Control Engineering Practice 110 (2021) 104782

Contents lists available at ScienceDirect

Control Engineering Practice

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

Accelerated distributed model predictive control for HVAC systems

Liangliang Chen, Ying Zhang *

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Keywords:
Distributed model predictive control
Fast dual gradient algorithm
HVAC systems
Thermal couplings
Distributed average consensus

ABSTRACT

This paper investigates the accelerated distributed model predictive control (MPC) strategy for the heating, ventilation and air conditioning (HVAC) systems with local and global power input constraints. The problems are firstly formulated in the distributed MPC framework and then the constrained optimization is converted into a quadratic programming problem. In the problem formulation, the thermal couplings between immediate neighboring zones are considered while designing the distributed controller, and the unknown thermal disturbances are incorporated by the robust optimization scheme. Then, using the accelerated dual gradient-projection method, a distributed fast MPC protocol is designed for HVAC systems considering both the electricity cost and occupant comforts. A distributed stopping criterion based on the distributed average consensus algorithm is utilized. Finally, numerical simulations are used to demonstrate the effectiveness of the proposed distributed MPC algorithm, and its computational advantages comparing with an existing distributed method and a centralized algorithm.

1. Introduction

The energy consumption in building systems accounts for almost 40% of the total energy consumption in the United States (Koebrich, Bowen, & Sharpe, 2020), while the heating, ventilation and air conditioning (HVAC) systems make up 30% of the energy consumption in commercial building systems (Goetzler et al., 2017). The consumed energy is responsible for the large amount of green house gas emissions. On the other hand, buildings are where people spend their much time (Klein et al., 2012), so that the occupant comfort is one another critical consideration when designing HVAC control systems. Thus, it is essential to develop the HVAC systems which can both save energy and guarantee occupants' comfort (Hussain, Gabbar, Bondarenko, Musharavati, & Pokharel, 2014).

In order to ensure the comfort of occupants as far as possible, we should consider the thermal preferences of different individuals, instead of the average thermal preference (Kim, Zhou, Schiavon, Raftery, & Brager, 2018). One strategy for this problem is to use different HVAC subsystems in distinct zones. Distributed control is a method that each subsystem makes its own local decisions by communicating with other subsystems in a certain way (Cao, Chen, Xiao, & Sun, 2009; Mei & Xia, 2019). Besides, every single HVAC system has a power limit, and the total HVAC energy consumption of a building should be constrained due to the distribution infrastructure limit within the building system (Zhang, Deng, Yuan, & Qin, 2017a). Thus, the control aim of distributed HVAC systems is to minimize the energy consumption while maintaining occupants' thermal comfort, under both local and global power constraints (Yang, Hu, & Spanos, 2020). Model predictive control (MPC) can serve as a control method that makes explicit use of the model information and obtains the optimal control signal by minimizing a specified objective function over a given time horizon under certain constraints (Camacho & Alba, 2013). Thus, MPC is widely used in the energy efficiency problems of building and HVAC systems (O'Dwyer, De Tommasi, Kouramas, Cychowski, & Lightbody, 2017; Serale, Fiorentini, Capozzoli, Bernardini, & Bemporad, 2018). West, Ward, and Wall (2014) solved the multi-objective optimization problem for commercial buildings under the framework of MPC. The considered objectives include running cost, CO_2 emissions, and occupant thermal comfort. A real-time online thermal comfort was measured based on the predicted mean vote (PMV) model (Humphreys & Nicol, 2002). However, the authors used the average PMV from individuals wherein the considered thermal preference was not personalized. Valenzuela, Ebadat, Everitt, and Parisio (2019) presented a robust multivariable HVAC supervisory MPC framework with data-driven technique to identify the HVAC system dynamics. It was demonstrated that the energy efficiency was improved and the proposed controller could deal with multiple set points, such as both indoor temperatures and air flow rate. Lee, Ooka, Ikeda, Choi, and Kwak (2020) investigated the MPC strategy for commercial buildings when occupancy schedules and electricity prices were time-varying. The simulations showed that the MPC could reduce the total operating cost compared with a conventional rule-based controller. It should be noted that only one building zone was considered in Lee et al. (2020), Valenzuela et al. (2019) and West et al. (2014). When there are

* Corresponding author.
E-mail address: yzhang@gatech.edu (Y. Zhang).

https://doi.org/10.1016/j.conengprac.2021.104782
Received 11 January 2021; Received in revised form 16 February 2021; Accepted 18 February 2021
Available online 1 March 2021
0967-0661/

multiple zones or multiple buildings, the controllers in different zones cannot collaborate with each other to avoid violating the global power constraint. As mentioned above, when there are multiple controllers, a distributed control method allows subsystems to make their own local decisions when there exist coupled constraints. Distributed MPC provides an excellent strategy to achieve both customized thermal preferences and energy consumption optimization under global and local power constraints (Afram & Janabi-Sharifi, 2014; Camponogara, Jia, Krogh, & Talukdar, 2002). Moroşan, Bourdais, Dumur, and Buisson (2010) firstly developed a building temperature regulation approach in the case of a single zone and then extended the algorithm to the multi-zone building case. It was demonstrated that the distributed MPC approach can effectively reduce the computational requirements compared with the centralized controller. However, the building model in Moroşan et al. (2010) did not consider the influence of the outdoor temperature, which had a non-negligible impact on the indoor temperature. Radhakrishnan, Srinivