circuit. The air temperature, T_i (K), inside the room is related to other parameters by the following differential equations [14]:

$$\begin{aligned}
&C_1\dot{T}_1 = (-U_{1,1} - U_{1,2})T_1 + U_{1,1}T_i + U_{1,2}T_o,\\
&C_2\dot{T}_2 = (-U_{2,1} - U_{2,2})T_2 + U_{2,1}T_i + U_{2,2}T_o,\\
&\dot{T}_{3,4} = -\beta_1 T_{3,4} + \frac{U_{3,2}}{C_3}U_{3,1}T_{z1} + \frac{U_{4,2}}{C_4}U_{4,1}T_{z2} + \left(\frac{U_{3,1}^2}{C_3} + \frac{U_{4,1}^2}{C_4}\right)T_i,\\
&\dot{T}_{5,6} = -\beta_2 T_{5,6} + \frac{U_{5,2}}{C_5}U_{5,1}T_{zc} + \frac{U_{6,2}}{C_6}U_{6,1}T_{zf} + \left(\frac{U_{5,1}^2}{C_5} + \frac{U_{6,1}^2}{C_6}\right)T_i,\\
&C_i\dot{T}_i = U_{1,1}T_1 + U_{2,1}T_2 + T_{3,4} + T_{5,6}\\
&\qquad + (-U_{1,1} - U_{2,1} - U_{3,1} - U_{4,1} - U_{5,1} - U_{6,1})T_i + \dot{Q}_s^i,
\end{aligned} \tag{1}$$

where

$$\begin{aligned}
&U_3 = U_{3,1} + U_{3,2},\ \ U_4 = U_{4,1} + U_{4,2},\\
&U_5 = U_{5,1} + U_{5,2},\ \ U_6 = U_{6,1} + U_{6,2},\\
&\frac{U_3}{C_3} = \frac{U_4}{C_4} = \frac{A_3 u}{\rho A_3 dc} = \frac{A_4 u}{\rho A_4 dc} = \frac{u}{\rho dc} = \beta_1,\\
&\frac{U_5}{C_5} = \frac{U_6}{C_6} = \beta_2,\\
&T_{3,4} = U_{3,1}T_3 + U_{4,1}T_4,\\
&T_{5,6} = U_{5,1}T_5 + U_{6,1}T_6,
\end{aligned} \tag{2}$$

where ρ ($\frac{\text{kg}}{\text{m}^3}$) is the density, A_i (m^2) is the area of walls, etc., d (m) is the thickness of walls, etc., c ($\frac{\text{J}}{\text{kgK}}$) is the specific heat capacity of wall, and u ($\frac{\text{W}}{\text{m}^2\text{K}}$) is the specific thermal transmittance. For a more comprehensive overview of the above formulation, refer to [14,22]. In the above equation, $\dot{Q}_s^i$ (W) is the heat transfer rate of the source to subsystem i; and it is obtained using the following expression [31]:

$$\dot{Q}_s^i = \dot{m}_s{}^i c_p (T_s^i - T_i), \tag{3}$$

where $\dot{m}_s{}^i$ ($\frac{\text{kg}}{\text{s}}$) is the mass flow rate from the heater to room i, c_p ($\frac{\text{J}}{\text{kg K}}$) is the specific heat capacity of air, and T_s^i (K) is the supply air temperature of room i. $\dot{m}_s{}^i$ is also given by

$$\dot{m}_s{}^i = \rho \dot{V}_s{}^i, \tag{4}$$

where ρ ($\frac{\text{kg}}{\text{m}^3}$) is the density of air and $\dot{V}$ ($\frac{\text{m}^3}{\text{s}}$) is the volumetric flow rate of the source. Therefore, $\dot{Q}_s^i$ is given by:

$$\dot{Q}_s^i = \rho \dot{V}_s{}^i c_p (T_s^i - T_i). \tag{5}$$

This substitution is done because $\dot{V}_s$ is easier to measure than $\dot{m}_s{}^i$. It is not practical to use $\dot{Q}_s^i$ as the input of the model; because the controller cannot act on the heat transfer rate directly. Therefore, in the heating system used in this paper, the volumetric airflow rate will be constant, and the supply air temperature is used as a decision variable. This is known as the Constant Air Volume (CAV) system. The CAV system can be realized, for example, using Fan Coil Units (FCUs) with constant air volume, where its coil temperature is adjusted by controlling the flow rate of hot water or refrigerant liquid passing through its coil. Obviously, when we set a new temperature for an FCU, it takes some time that its coil temperature reaches the desired temperature. However, as the dynamic between the coil desired temperature and the actual coil temperature is much faster than that of building temperature, throughout without loss of generality, we can consider the source temperature, T_s, as the decision variable when we use FCUs under the CAV system.

Note that in this paper, sol-air temperature is used instead of the outdoor temperature, T_o, in (1) in order to consider the heat flux through external building walls and roof, which consists of incident solar radiation, radiant heat exchange with the sky, and other outer surroundings, as well as convective heat transfer from the outdoor air. The sol-air temperature as defined in [20,26,43] is the temperature of the outdoor air, which in the absence of all radiation variations, gives the same heat transfer rate into the surface as the above variations (incident solar radiation, etc.) gives. Having that, in order to complete the thermal model of a

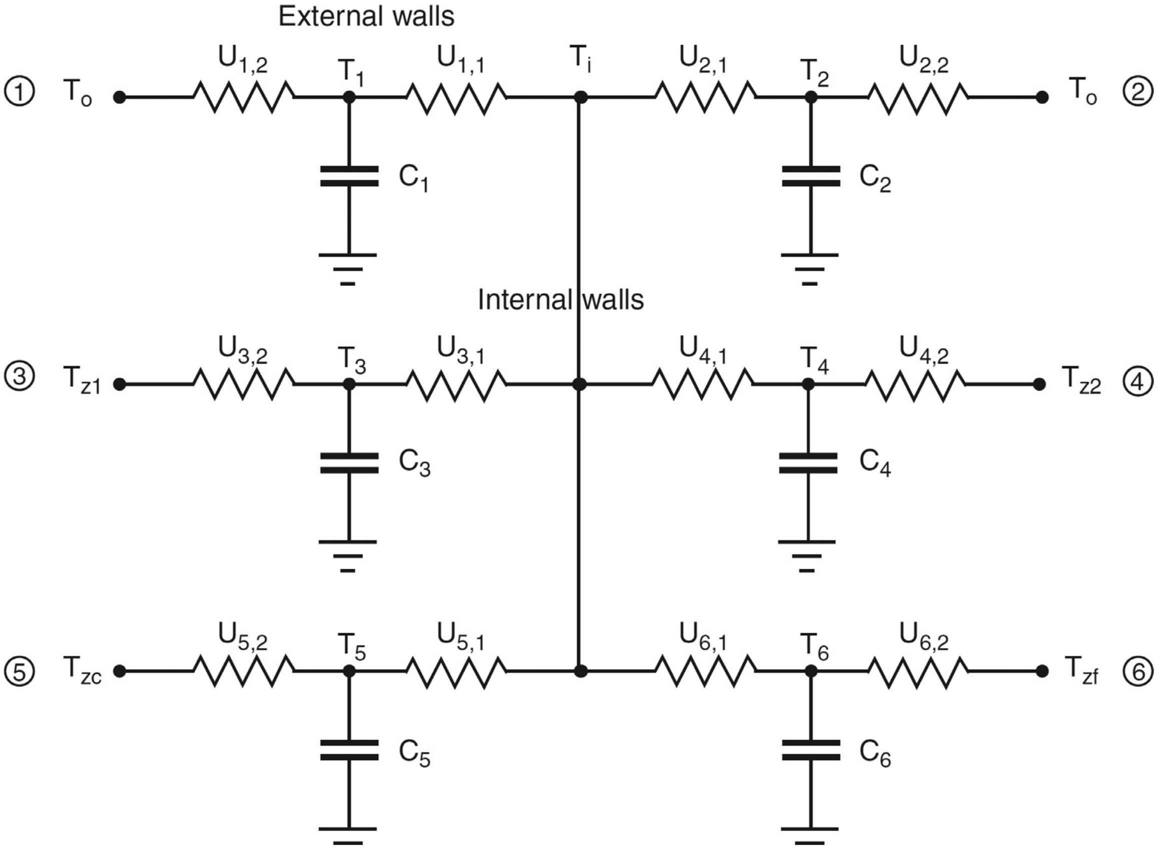

Fig. 1. The equivalent electrical model of the heat transfer in a room [14].

building, the sol-air temperature is calculated using the following expression.

$$T_{sa} = T_o + \frac{\alpha S}{h_o} - \frac{\xi \Delta R}{h_o}, \tag{6}$$

where T_{sa} (K) is the sol-air temperature, T_o (K) is the daily mean value of hourly outdoor air temperature, α is the absorption coefficient of the external surface for solar radiation, S ($\frac{\mathrm{W}}{\mathrm{m}^2}$) is the global daily mean value of hourly solar radiation incident on the surface, ξ is the emissivity of the external surface, ΔR ($\frac{\mathrm{W}}{\mathrm{m}^2}$) is the difference between the longwave radiation from the surroundings of the building (including the sky) and the radiation emitted by a black body at the outdoor air temperature, and h_o ($\frac{\mathrm{W}}{\mathrm{m}^2\mathrm{k}}$) is the outside heat transfer coefficient.

The above equation implies that the sol-air temperature is equivalent to the outdoor air temperature plus the impacts of the incident radiation absorbed by the outer surface minus the impacts of the emitted radiation to the sky and surroundings.

Note also that in order to consider the impacts of open doors and to calculate the heat transfer through them, the following equation borrowed from [3,33,39] is used for each door:

$$\dot{Q}^i_{OD} = \rho \dot{V}^i c_p (T_j - T_i), \tag{7}$$

where ρ ($\frac{\mathrm{kg}}{\mathrm{m}^3}$) is the density of air, $\dot{V}$ ($\frac{\mathrm{m}^3}{\mathrm{s}}$) is the volumetric flow rate between two rooms which is calculated by multiplying air velocity (0.1 ($\frac{\mathrm{m}}{\mathrm{s}}$) [42]) by the cross-section area of the door (m^2), and c_p ($\frac{\mathrm{J}}{\mathrm{kg\ K}}$) is the specific heat capacity of air. T_i and T_j are the air temperature of rooms i and j.

This paper is concerned with the temperature control of a typical office building with four floors and 98 rooms/hallways. As it is shown in Figs. 2–5, this building was simulated by DesignBuilder Software [8]. The thermal model of this building is obtained using the above-mentioned RC model and the software construction details for typical offices are used to obtain the thermal parameters of this model. The thermal model of this building has the following form:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx, \end{aligned} \tag{8}$$

where $x = [T_1, T_2, ..., T_{N_t}]'$ is the state vector that includes the temperatures of not only the rooms but also the internal nodes of the components. These nodes are indexed in such a way that the first $N_v = 98$ components of x correspond to the space temperature of rooms/hallways, and the remaining $N_t - N_v = 292$ states correspond to internal node temperatures of the components (walls, ceilings, roofs, etc.). The input vector $u = [T_s^1, T_s^2, ..., T_s^{N_v}, T_{sa}]'$ includes the supply air temperature of rooms and the sol-air temperatures. Note that only the first N_v components of the vector u include the decision variables. The elements of the matrices A and B are determined by the thermal capacitance and resistance of all the components of rooms/hallways and they are extracted from the DesignBuilder Software. Note that A is a 390×390 matrix, and B is a 390×103 matrix. Therefore, due to the huge dimensions of these matrices, we are not able to present them in the paper. Finally, C is chosen in such a way that $y = [T_1, T_2, ..., T_{N_v}]'$.

To avoid the unnecessary computational cost, in this paper, a suitable model reduction technique, as described below, is implemented on the above model. The model reduction technique applied in this paper is based on the balanced realization theory [29]. By using this technique, the most significant portion of the input-output relations of the actual model is preserved by the reduced model.

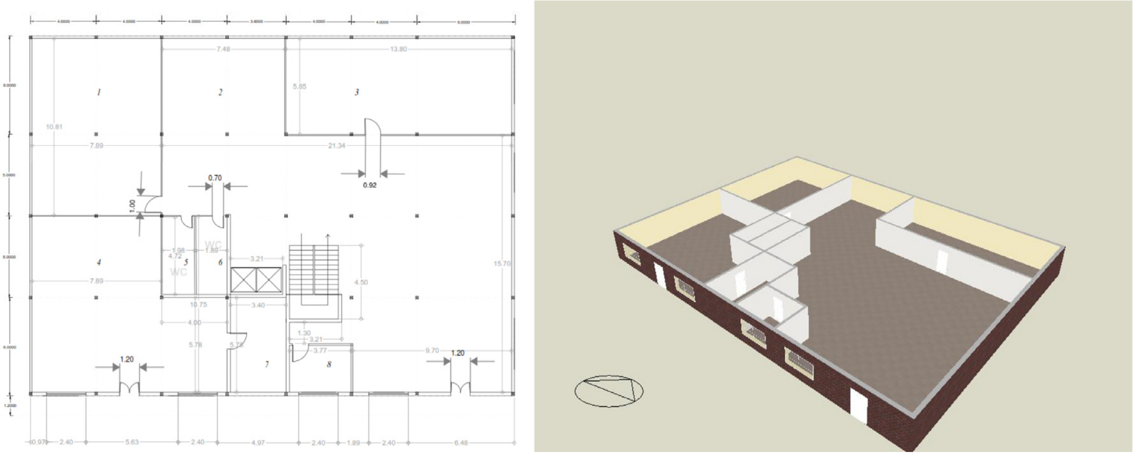

Fig. 2. Ground floor map of the office building.

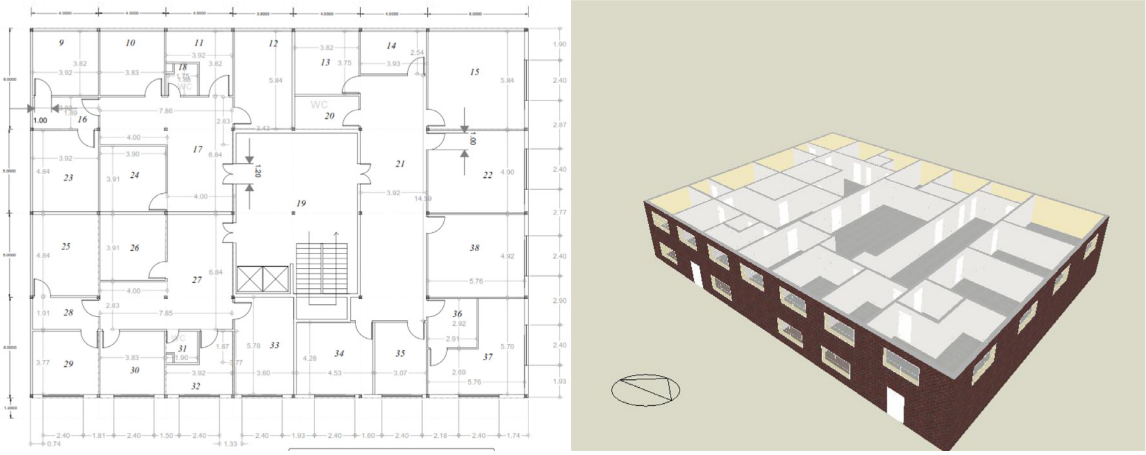

Fig. 3. First floor map of the office building.

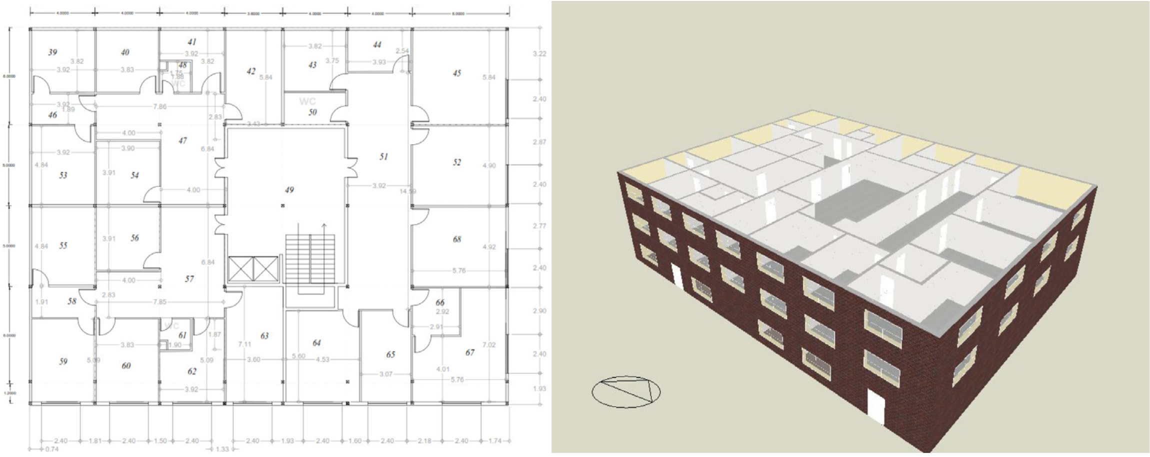

Fig. 4. Second floor map of the office building.

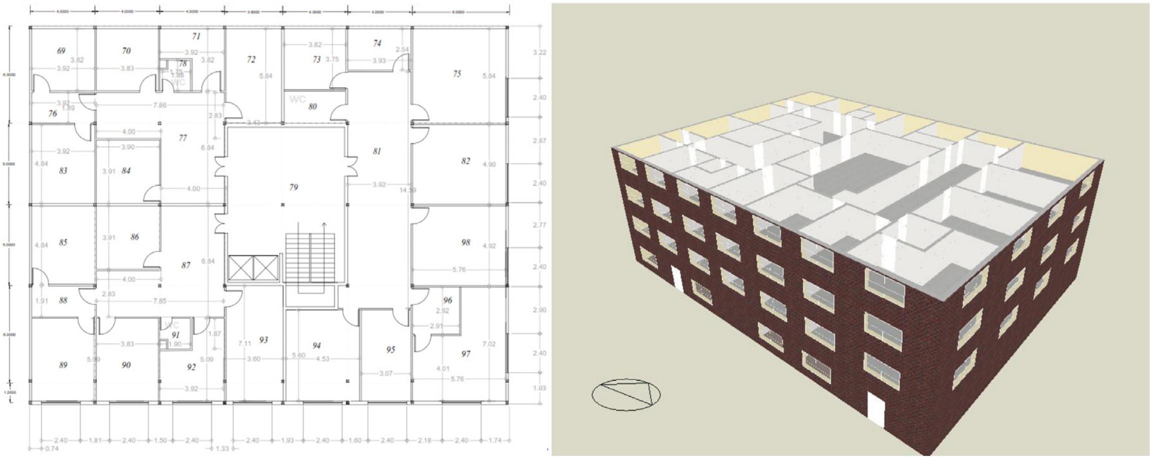

Fig. 5. Third floor map of the office building.

Consider the Nth order LTI model of (8) that is discretized with the sample time of 1 s:

$$x[k+1] = Ax[k] + Bu[k], \\ y[k] = Cx[k] \tag{9}$$

This system is open-loop stable and is balanced if its controllability Gramian (W_c) and observability Gramian (W_o) are identical and diagonal. It is known that there is an invertible state-space transformation matrix, T_b:

$$\bar{x} = T_b x, \tag{10}$$

such that

$$\begin{aligned} \bar{x}[k+1] &= T_b A T_b^{-1}\bar{x}[k] + T_b B u[k] \\ &= \bar{A}\bar{x}[k] + \bar{B}u[k], \\ y[k] &= C T_b^{-1}\bar{x}[k] = \bar{C}\bar{x}[k], \end{aligned} \tag{11}$$

$$\bar{W}_c = T_b W_c T_b^{'}, \\ \bar{W}_o = (T_b^{-1})' W_o T_b^{-1},$$

$$\bar{W}_c = \bar{W}_o = \Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n), \tag{12}$$

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0,$$

where σ_i is the Hankel singular value which is independent of the choice of T_b. The main idea of the balanced model reduction is that the states that correspond to larger singular values have more effect on the input-output behavior of the system; and therefore, they must be kept. Conversely, those correspond to smaller singular values have less effect on the input-output behavior of the system, and therefore, they can be removed. Hence, in the system of (11), which is in the balanced form [34], by setting a threshold for singular values (in this paper $\sigma_i \leq 0.1$), $\bar{x}$ can be divided into more important states ($\bar{x}_1 \in \mathbb{R}^{N_r}$) and less important states ($\bar{x}_2 \in \mathbb{R}^{N-N_r}$), as follows:

$$\begin{bmatrix} \bar{x}_1[k+1] \\ \bar{x}_2[k+1] \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} = \begin{bmatrix} \bar{x}_1[k] \\ \bar{x}_2[k] \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} u[k],$$

$$y[k] = \begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix} \begin{bmatrix} \bar{x}_1[k] \\ \bar{x}_2[k] \end{bmatrix},$$

By removing the less important portion of the model (that corresponds to $\bar{x}_2$), the reduced N_rth order system ($N_r = 98$) is derived, as follows:

$$\bar{x}_1[k+1] = \bar{A}_{11}\bar{x}_1[k] + \bar{B}_1 u[k]$$

$$y[k] = \bar{C}_1\bar{x}_1[k]. \tag{13}$$

The reduced system of (13) has most of the system features. However, its steady-state response is different from that of the original system. The authors of [25] suggested the following modification for maintaining the steady-state response, which is also used in this paper:

$$\begin{aligned} \bar{x}_1[k+1] &= (\bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21})\bar{x}_1[k] + (\bar{B}_1 - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{B}_2)u[k] \\ &= \hat{A}\bar{x}_1[k] + \hat{B}u[k], \\ y[k] &= (\bar{C}_1 - \bar{C}_2\bar{A}_{22}^{-1}\bar{A}_{21})\bar{x}_1[k] \\ &= \hat{C}\bar{x}_1[k] \end{aligned} \tag{14}$$

Remark 2.1. As clarified above, the original state-space representation of the system, i.e., Eq. (8) has 98+292 state variables, where after model reduction, only 98 states remain for designing the optimal control strategy. Note that as clarified, the balanced realization model reduction method is used in this paper. This method preserves input and output vectors. Because the output vector consists of the first 98 states of the state vector of Eq. (8), the number of state variables of the reduced model must be at least 98 states. These remaining state variables summarize the effects of 98+292 state variables of the original state-space representation. Note also that the state-space representation of Eq. (8) is used as the system dynamics for the simulation, and the reduced model (14) is used for designing the optimal control strategy.

Furthermore, in order to compensate for the effects of the computational latency, the time step for updating the control action, T, must be chosen much larger than the average computational latency. In this paper, this time step is chosen to be at least four times greater than the computational latency. Therefore, for designing the controller and obtaining optimal inputs, the reduced-order model of (14), which was discretized with the sample time of 1 s, is re-sampled with a new sample period of T seconds as follows:

$$\bar{x}_1[k+1] = \hat{A}^T\bar{x}_1[k] + \left(\sum_{j=0}^{T-1}\hat{A}^{T-j-1}\hat{B}\right)u[k]$$

$$y[k] = \hat{C}\bar{x}_1[k] \quad y[k] = y(kT), \quad \bar{x}_1[k] = \bar{x}_1[kT]. \tag{15}$$

As clarified in Remark 2.1, the state-space model of (15) is used for designing the controller. The control action is then applied to the state space representation of (8) whenever it is updated. It is generally assumed that the control action applies to the system as soon as the sensor measurements are provided for the controller. However, in practical scenarios, there may be a delay between making measurements and applying control commands due to computational latency for updating control action. Therefore, in order to consider the impacts of this delay into the system, the continuous-time system of (8) is discretized as follows:

$$\dot{x}(t) = Ax(t) + Bu(t),$$

$$x(t) = e^{At}x(0) + e^{At}\int_0^t e^{-A\gamma}Bu(\gamma)d\gamma,$$

Now from the above system response in the kth sample time, we have:

$$x[k] = x(kT) = e^{AkT}x[0] + e^{AkT}\int_0^{kT} e^{-A\gamma}Bu(\gamma)d\gamma. \tag{16}$$

Similarly, in the $k+1$th sample time, we have:

$$\begin{aligned} x[k+1] &= x((k+1)T) \\ &= e^{A(k+1)T}x[0] + e^{A(k+1)T}\int_0^{(k+1)T} e^{-A\gamma}Bu(\gamma)d\gamma, \end{aligned} \tag{17}$$

Now, by subtracting this equation from the Eq. (16) multiplied by e^{AT}, we have:

$$x[k+1] - e^{AT}x[k] = e^{A(k+1)T}\int_{kT}^{(k+1)T} e^{-A\gamma}Bu(\gamma)d\gamma, \tag{18}$$

Now, if the computational latency is T_d, then

$$\begin{aligned} x[k+1] = e^{AT}x[k] &+ e^{A(k+1)T}\int_{kT}^{(kT+T_d)} e^{-A\gamma}Bu_O d\gamma \\ &+ e^{A(k+1)T}\int_{(kT+T_d)}^{(k+1)T} e^{-A\gamma}Bu_N d\gamma, \end{aligned} \tag{19}$$

where

$$u(\gamma) = \begin{cases} u_O(\text{the previous control input}), & kT \le \gamma < kT + T_d \\ u_N(\text{the new control input}), & kT + T_d \le \gamma < (k+1)T \end{cases} \tag{20}$$

$$\Longrightarrow$$

$$\begin{aligned} x[k+1] = e^{AT}x[k] &+ Bu_O e^{A(k+1)T}\int_{kT}^{(kT+T_d)} e^{-A\gamma}d\gamma \\ &+ Bu_N e^{A(k+1)T}\int_{(kT+T_d)}^{(k+1)T} e^{-A\gamma}d\gamma. \end{aligned} \tag{21}$$

The above dynamic model is used to present the effects of the control actions for the temperature regulation of the office building.

2.2. Optimal control problem

In this paper, we are interested in the temperature control of a typical office building. In building temperature control, the energy use should be optimized, and at the same time, zones temperature should be maintained around desired values. To this end, an optimal control problem subject to thermal dynamics of the building and comfort constraints is formulated as follows:

$$\min_{(u_1,u_2,\ldots,u_{N_v})} \sum_{i=1}^{N_r}\sum_{k=0}^{L-1} ||y_i[k] - y_i^d||_Q^2 + ||u_i[k]||_R^2 \tag{22}$$

subject to thermal dynamics (15) and the following operational constraints:

$$u_{i,min} \le u_i[k] \le u_{i,max},\, y_{i,min} \le y_i[k] \le y_{i,max}. \tag{23}$$

In the above optimal control problem, y_i^d is the desired value for the temperature of the ith subsystem, L is the control horizon length, which is often long, and $Q \ge 0$ and $R > 0$ are weighting matrices. Similar quadratic cost functions for BACS have been considered in the literature (e.g., [31]).

The above optimal control problem is a constrained problem, and hence, the receding horizon idea (i.e., the model predictive control method) can be used to solve this problem in real time. That is, in each time step, a finite-horizon optimal control problem is formulated using the above problem and solved with the cost function J_k, where

$$J_k = \sum_{i=1}^{N_v}\sum_{j=k}^{k+N-1} ||y_i[j] - y_i^d||_Q^2 + ||u_i[j]||_R^2, \quad N << L. \tag{24}$$

This results in the optimal trajectories of inputs and states into the future satisfying the dynamics and constraints of the building while minimizing the quadratic cost function, J_k, over the future inputs within a window size of $N << L$, where only the first elements of the optimal trajectory of inputs are applied to the system. This procedure is then repeated for the next time steps. In terms of building temperature control, this means that at the present time instant, a heating/cooling plan is formulated for the next several hours to days, based on predictions of the upcoming conditions.

2.3. Distributed optimization method

In large-scale building automation and control systems, the total number of constraints and decision variables can be very large. This means, in many cases, that the computation overhead (the time spent computing the optimal solution) using the centralized optimization methods at each receding horizon may not be practical. A distributed optimization method exploiting a hierarchical (two-level) architecture for communication (see Fig. 6) and a three-step algorithm were introduced in [44] to overcome this computation scalability problem. In the mentioned reference, the authors did not provide any mathematical proofs for the feasibility, convergence, and optimality under the hierarchical exchange of updates. This gap was filled in [19]. The distributed optimization method of [19] is concerned with a system with N_v distributed interacting linear time-invariant subsystems each equipped with a decision-maker, denoted by S_i with the decision variable u_i, which has the following form:

$$\begin{aligned} &S_i : x_i[k+1] = A_i x_i[k] + B_i u_i[k] + v_i[k], \quad y_i[k] = C_i x_i[k], \\ &\quad i = 1, 2, \ldots, N_v, \quad k \in \{0, 1, 2, \ldots, N-1\}, \end{aligned} \tag{25}$$

where x_i, u_i and y_i are the state variable, the decision variable and the output of the ith subsystem, respectively, and v_i is the interacting variable that summarizes the effect of other subsystems on subsystem with the decision-maker S_i:

$$v_i[k] = \sum_{j=1, j\neq i}^{N_v} (M_{ij}x_j[k] + N_{ij}u_j[k]). \tag{26}$$

For this system, the following Linear-Quadratic (LQ) constrained optimization problem must be solved subject to the dynamics of subsystems and operational constraints, $x_i[k] \in \mathcal{X}_i$ and $u_i[k] \in \mathcal{G}_i$, where $\mathcal{X}_i$ is a closed convex subset of the real Euclidean space with the dimension of $n_i > 0(i.e., \mathcal{X}_i \subset \mathbb{R}^{n_i})$ modeling the constraint set on the ith state variable, and $\mathcal{G}_i$ is a closed convex subset of $\mathbb{R}^{m_i}$ modeling the time-invariant constraint set on the ith decision variable. That is, we are dealing with the following optimization problem:

$$\begin{aligned} &\min_u\{J_k(x[k], u_1, \ldots, u_{N_v}), \text{subject to (25) } x_i[j] \in \mathcal{X}_i, u_i[j] \in \mathcal{G}_i, \\ &\quad \forall i, j \in \{k, k+1, \ldots, k+N-1\}\}, \end{aligned} \tag{27}$$

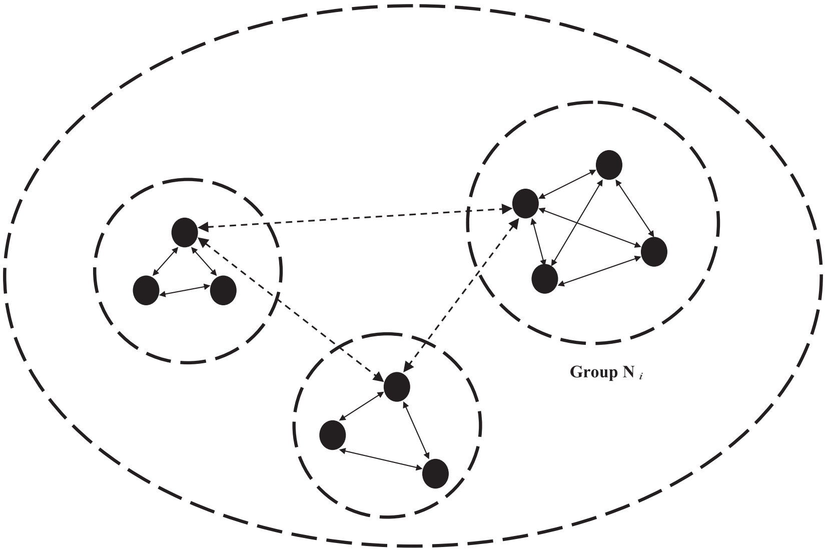

Fig. 6. Two-level architecture for exchanging information between distributed decision-makers. Solid circle represent distributed decision-makers, solid arrows between distributed decision-makers mean frequent update and communication and dashed arrows between neighborhoods mean less frequent update and communication.

$$J_k(x[k], u_1, ..., u_{N_v}) \doteq \sum_{i=1}^{N_v} \sum_{j=k}^{k+N-1} \|x_i[j] - x_i^d\|_Q^2 + \|u_i[j]\|_R^2 \tag{28}$$

where x_i^d is the desired value for the state variable (desired set point), and $Q \geq 0$ and $R > 0$ are weighting matrices. Note that without considering the remaining disturbance vector, the optimization problem that must be solved at each receding horizon for building temperature control of the previous section is of the form of the above optimization problem.

Since the centralized optimization methods are not sufficient to provide a feasible solution for a large-scale building automation system within a desired time period, in this paper, the distributed method of [19] is used to solve the above optimization problem. Towards solving this problem using the method proposed in [19], the system (25) with N_v distributed decision-makers: $S_1, S_2, \dots, S_{N_v}$, is decomposed into q disjoint neighborhoods: $\mathcal{N}_1, \dots, \mathcal{N}_q$, with specified sizes, as follows: $\mathcal{N}_1 = \{S_1, ..., S_{l_1}\}$, $\mathcal{N}_2 = \{S_{l_1+1}, ..., S_{l_2}\}$,...., $\mathcal{N}_q = \{S_{l_{q-1}+1}, ..., S_{N_v}\}$. Having that the distributed optimization method of [19] approximates the solution of the above optimization problem by taking the following steps:

- *Initialization:* The information exchange between neighborhoods at the outer iterate, $t \in \{0, 1, 2, ...\}$, makes it possible for every decision-maker, S_i, to initialize its local decision variable as $h_i^0 = u_i^t \in \mathbb{R}^{Nm_i}$, $i \in \{1, ..., N_v\}$, where $u_i^0 \in \mathcal{U}_i$ are chosen arbitrarily at $t = 0$. Note that $\mathcal{U}_i$ is the convex constraint set for the decision variable u_i resulted from the constraint sets $\mathcal{G}_i$ and X_i by considering the dynamic system mapping (25).

- *Inner iterate:* Between every two successive outer iterates, there are $\bar{p}$ inner iterates. Every decision-maker, S_i, of the neighborhood, $\mathcal{N}_e$ $(e = 1, 2, ..., q)$, performs $\bar{p}$ inner iterates simultaneously with other subsystems as follows:
  For each inner iterate, $p \in \{0, 1, ..., \bar{p} - 1\}$, decision-maker S_i first updates its decision variable via
  $$h_i^{p+1} = \pi_i h_i^* + (1 - \pi_i) h_i^p, \tag{29}$$
  where π_i are chosen subject to $\pi_i > 0, \sum_{j=1}^{l_1} \pi_j = 1, \dots \sum_{j=l_{q-1}+1}^{N_v} \pi_j = 1$ and $h_i^* \doteq \text{argmin}_{h_i \in \mathcal{U}_i} J_k(x[k], h_1^0, ..., h_{l_{e-1}}^0, h_{l_{e-1}+1}^p, ..., h_i, ..., h_{l_e}^p, h_{l_e+1}^0, ..., h_{N_v}^0)$ (note that $l_0 = 0$, $l_q = N_v$). Then, it trades its updated decision variable, h_i^{p+1}, with all other distributed decision-makers in its neighborhood, $\mathcal{N}_e$.

- *Outer iterate:* After $\bar{p}$ inner iterates, there is an outer iterate update as follows:
  $$u_i^{t+1} = \lambda_i h_i^{\bar{p}} + (1 - \lambda_i) u_i^t, \tag{30}$$
  where $u_i^t = \begin{bmatrix} u_i^{t'}[0] & u_i^{t'}[1] & . & . & . & u_i^{t'}[N-1] \end{bmatrix}' \in \mathbb{R}^{Nm_i}$, $u_i^t[j] \in \mathbb{R}^{m_i}$, $j = 0, 1, 2, ..., N-1$, and λ_i, $i = 1, 2, ..., N_v$, are chosen subject to $\lambda_i > 0$, $\lambda_1 = \cdots = \lambda_{l_1}$, $\lambda_{l_1+1} = \cdots = \lambda_{l_2}$, ..., $\lambda_{l_{q-1}+1} = \cdots = \lambda_{l_q} (\lambda_{l_q} = \lambda_{N_v})$, $\lambda_{l_1} + \lambda_{l_2} + \cdots + \lambda_{l_q} = 1$. Then, there is an outer iterate communication in which the updated decision variables, u_i^{t+1}, are shared between all neighborhoods, and subsequently between all distributed decision-makers.

It is shown [19] that by increasing t, $(u_1^t, u_2^t, ..., u_{N_v}^t)$ converges to $(u_1^*, u_2^*, ..., u_{N_v}^*)$, which is the optimal solution of the above optimization problem. Hence, for a large enough t (denoted by T_ϵ), $(u_1^{T_\epsilon}, u_2^{T_\epsilon}, ..., u_{N_v}^{T_\epsilon})$ is a good approximation of the optimal solution.

We now define communication overhead, computation overhead, and computational latency of the above three-step algorithm.

Definition 2.2. (Communication overhead) Communication overhead is defined as the total time spent for exchanging information between distributed decision-makers to approximate the solution of the above optimization problem by $u_i^{T_\epsilon}$, where for a priori fixed precision ϵ, T_ϵ is the smallest integer that satisfies the following inequality:

$$|J_k(x[k], u_1^t, ..., u_{N_v}^t) - J_k(x[k], u_1^{t+1}, ..., u_{N_v}^{t+1})| \leq \epsilon, \quad t \geq T_\epsilon.$$

Definition 2.3. (Computation overhead) At a given outer iteration, the neighborhood with the longest processing time for $\bar{p}$ inner iterations is defined as the dominating neighborhood at this iteration, and accordingly, the computational overhead is defined as the summation of the processing times of the dominated neighborhoods for approximating the solution of the optimization problem by $u_i^{T_\epsilon}$.

Definition 2.4. (Computational latency) Computational latency is the summation of the communication overhead and computation overhead.

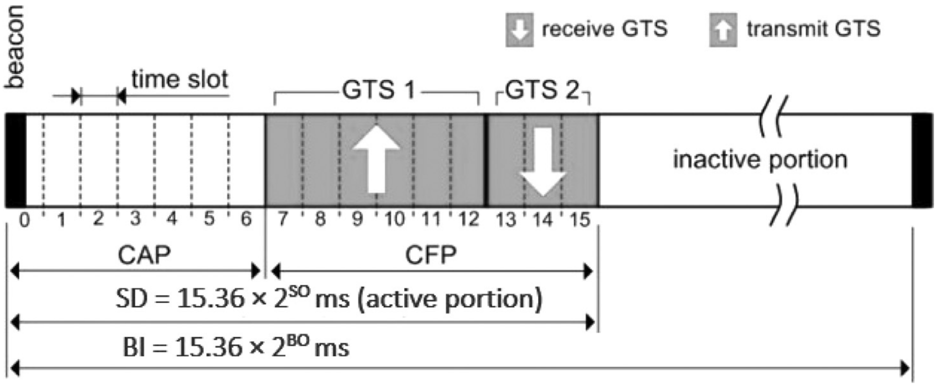

Fig. 7. Superframe structure [47].

The feasibility, convergence, and optimality of the above three-step method have been proven in [19].

3. Calculation of the communication overhead

For communication purposes, this paper focuses on IEEE 802.15.4/ZigBee communication technology. IEEE 802.15.4 standard and other specifications of ZigBee illustrate outstanding communication technologies for large-scale, low data rate, low cost, low power consumption, and simple operation wireless networks [38]. Note that XBee modules of Digi international have been developed based on this standard.

The Standard supports two medium access modes that can be selected by the Personal Area Network (PAN) Coordinator: 1. The non-beacon-enabled mode 2. The beacon-enabled mode. In non-beacon-enabled mode, the nodes can simply dispatch messages using unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) channel access protocol. As a matter of fact, the “collision avoidance” mechanism is based on a random delay prior to transmission, which only decreases the possibility of collisions. Therefore, this mode cannot ensure collision-free and predictable access to the shared wireless medium, and, as a result, it is not able to provide any time and resource guarantees. More details about the non-beacon-enabled mode can be found in [10].

Now, in the beacon-enabled mode, beacon frames are periodically sent by the coordinator to synchronize nodes that are associated with them and to recognize the PAN. In addition to transmission based on CSMA/CA, the beacon-enabled mode provides a contention-free Guaranteed Time Slot (GTS) mechanism to support time-critical data transmissions as well as collision-free and predictable access to the wireless medium.

This paper focuses on IEEE 802.15.4 beacon-enabled mode. Fig. 7 shows a superframe structure approved by IEEE 802.15.4 beacon-enabled mode. A superframe starts with a beacon (shown in the dark) issued by a PAN coordinator, containing an active portion and an inactive portion. The time between two consecutive beacon frames is called the Beacon Interval (BI). The Superframe Duration (SD) describes the active portion of the BI and is divided into 16 equal time slots, during which the coordinator and devices can communicate with each other. An inactive period may be optionally defined if BI $>$ SD. During the inactive period (if it exists), all nodes may enter a sleep and low-power mode (to save energy). However, in the communication technique used in this paper, the inactive period is considered to be the time duration during which nodes perform their process to derive the optimal solution. The GTS reservation scheme resembles the Time Division Multiple Access (TDMA) scheme in terms of time slot reservation. However, for a TDMA, time is partitioned into frames of a fixed duration, and each frame is partitioned into a fixed number of time slots. Nevertheless, the Guaranteed Time Slot (GTS) in IEEE 802.15.4 does not have fixed duration but rather has tunable duration by beacon parameters, and it is for dynamic time synchronization. BI and SD are specified by two parameters, the Beacon Order (BO) and the Superframe Order (SO), respectively, as follows:

For integer numbers, SO and BO such that $0 \leq SO \leq BO \leq 14$:

$$BI = aBaseSuperFrameDuration \times 2^{BO} = 15.36 \times 2^{BO}\ \text{ms},$$
$$SD = aBaseSuperFrameDuration \times 2^{SO} = 15.36 \times 2^{SO}\ \text{ms},$$

In the SD, nodes compete for medium access using Slotted CSMA/CA during the Contention-Access Period (CAP). MAC commands are transmitted in the CAP. The GTS mechanism permits a device to access the medium without contention during the Contention Free Period (CFP). The GTS is assigned by the coordinator and used only for data transmission between the coordinator and a device. This assignment of the GTS cannot decrease the length of the CAP to less than aMinCAPLength (7.04 ms) constant [10] to ensure that MAC commands can still be transmitted when GTSs are being used. The CFP supports up to seven GTSs, and a single GTS may spread over one or more time slots. Each GTS has only one orientation: From the device to the coordinator or from the coordinator to the device. A device to which a GTS has been allocated can also communicate during the CAP. The coordinator is responsible for implementing the GTS management.

Recall that in this paper, the ZigBee module is used for communication. Three types of devices are defined in ZigBee specification: ZigBee Coordinator (ZC), ZigBee Router (ZR), and ZigBee End Device (ZED). A ZC is an FFD (full function device), and there is only one for each ZigBee Network capable of initiating and configuring the network formation, acting as a PAN coordinator. Furthermore, any ZC can act as a ZR once the network is formed. A ZR is an FFD, associated with a ZC or with a formerly associated ZR, and takes part in multi-hop routing of messages. A ZED is any FFD or RFD (reduced function device) that does not allow other devices to associate with it and does not take part in routing.

Three network topologies are supported in IEEE 802.15.4/ZigBee: Star, Mesh, and Cluster-tree. In all cases, a unique node operates as a ZC. The ZC picks out a PAN identifier, which must not be used by any other ZigBee network in the neighborhood.

The Cluster-tree topology (Fig. 8c) is a special case that enjoys the benefits of both Star and Mesh networks, such as good scalability, network synchronization, and predictable connectivity, which is suitable for large-scale time-sensitive applications and when there is a distributed synchronization mechanism (beacon-enabled mode). The advantage of this synchronization with periodic beacon frame transmissions from the ZCs is that nodes can transmit their data at specific time slots without collision with data from other nodes. Note that the beacon-enabled mode is not

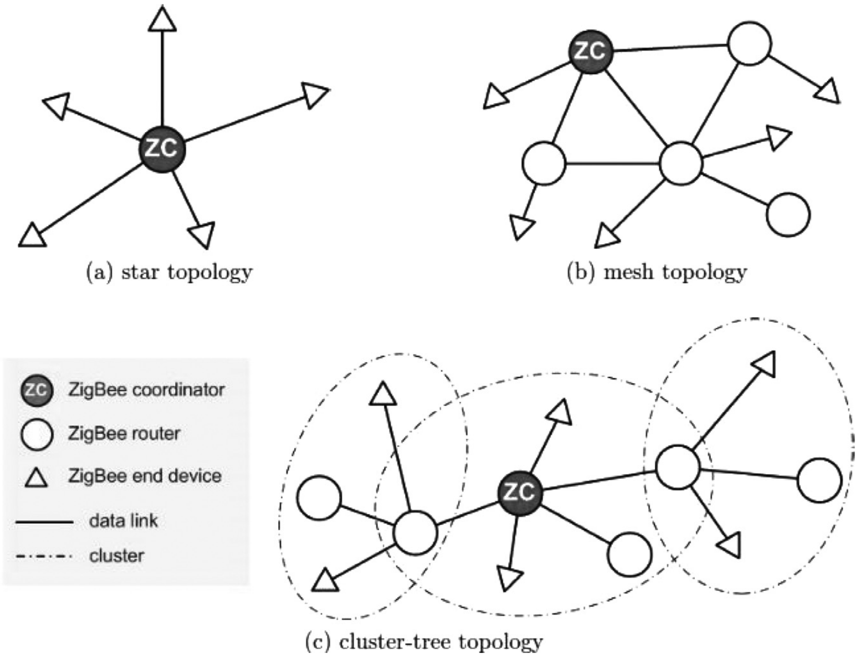

Fig. 8. Star, mesh, and cluster-tree topology examples [47].

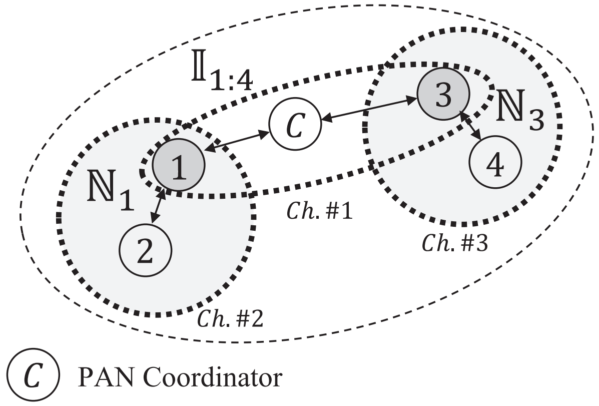

Fig. 9. Two-level architecture for exchanging information between distributed decision-makers using ZigBee modules.

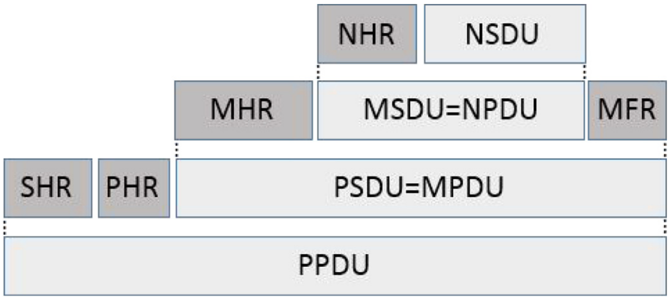

Fig. 10. Structure of IEEE 802.15.4/ZigBee frames [15].

Table 1
Comparison of star, mesh, and cluster-tree topologies.

	Star	Mesh	Cluster-Tree
Scalability	No	Yes	Yes
Network synchronization	Yes	No	Yes
Deterministic routing	Yes	No	Yes
Contention free medium access	Yes	No	Yes

permitted in Mesh networks. The important aspects of the above three topologies are summarized in Table 1.

Fig. 9 illustrates the two-level architecture for exchanging data between distributed decision-makers of the distributed optimization method presented in the previous section using ZigBee modules. Note that each ZigBee module can use 16 different communication channels with orthogonal carrier frequencies for exchanging data. Each subsystem is equipped with a ZigBee module, and to avoid inter-cluster collisions, all ZigBee modules (nodes) in a neighborhood use the same communication channel with specific carrier frequency different from neighboring neighborhoods carrier frequencies [1,28,48]. In each neighborhood, the nodes are organized in logical groups called clusters. Each ZR or ZC can organize a cluster and be a cluster-head such that all other nodes in the cluster are within its effective communication range. Communication between cluster-heads for outer iterate communication is via another ZigBee module known as PAN coordinator. PAN coordinator and cluster-heads use a communication channel with a carrier frequency that is different from clusters carrier frequencies used for inner neighborhoods communication.

Any frame communicated in the network contains several parts. Fig. 10 illustrates how the data payload is generated from the application layer to the physical layer. On the top, the payload is known as Network Service Data Unit (NSDU) which is prefixed with a network header (NHR). Together, NHR and NSDU constitute the Network Protocol Data Unit (NPDU), which is passed to the MAC sub-layer as the MAC frame payload (MAC Service Data Unit (MSDU)). The MAC payload, prefixed with a MAC Header (MHR) and appended with a MAC Footer (MFR), forms the MAC Protocol Data Unit (MPDU). The MPDU is passed to the physical layer as the PHY payload (Physical Service Data Unit (PSDU)). The PHY payload, prefixed with a Physical Header (PHR) and a Synchronization Header (SHR), forms the Physical Protocol Data Unit (PPDU), which can be dispatched to a wireless channel [2,47].

Note that in the distributed optimization control method of Section 2.3, the payload data are u_i^0, $x_i[k]$, h_i^{p+1}, and u_i^{t+1}, in which four bytes are allocated for exchanging them. To initialize the

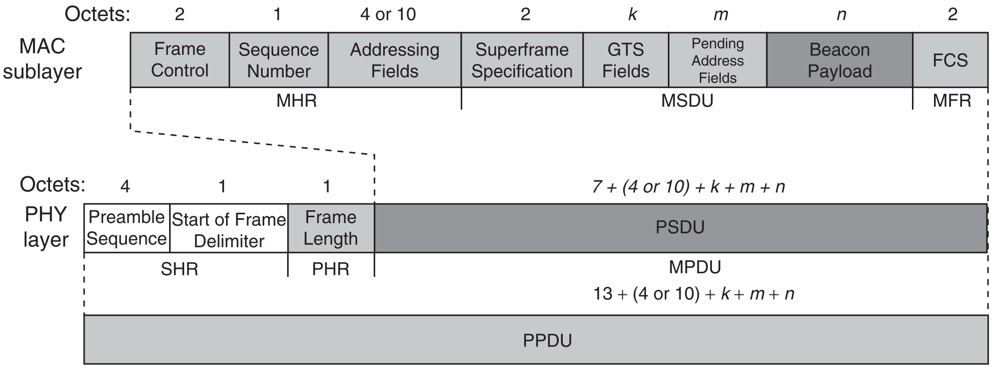

Fig. 11. MAC beacon frame format [11].

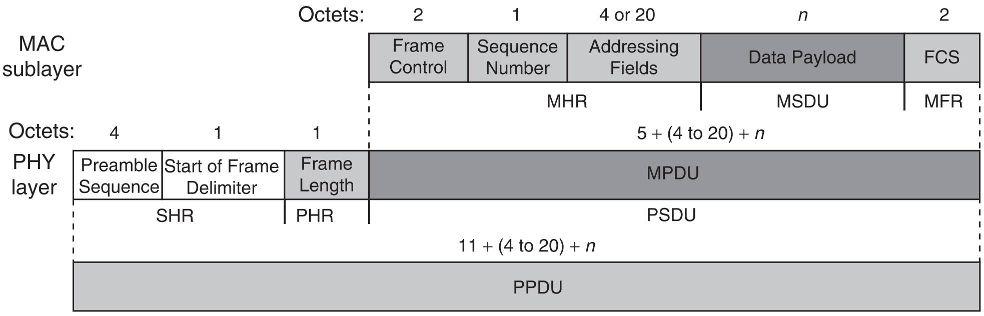

Fig. 12. Data frame format [11].

distributed optimal control method, at the beginning of the sample time, k, it is required that u_i^0 and $x_i[k]$ be exchanged between all distributed decision-makers, and to this end, an inner iterate communication followed by an outer iterate communication is required for exchanging u_i^0 and $x_i[k]$. For other iterations, only exchanging h_i^{p+1} (in inner iterates) or u_i^{t+1} (in outer iterates) is required.

Note also that each cluster can be seen as a Star sub-network, and the cluster-head of a neighborhood periodically sends a beacon frame to other nodes of its neighborhood to identify the time slots for receiving data from other nodes. Note that the Beacon frame is a frame with an MPDU of Fig. 11.

If a node is ready for sending data to its cluster-head, it sends a GTS request command (i.e., a MAC command with MPDU that is illustrated in Fig. 14) to its cluster-head, to indicate its GTS characteristics according to the sending time requirement that receives and replies it with an acknowledgment (an acknowledgment frame is illustrated in Fig. 13). In the next beacon frame, cluster-head specifies the configuration parameters of each allocated GTS, i.e., GTS device, GTS direction, GTS length, and GTS starting slot, and each node sends its information data by data frame with the MPDU structure shown in Fig. 12 at the allocated time slot to the cluster-head. When cluster-head receives all information data from other nodes, it broadcasts its information data to other nodes; and in this way, inner iterate communication is completed.

For the outer iterate, this procedure is repeated, but this time between cluster-heads and PAN coordinator.

Note that the consecutive frames, as shown in Fig. 15, are separated by a finite amount of time as shown in Fig. 16, where Long Inter-Frame Spacing (LIFS) and Short Inter-Frame Spacing (SIFS) are the amount of time that is needed by the MAC sublayer to process data received. SIFS is a duration of at least 0.192 ms for the frames with an MPDU size of smaller than 18 bytes, and LIFS is a duration of at least 0.64 ms for the frames with an MPDU size of more than 18 bytes. The cluster-head can begin transmission of acknowledgment frame after at least a Turn Around Time = 0.192 ms because each device, to receive an acknowledgment, needs a Turn Around Time to change its radio from the TX (transmission) to RX (receive) mode.

The bandwidth of ZigBee is equal to $250\frac{\text{kb}}{\text{s}} = 250 \times 10^3 \frac{\text{bits}}{\text{s}}$. Knowing that, the different frame transfer time can be calculated. The beacon frame transfer time (considering $k = 23, m = 14, n = 22$ in Fig. 11) is:

$$\mathcal{B} = \frac{(6 + 11 + 23 + 14 + 22) \times 8}{250 \times 10^3} = 2.432 \text{ ms}.$$

The data frame transfer time without considering subsystems data (considering $n = 16$ in Fig. 12) is:

$$\mathcal{D} = \frac{(6 + 9 + 16) \times 8}{250 \times 10^3} = 1.120 \text{ ms}.$$

Also, data payload contains subsystems data, and each sub-system data is 64 bits with the transfer time of $\mathcal{L}_0 = \frac{64}{250\times10^3} = 0.256$ ms for the first data exchange, and 32 bits with the transfer time of $\mathcal{L} = \frac{32}{250\times10^3} = 0.128$ ms for the rest of data exchanges. The acknowledgment frame transfer time is:

$$\text{ACK} = \frac{(6 + 5) \times 8}{250 \times 10^3} = 0.352 \text{ ms}.$$

The command frame transfer time (considering $n = 17$ in Fig. 14) is:

$$\mathcal{C} = \frac{(6 + 10 + 17) \times 8}{250 \times 10^3} = 1.056 \text{ ms}.$$

Now having that, we are ready to calculate the communication overhead. Recall that the communication overhead, C_{com}, is the total time spent for exchanging information between subsystems/decision-makers to have an approximation of the optimal solution up to a priori fixed precision, ϵ; and T_ϵ is the smallest integer such that the following inequality holds for all $t \geq T_\epsilon$:

$$|J_k(x[k], u_1^t, \ldots, u_{N_\nu}^t) - J_k(x[k], u_1^{t+1}, \ldots, u_{N_\nu}^{t+1})| \leq \epsilon, \quad \forall t \geq T_\epsilon.$$

Note that the total number of outer iterates for this approximation is $1 + T_\epsilon$.

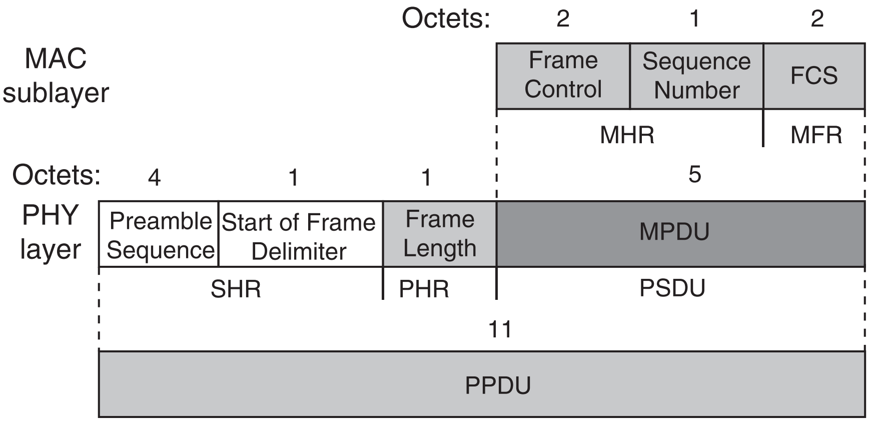

Fig. 13. MAC acknowledgment frame format [11].

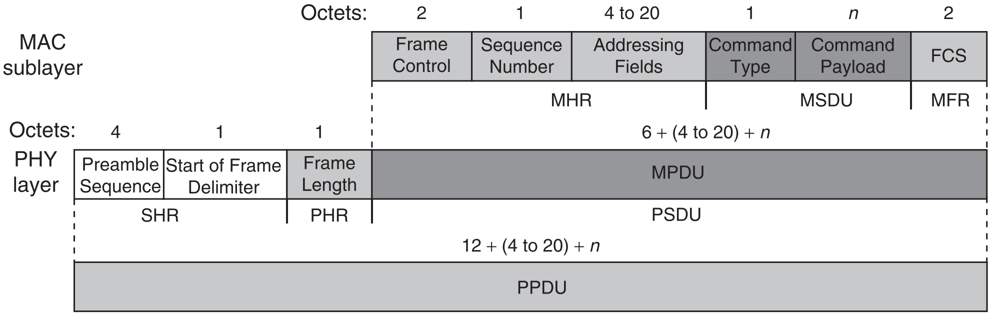

Fig. 14. MAC command frame format [11].

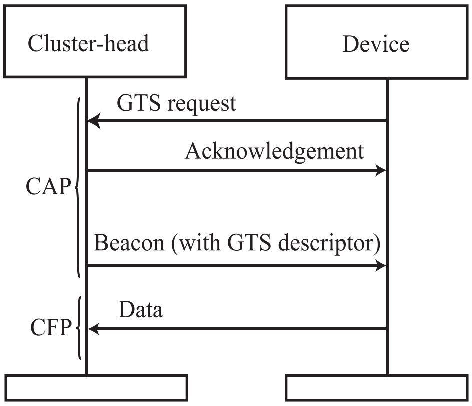

Fig. 15. The transfer of frames during the CAP and CFP [37].

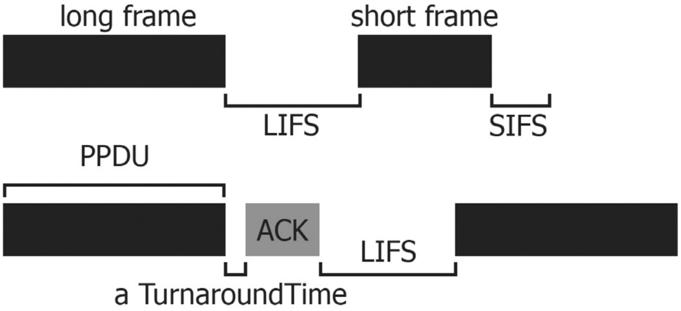

Fig. 16. Time delay between two consecutive frames [47].

Hence, in each inner iterate, there are two flows of information in the jth cluster:
i) From nodes to cluster-heads, in which for the inner iterate communication associated with the first outer iterate, it takes at least $\mathcal{B}+(m_j-1)((\mathcal{C}+aTurnaroundTime+\mathcal{A}+LIFS)+(\mathcal{D}+\mathcal{L}_0+LIFS))+n_{si}\times aMinCAPLength$ milliseconds; and for the rest of the inner iterates communication, it takes $\mathcal{B}+(m_j-1)((\mathcal{C}+aTurnaroundTime+\mathcal{A}+LIFS)+(\mathcal{D}+\mathcal{L}+LIFS))+n_{si}\times aMinCAPLength$ milliseconds for sending h_i^{p+1}s of the jth cluster to its cluster-head.

ii) From cluster-heads to other nodes, which takes at least $n_{si}\times aMinCAPLength+(m_j-1)((m_j-1)(\mathcal{D}+\mathcal{L}_0+LIFS))$ milliseconds for the inner iterate associated with the first outer iterate communication; and $n_{si}\times aMinCAPLength+(m_j-1)((m_j-1)(\mathcal{D}+\mathcal{L}+LIFS))$ milliseconds for the rest of the iterations.

Similarly, for the outer iterate communication, there are two flows of information:
i) From cluster-heads to PAN coordinator, which takes $\mathcal{B}+q(\mathcal{C}+aTurnaroundTime+\mathcal{A}+LIFS)+n_{so}\times aMinCAPLength+\sum_{j=1}^{q}(\mathcal{D}+m_j\times\mathcal{L}_0+LIFS)$ milliseconds in the first outer iterate; and $\mathcal{B}+q(\mathcal{C}+aTurnaroundTime+\mathcal{A}+LIFS)+n_{so}\times aMinCAPLength+\sum_{j=1}^{q}(\mathcal{D}+m_j\times\mathcal{L}+LIFS)$ milliseconds for the rest of the iterations.

ii) From PAN coordinator to cluster-heads, which takes $n_{so}\times aMinCAPLength+\sum_{j=1}^{q}((N_v-m_j)\times\mathcal{L}_0+(q-1)\mathcal{D}+(q-1)LIFS)$ milliseconds for the first outer iterate and $n_{so}\times aMinCAPLength+\sum_{j=1}^{q}((N_v-m_j)\times\mathcal{L}+(q-1)\mathcal{D}+(q-1)LIFS)$ milliseconds for the rest of the outer iterates.

Now, knowing that the total number of inner iterates communication is $1+\bar{p}T_\epsilon$, the total number of outer iterates communication is $1+T_\epsilon$, and the outer iterate occurs when all clusters finish their inner iterates, the communication overhead is calculated as follows:

$$
\begin{aligned}
C_{com}^{in} &= \mathcal{B} + (m_{max}-1)((\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + (\mathcal{D} + \mathcal{L}_0 + LIFS)) + (m_{max}-1)((m_{max}-1) \\
&\quad \times (\mathcal{D} + \mathcal{L}_0 + LIFS)) + 2n_{si} \times aMinCAPLength \\
&\quad + (\mathcal{B} + (m_{max}-1)((\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + (\mathcal{D} + \mathcal{L} + LIFS)) + (m_{max}-1)((m_{max}-1) \\
&\quad \times (\mathcal{D} + \mathcal{L} + LIFS)) + 2n_{si} \times aMinCAPLength)\bar{p}T_\epsilon \\
&= \mathcal{B} + (m_{max}-1)(\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + (m_{max}-1)(m_{max})(\mathcal{D} + \mathcal{L}_0 + LIFS) + 2n_{si} \\
&\quad \times aMinCAPLength + (\mathcal{B} + (m_{max}-1) \\
&\quad \times (\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) + (m_{max}-1)(m_{max}) \\
&\quad \times (\mathcal{D} + \mathcal{L} + LIFS) + 2n_{si} \times aMinCAPLength)\bar{p}T_\epsilon, \qquad (31)
\end{aligned}
$$

$$
\begin{aligned}
C_{com}^{out} &= \mathcal{B} + q(\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + \sum_{j=1}^{q}(\mathcal{D} + m_j \times \mathcal{L}_0 + LIFS) + \sum_{j=1}^{q}((N_v - m_j) \times \mathcal{L}_0 \\
&\quad + (q-1)\mathcal{D} + (q-1)LIFS) + 2n_{so} \times aMinCAPLength \\
&\quad + (\mathcal{B} + q(\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + \sum_{j=1}^{q}(\mathcal{D} + m_j \times \mathcal{L} + LIFS) + \sum_{j=1}^{q}((N_v - m_j) \times \mathcal{L} + (q-1)\mathcal{D} \\
&\quad + (q-1)LIFS) + 2n_{so} \times aMinCAPLength)T_\epsilon \\
&= \mathcal{B} + q(\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + q(q \times \mathcal{D} + N_v \times \mathcal{L}_0 + q \times LIFS) + 2n_{so} \times aMinCAPLength \\
&\quad + (\mathcal{B} + q(\mathcal{C} + aTurnaroundTime + \mathcal{A} + LIFS) \\
&\quad + q(q \times \mathcal{D} + N_v \times \mathcal{L} + q \times LIFS) + 2n_{so} \times aMinCAPLength)T_\epsilon, \qquad (32)
\end{aligned}
$$

$$
C_{com} = C_{com}^{in} + C_{com}^{out}, \qquad (33)
$$

where n_{si} and n_{so} are the numbers of superframes required in inner and outer iterates, respectively, and m_{max} is the size of the biggest cluster.

4. Calculating the optimal size of neighborhoods

In this section, for the typical office building of Figs. 2–5, we find the size of neighborhoods yielding the lowest computational latency, which is the summation of the computation overhead and communication overhead. Here, for the sake of simplicity, it is assumed that almost all neighborhoods have an equal size (i.e., $m_j = m$). For a given m, each neighborhood includes nearby rooms/hallways from the same floor or different floors.

Since obtaining an explicit mathematical relation between the size of neighborhoods and computation overhead for large-scale building automation and control systems is very difficult, or perhaps impossible, in this section for the typical office building of Figs. 2–5, the best possible size of the neighborhood (m^*) is determined using computer simulations. To achieve this goal, for each m, the distributed optimization method of Section 2.3 is applied to the typical office building of Figs. 2–5 with different initial conditions; and the average computation overhead is calculated for each m. Then, by comparing the average computational latencies, the m^* that results in the lowest average computational latency is determined. Note that the average computational latency obtained here is a measure of the computational latency (in transient regime) of the distributed model predictive control method of Section 2 applied to this office building.

Table 2
Different types of neighborhoods in the typical office building.

Number of neighborhoods (q)	Size of neighborhoods (m_j)	m_{max}	n_{si}	q	n_{so}
6	18, 18, 18, 18, 18, 8	18	3	6	1
7	14, 14, 14, 14, 14, 14, 14	14	2	7	1
9	11, 11, 11, 11, 11, 11, 11, 11, 10	11	2	9	2
14	7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7	7	1	14	2

Table 3
Initial conditions for different m.

Ambient temperature(T_{oa}) °C = Inside temperature($x_i[0]$) °C	Initial input (u_0^i) °C
10	30
12	30
14	30
16	30

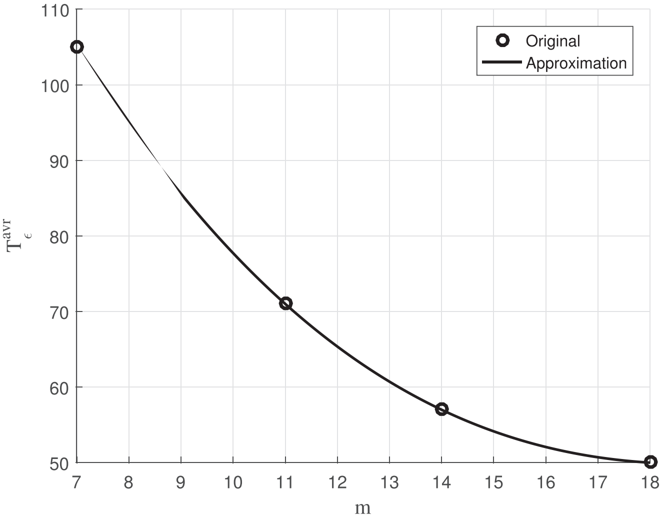

Fig. 17. Circle: average T_ϵ for m =7,11,14,18. Solid line: curve fit $T_\epsilon^{avr} = -0.0119m^3 + 0.9286m^2 - 22.2738m + 219.5$.

For the simulation, we set $N = 6$, $\lambda_i = \frac{1}{q}$, $w_i = \frac{1}{m}$, $\epsilon = 5$, $\bar{p} = 10$, $y_i^d = 22$ °C, $u_{i,min} = 15$, $u_{i,max} = 40$, $y_{i,min} = 10$, $y_{i,max} = 25$, $Q = 100R$, and $R = I$, $\mathcal{B} = 2.432$ ms, $\mathcal{D} = 1.120$ ms, $\mathcal{L}_0 = 0.256$ ms, $\mathcal{L} = 0.128$ ms, $\mathcal{A} = 0.352$ ms, $\mathcal{C} = 1.056$ ms, $aMinCAPLength = 7.04$ ms, $aTurnAroundTime = 0.192$ ms, and $LIFS = 0.64$ ms. Tables 2 and 3 illustrate values for m and initial conditions used for each m. All the simulations are run on a laptop with Windows 64-bit operating system, 1.73 GHz core i7 processor, and 8 GB RAM memory. In real applications, the embedded processor of each ZigBee module can be used for performing the required computation. As each node only needs to perform a very simple computation (e.g., (29)), the communication overhead is the dominating factor in computational latency.

Fig. 17 illustrates the average total number of iterations for ϵ-convergence (T_ϵ^{ave}). Fig. 18 illustrates the average computation overhead (C_{comp}^{ave}), Fig. 19 illustrates the average communication overhead (C_{comm}^{ave}), and Fig. 20 illustrates the average computational latency (C_{total}^{ave}).

As it is clear from the above figures, by increasing the size of neighborhoods m, the communication overhead increases (Fig. 19), and the computational overhead decreases (Fig. 17). Therefore, as it is seen from Fig. 20, the computational latency, which is

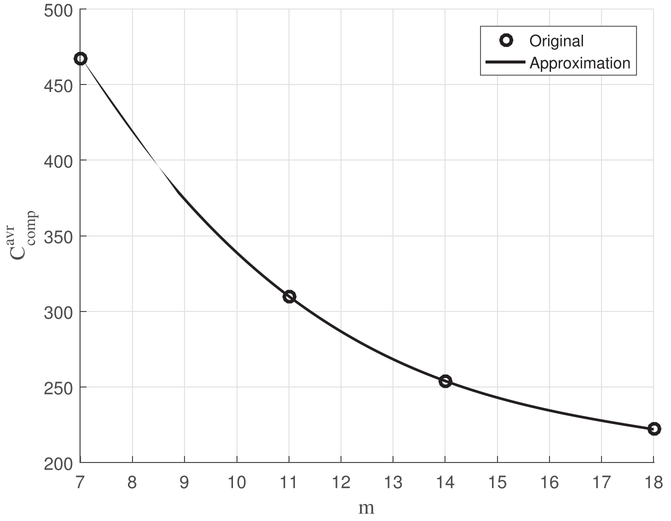

Fig. 18. Circle: average processors computation overhead in second for m =7,11,14,18. Solid line: curve fit $C^{avr}_{comp} = -0.1287m^3 + 7.0616m^2 - 134.5497m + 1107$.

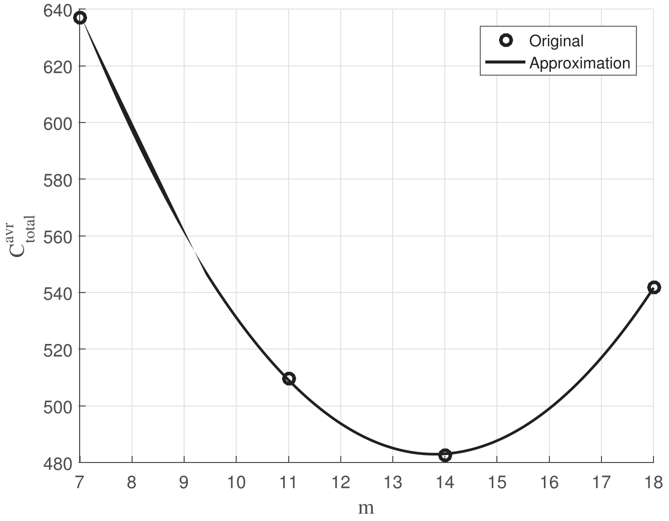

Fig. 20. Circle: average computational latency in seconds for m =7,11,14,18. Solid line: curve fit$C^{avr}_{total} = 3.33m^2 - 91.96m + 1117.6$.

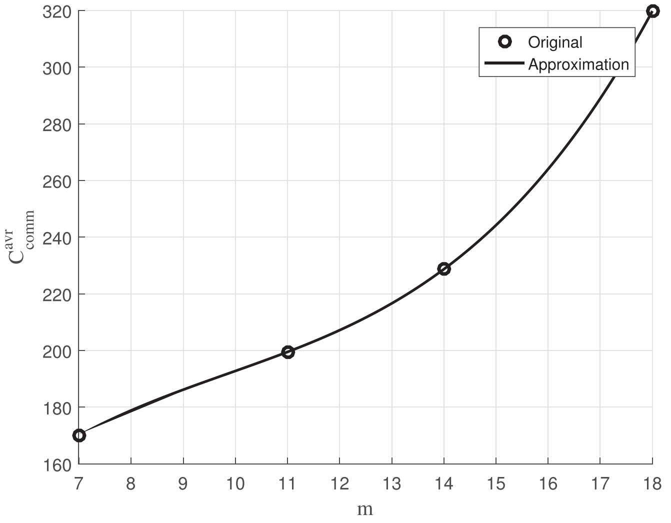

Fig. 19. Circle: average communication overhead in second for m =7,11,14,18. Solid line: curve fit $C^{avr}_{comm} = 0.1385m^3 - 4.0974m^2 + 46.8948m - 4.986$.

the summation of communication overhead and computation overhead, is approximated by a second-order polynomial, which has a minimum; and for the conditions simulated, this minimum is $m^* = 14$. m^* corresponds to the size of neighborhoods, which are neither very small nor very large.

5. Simulation results

In this section, the distributed model predictive control method of Section 2 is applied to the typical office building of Figs. 2–5, and its performance in disturbance rejection when $m^* = 14$ is shown.

As mentioned in the previous section, the size of neighborhoods for having the lowest computational latency is $m^* = 14$, where for this case the average computational latency is $C^{avr}_{total} = 482$s. Therefore, the time step must be considered much larger than this time latency to compensate for its effects. In this paper, the time step is chosen to be at least four times greater than the average computational latency. That is, for $m^* = 14$, the time step is chosen to be $T = 1800$s (half of an hour).

The result of tracking the reference temperature of 22°C for BACS of the typical office building of Figs. 2–5 with an initial temperature of 10°C, initial input (heater temperature at the beginning of the process) of 27°C and $y_{i,max} = 25$, $\dot{m}_s{}^{i} = 0.4530(\frac{\text{kg}}{\text{s}})$, $c_p = 1005(\frac{\text{J}}{\text{kg K}})$, and $m = 14$ with supply air and sol-air temperatures are shown in Fig. 21. In this simulation, we consider the initial heater temperature to be 27°C which results in an overshot that only takes a few minutes. This illustrates the satisfactory performance of the proposed method. In this simulation, it is assumed that no air exchange between rooms occurs via doors. Note that to obtain this result, the sol-air temperatures are calculated for $\alpha = 0.6$, $\xi = 0.9$, $h_0 = 20$ (roof), $h_0 = 16.67$ (walls), and mean hourly solar radiation values for Tehran in December as presented in Table 4. It is assumed that the controller cannot predict the heat flux through south, north, west, and east external building walls and the roof consists of solar radiation.

The result of simulations considering that all internal doors are open for the first 30 minutes of the simulation and then all of them are closed for the rest of the simulation is shown in Fig. 22 with supply air and sol-air temperatures. Note that the effects of open doors are not considered in obtaining control inputs, and hence, it is considered as a disturbance to the system.

For a better comparison between these two scenarios, the results for both scenarios are shown in Fig. 23. Also, for a better comparison, the percentage of the steady-state tracking errors for each subsystem/room in Figs. 21 and 22 are calculated using the following formula and are shown in Tables 5 and 6, respectively: The Tracking Error Percentage = The absolute value of the steady-state value of the temperature of each room minus the Desired Temperature, divided to the Desired Temperature ×100, where the Desired Temperature=22.

Remark 5.1. i) An attempt to repeat the above simulations using the centralized methods [16,18] failed after 8 h of simulation due to a computer crash caused by the enormous computational complexity of the centralized methods for large-scale systems. This indicates that for implementing these methods on large-scale office buildings, we need to purchase a very very powerful centralized computer, which is obviously very expensive and requires frequent

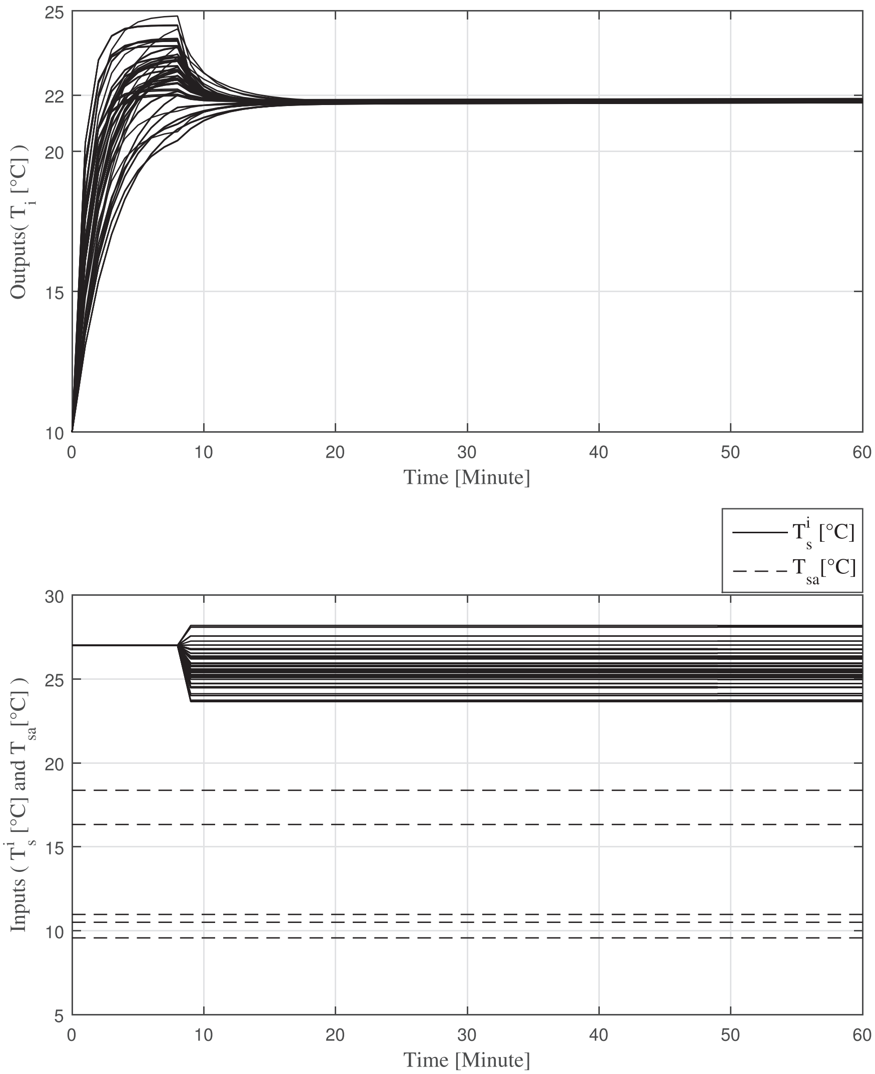

Fig. 21. The temperature and supply air temperature of each room/hallway of the typical office building, and sol-air temperature of each orientation considering internal doors are close when the distributed model predictive control method with $m^* = 14$ is applied for the temperature control under the assumption that the internal doors remain close during simulation time.

Table 4
Mean hourly solar radiation values for Tehran in December [21].

Hours of day	Horizontal orientation $\left(\frac{W}{m^2}\right)$	South orientation $\left(\frac{W}{m^2}\right)$	East orientation $\left(\frac{W}{m^2}\right)$	North orientation$\left(\frac{W}{m^2}\right)$	West orientation $\left(\frac{W}{m^2}\right)$
8	80.23	175.87	232.46	13.95	26.94

Table 5
The tracking error percentage for each room of Fig. 21.

Room1: 1.00%	Room2: 1.04%	Room3: 1.02%	Room4: 1.05%	Room5: 1.08%	Room6: 0.99%	Room7: 0.93%
Room8: 0.87%	Room9: 0.87%	Room10: 0.90%	Room11: 0.90%	Room12: 0.98%	Room13: 0.90%	Room14: 0.85%
Room15: 1.08%	Room16: 0.75%	Room17: 1.04%	Room18: 0.77%	Room19: 1.12%	Room20: 0.94%	Room21: 1.14%
Room22: 1.09%	Room23: 0.94%	Room24: 1.03%	Room25: 0.95%	Room26: 1.03%	Room27: 1.04%	Room28: 0.81%
Room29: 0.91%	Room30: 0.96%	Room31: 0.80%	Room32: 0.92%	Room33: 0.94%	Room34: 0.93%	Room35: 0.94%
Room36: 0.92%	Room37: 1.21%	Room38: 1.23%	Room39: 0.87%	Room40: 0.90%	Room41: 0.90%	Room42: 0.99%
Room43: 0.90%	Room44: 0.85%	Room45: 1.08%	Room46: 0.75%	Room47: 1.04%	Room48: 0.77%	Room49: 1.12%
Room50: 0.94%	Room51: 1.14%	Room52: 1.09%	Room53: 0.94%	Room54: 1.03%	Room55: 0.95%	Room56: 1.03%
Room57: 1.04%	Room58: 0.81%	Room59: 0.92%	Room60: 0.97%	Room61: 0.80%	Room62: 0.92%	Room63: 0.94%
Room64: 0.93%	Room65: 0.94%	Room66: 0.92%	Room67: 1.21%	Room68: 1.23%	Room69: 0.80%	Room70: 0.83%
Room71: 0.82%	Room72: 0.92%	Room73: 0.83%	Room74: 0.77%	Room75: 1.03%	Room76: 0.60%	Room77: 1.01%
Room78: 0.55%	Room79: 1.10%	Room80: 0.86%	Room81: 1.12%	Room82: 1.03%	Room83: 0.88%	Room84: 0.96%
Room85: 0.88%	Room86: 0.96%	Room87: 1.01%	Room88: 0.73%	Room89: 0.85%	Room90: 0.91%	Room91: 0.58%
Room92: 0.85%	Room93: 0.88%	Room94: 0.86%	Room95: 0.86%	Room96: 0.85%	Room97: 1.15%	Room98: 1.18%

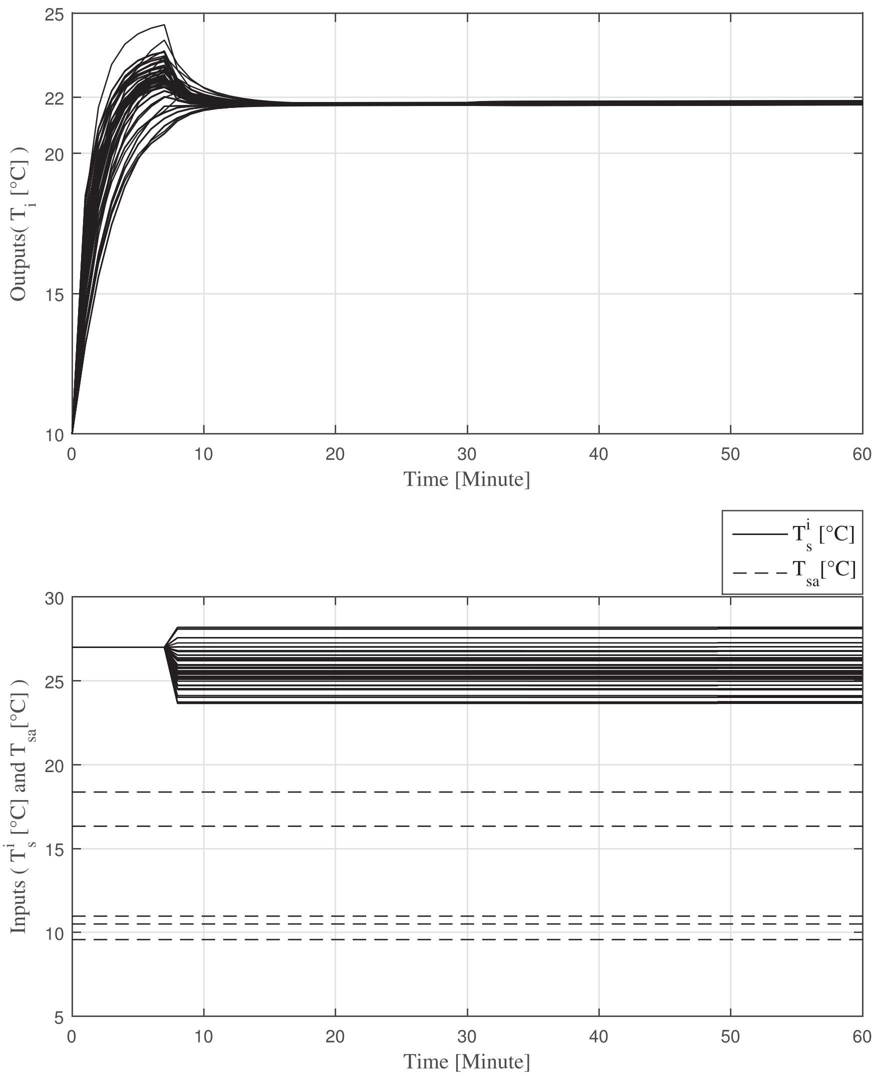

Fig. 22. The temperature and supply air temperature of each room/hallway of the typical office building, and sol-air temperature of each orientation considering the internal doors are open in the first 30 mins and then closed when the distributed model predictive control method with $m^* = 14$ is applied for the temperature control.

Table 6
The tracking error percentage for each room of Fig. 22.

Room1: 1.00%	Room2: 1.04%	Room3: 1.02%	Room4: 1.05%	Room5: 1.07%	Room6: 0.99%	Room7: 0.93%
Room8: 0.87%	Room9: 0.87%	Room10: 0.90%	Room11: 0.90%	Room12: 0.98%	Room13: 0.90%	Room14: 0.85%
Room15: 1.08%	Room16: 0.75%	Room17: 1.04%	Room18: 0.77%	Room19: 1.12%	Room20: 0.94%	Room21: 1.14%
Room22: 1.09%	Room23: 0.94%	Room24: 1.03%	Room25: 0.95%	Room26: 1.03%	Room27: 1.04%	Room28: 0.81%
Room29: 0.91%	Room30: 0.96%	Room31: 0.80%	Room32: 0.92%	Room33: 0.94%	Room34: 0.93%	Room35: 0.94%
Room36: 0.93%	Room37: 1.21%	Room38: 1.23%	Room39: 0.87%	Room40: 0.90%	Room41: 0.90%	Room42: 0.99%
Room43: 0.90%	Room44: 0.85%	Room45: 1.08%	Room46: 0.75%	Room47: 1.04%	Room48: 0.77%	Room49: 1.12%
Room50: 0.94%	Room51: 1.13%	Room52: 1.09%	Room53: 0.94%	Room54: 1.03%	Room55: 0.95%	Room56: 1.03%
Room57: 1.04%	Room58: 0.81%	Room59: 0.92%	Room60: 0.97%	Room61: 0.80%	Room62: 0.92%	Room63: 0.94%
Room64: 0.93%	Room65: 0.94%	Room66: 0.93%	Room67: 1.21%	Room68: 1.23%	Room69: 0.80%	Room70: 0.83%
Room71: 0.82%	Room72: 0.92%	Room73: 0.83%	Room74: 0.77%	Room75: 1.03%	Room76: 0.60%	Room77: 1.01%
Room78: 0.55%	Room79: 1.10%	Room80: 0.86%	Room81: 1.12%	Room82: 1.03%	Room83: 0.88%	Room84: 0.96%
Room85: 0.88%	Room86: 0.96%	Room87: 1.01%	Room88: 0.73%	Room89: 0.85%	Room90: 0.91%	Room91: 0.58%
Room92: 0.85%	Room93: 0.88%	Room94: 0.86%	Room95: 0.87%	Room96: 0.85%	Room97: 1.15%	Room98: 1.18%

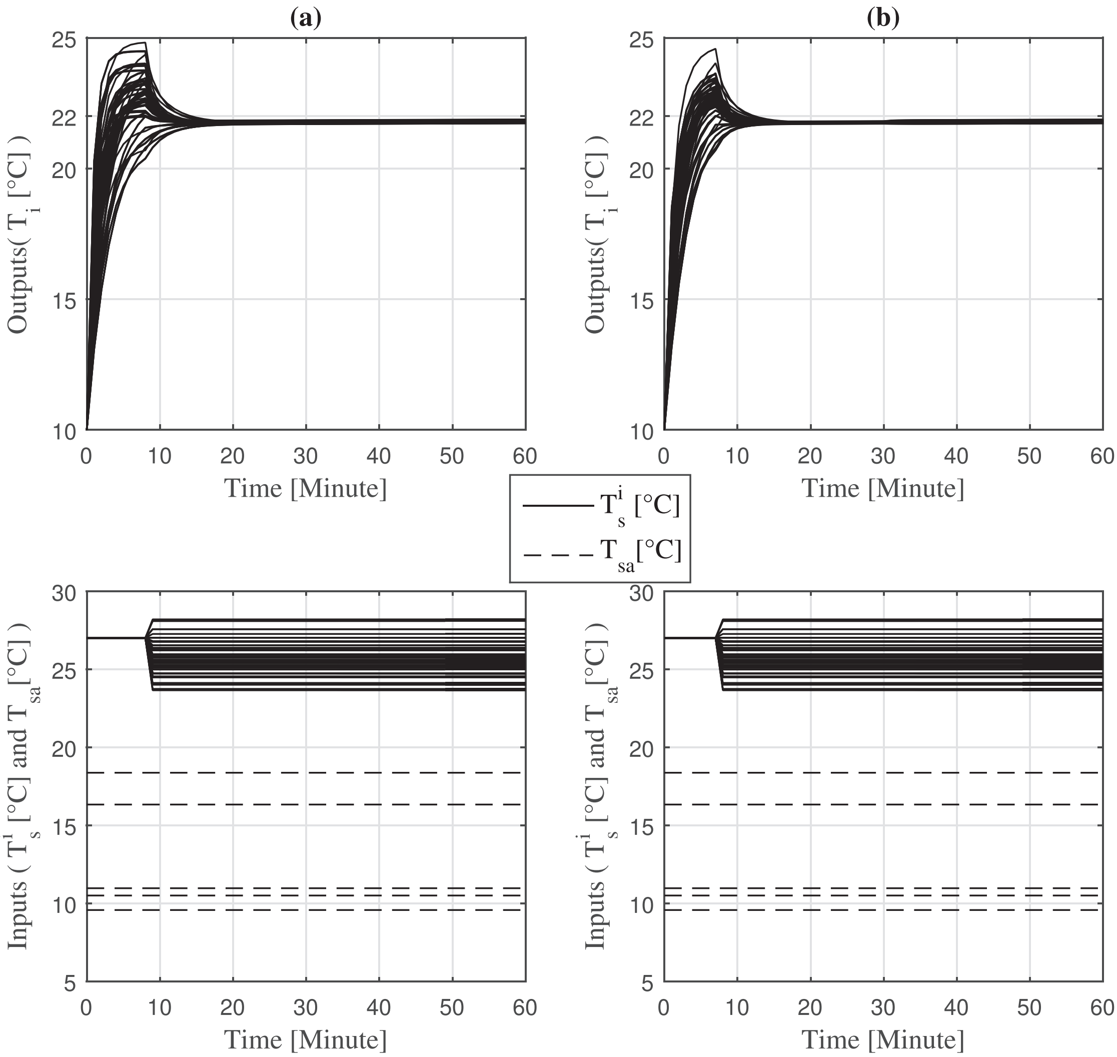

Fig. 23. The temperature and supply air temperature of each room/hallway of the typical office building, and sol-air temperature of each orientation considering two-level communication architecture with the optimal size of neighborhoods and (a) doors are close during simulation time (b) doors are open in the first 30 min of simulation and then are closed.

maintenance. However, using the distributed method proposed in this paper, we can successfully achieve the desired performance using the available distributed embedded devices without requiring to purchase a very expensive centralized computer.
ii) Note that all simulations of this section include disturbances. Disturbances considered in the simulations are solar radiation, outdoor temperature, occupancy and lights disturbance and open door effects. For the occupancy and lights disturbance, we randomly considered between 1 to 3 persons (100W each) and between 1 to 2 lamps (100W each) in rooms. In the first set of simulations, i.e., Fig. 21, we considered the first three disturbances and in the second set of simulation, i.e., Fig. 22 we also considered the open door effects; and we observed that the proposed method completely compensated the effects of these disturbances.

Now, to illustrate the significant impact of the proposed two-level communication architecture with the optimal size of neighborhoods on reducing the computational latency and enhancing the tracking performance, we repeat the simulations of Fig. 23 without implementing this architecture. That is, we consider only one neighborhood with 98 subsystems subject to all to all communication all the time; and we repeat the simulations of Fig. 23. For this case, $T_{\epsilon}^{avr} = 563$, $C_{comp}^{avr} = 326$s, $C_{comm}^{avr} = 9771$s and therefore $C_{total}^{avr} = 10097$s. The simulation results for this case are given in Fig. 24. As it is clear from this figure and Fig. 23, implementing the proposed two-level communication architecture with the optimal size of neighborhoods has a significant impact on reducing the computational latency and enhancing the tracking performance for large-scale smart buildings.

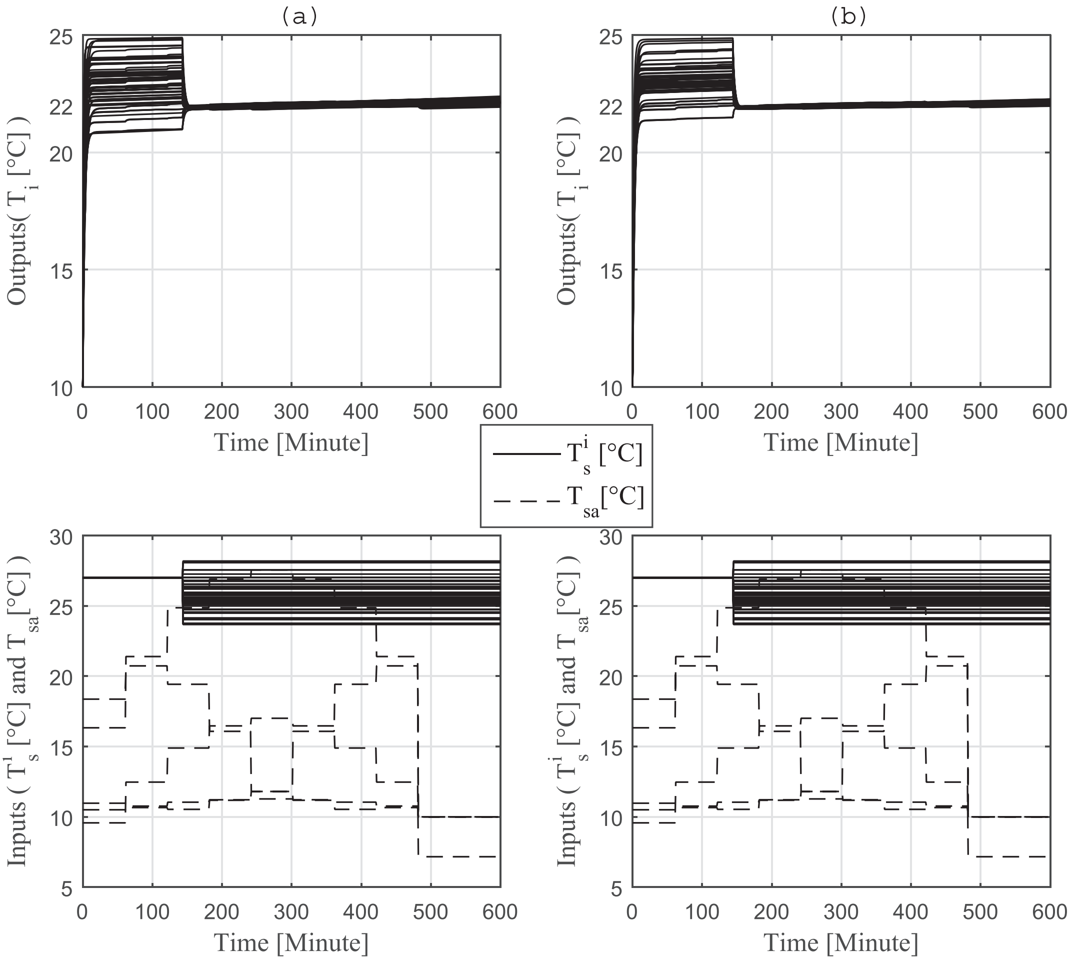

Fig. 24. The temperature and supply air temperature of each room/hallway of the typical office building, and sol-air temperature of each orientation considering one neighborhood with 98 subsystems and (a) doors are close during simulation time (b) doors are open in the first 30 min of simulation and then are closed.

6. Conclusion

This paper introduced a cyber-physical system for building automation and control, which was developed based on the distributed optimization method of [19] with the optimal size of neighborhoods. To address this problem, the associated communication overhead was calculated. Then, a novel algorithm for finding the size of neighborhoods for a typical office building, yielding the lowest computational latency was presented. Finally, the satisfactory performance of the proposed wireless cyber-physical system with the optimal size of neighborhoods for the temperature control of this office building was illustrated using computer simulations in the presence of disturbance. As it has been shown in this paper and also in [16,18], the distributed method of [19] used in this paper is practical for large-scale BACSs with respect to the centralized methods as the centralized methods are computationally expensive.

In this paper, the decision variable is the temperature of the source, i.e., T_s. However, another way for controlling the temperature of office buildings is via source flow rate. Therefore, for the future, it is interesting to use the source flow rate as the decision variable for controlling the temperature of office buildings. This problem is left for future investigation.

Declaration of Competing Interest

There is no conflict of interest associated with this paper.

References

[1] N. Abdeddaim, F. Theoleyre, F. Rousseau, A. Duda, Multi-channel cluster tree for 802.15. 4 wireless sensor networks, in: Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on, IEEE, 2012, pp. 590–595.

[2] Z. Alliance, ZigBee Specification, ZigBee Standards Organization, 2005.

[3] N. Artmann, R.L. Jensen, H. Manz, P. Heiselberg, Experimental investigation of heat transfer during night-time ventilation, Energy and buildings 42 (3) (2010) 366–374.

[4] F.A. Barata, J.M. Igreja, R. Neves-Silva, Distributed mpc for thermal comfort and load allocation with energy auction, International Journal of Renewable Energy Research (IJRER) 4 (2) (2014) 371–383.

[5] W. Beckman, J. Mitchell, Transfer functions for efficient calculation of multidimensional transient heat transfer, Transfer (1930).

[6] T.L. Bergman, Introduction to heat transfer, John Wiley & Sons, 2011.

[7] J.E. Braun, K.W. Montgomery, N. Chaturvedi, Evaluating the performance of building thermal mass control strategies, HVAC&R Research 7 (4) (2001) 403–428.

[8] D. Builder, Design builder software ltd, Recuperado de (2009). https://www.designbuilder.co.uk

[9] Building energy data book of doe, (http://buildingsdatabook.eren.doe.gov/default.aspx/). February 2013.

[10] L.S. Committee, et al., Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans), IEEE Computer Society (2003).

[11] D. Dahlhaus, T. Hunziker, S. Vassilaras, H. Al-Raweshidy, M. De Sanctis, Pan-optimized air interfaces, in: My personal Adaptive Global NET (MAGNET), Springer, 2010, pp. 135–243.

[12] M.M. Delbari, M.T.H. Beheshti, A. Ramezani, S. Ozgoli, Time delay sensitivity analysis in a wireless network control system using lmi approach, Asian Control of Control (2013).

[13] K. Deng, P. Barooah, P.G. Mehta, S.P. Meyn, Building thermal model reduction via aggregation of states, in: American Control Conference (ACC), 2010, 2010, pp. 5118–5123.

[14] M. Długosz, Aggregation of state variables in an rc model, Building Services Engineering Research and Technology 39 (1) (2018) 66–80r.

[15] S. Farahani, ZigBee wireless networks and transceivers, newnes, 2011.

[16] A. Farhadi, M. Cantoni, P.M. Dower, Computation time analysis of a distributed optimization algorithm applied to automated irrigation networks, Proceedings of the 2013 IEEE Conference on Decision and Control (2013) 2193–2199.

[17] A. Farhadi, J. Domun, C. Canudas de Wit, A supervisory control policy over an acoustic communication network, International Journal of Control 85 (5) (2015) 946–958.

[18] A. Farhadi, P.M. Dower, M. Cantoni, Computation time analysis of a centralized and distributed optimization algorithms applied to automated irrigation networks, Proceedings of the 2013 Australian Control Conference (2013) 263–269.

[19] A. Farhadi, A. Khodabandehlou, Distributed model predictive control with hierarchical architecture for communication: Application in automated irrigation channels, International Journal of Control 89 (8) (2016) 1725–1741.

[20] S. Ferrari, V. Zanotto, Building energy performance assessment in Southern Europe, Springer, 2015.

[21] Y. Gorji-Mahlabani, Climatic effects on school buildings: methods of optimising the energy performance of school buildings in the different climatic regions of Iran, University of Sheffield, 2002 Ph.D. thesis.

[22] M. Gouda, S. Danaher, C. Underwood, Low-order model for the simulation of a building and its heating system, Building Services Engineering Research and Technology 21 (3) (2000) 199–208.

[23] M. Gouda, S. Danaher, C. Underwood, Building thermal model reduction using nonlinear constrained optimization, Building and Environment 37 (12) (2002) 1255–1265.

[24] S. Goyal, P. Barooah, A method for model-reduction of non-linear thermal dynamics of multi-zone buildings, Energy and Buildings 47 (2012) 332–340.

[25] J. Hahn, T.F. Edgar, An improved method for nonlinear model reduction using balancing of empirical gramians, Computers & chemical engineering 26 (10) (2002) 1379–1397.

[26] A. Handbook, et al., Fundamentals, American Society of Heating, Refrigerating and Air Conditioning Engineers, Atlanta 111 (2001).

[27] X. Hou, Y. Xiao, J. Cai, J. Hu, J.E. Braun, A distributed model predictive control approach for optimal coordination of multiple thermal zones in a large open space, in: 4th International High Performance Buildings Conference, 2016.

[28] X. Jin, Q. Zhang, P. Zeng, F. Kong, Y. Xiao, Collision–free multichannel superframe scheduling for ieee 802.15. 4 cluster–tree networks, International Journal of Sensor Networks 15 (4) (2014) 246–258.

[29] J. Ma, S.J. Qin, T. Salsbury, Model predictive control of building energy systems with balanced model reduction, in: 2012 American Control Conference (ACC), 2012, pp. 3681–3686.

[30] Y. Ma, G. Anderson, F. Borrelli, A distributed predictive control approach to building temperature regulation, in: American Control Conference (ACC), 2011, 2011, pp. 2089–2094.

[31] Y. Ma, A. Kelman, A. Daly, F. Borrelli, Predictive control for energy efficient buildings with thermal storage, IEEE Control system magazine 32 (1) (2012) 44–64.

[32] J. Maestre, D.M. De La Pena, E. Camacho, T. Alamo, Distributed model predictive control based on agent negotiation, Journal of Process Control 21 (5) (2011) 685–697.

[33] P. Maxwell, F. Durrani, M. Eftekhari, Investigating heat loss through vestibule doors for a non-residential building, Scholink, 2016.

[34] B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, IEEE transactions on automatic control 26 (1) (1981) 17–32.

[35] P.-D. Moroşan, R. Bourdais, D. Dumur, J. Buisson, Building temperature regulation using a distributed model predictive control, Energy and Buildings 42 (9) (2010) 1445–1452.

[36] R.R. Negenborn, Multi-agent model predictive control with applications to power networks, PhD thesis, Delft University of Technology, 2007.

[37] P. Park, C. Fischione, K.H. Johansson, Performance analysis of gts allocation in beacon enabled ieee 802.15. 4, in: 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009, pp. 1–9.

[38] P. Rawat, K.D. Singh, H. Chaouchi, J.M. Bonnin, Wireless sensor networks: a survey on recent developments and potential synergies, The Journal of supercomputing 68 (1) (2014) 1–48.

[39] S. Sawaf, J. Barlow, E. Essah, J. Broadbent, B. Gregson, Calculation of heat flow across doorways in a food supermarket (????).

[40] H. Scherer, M. Pasamontes, J. Guzmán, J. Álvarez, E. Camponogara, J. Normey-Rico, Efficient building energy management using distributed model predictive control, Journal of Process Control 24 (6) (2014) 740–749.

[41] J. Širokỳ, F. Oldewurtel, J. Cigler, S. Prívara, Experimental analysis of model predictive control for an energy efficient building heating system, Applied Energy 88 (9) (2011) 3079–3087.

[42] I. Standard, 7730.(2005): Ergonomics of the thermal environment-Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria, iso.org, 2005.

[43] D.G. Stephenson, Extreme temperatures at the outer surfaces of buildings, National Government Publication (Canada), 1963.

[44] B.T. Stewart, J.B. Rawlings, S.J. Wright, Hierarchical cooperative distributed model predictive control, in: American Control Conference (ACC), 2010, IEEE, 2010, pp. 3963–3968.

[45] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, G. Pannocchia, Cooperative distributed model predictive control, Systems & Control Letters 59 (8) (2010) 460–469.

[46] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, G. Pannocchia, Cooperative distributed model predictive control, Systems & Control Letters 59 (8) (2010) 460–469.

[47] I. TG4a, Part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans), IEEE Standard for Information Technology (2007).

[48] E. Toscano, L.L. Bello, Multichannel superframe scheduling for ieee 802.15. 4 industrial wireless sensor networks, IEEE Transactions on Industrial Informatics 8 (2) (2012) 337–350.

[49] X. Xu, S. Wang, A simplified dynamic model for existing buildings using ctf and thermal network models, International Journal of Thermal Sciences 47 (9) (2008) 1249–1262.

[50] S. Yang, B. Yu, Output feedback stabilization of networked control systems with random delays modeled by markov chains, IEEE Transactions on Automatic Control (2009).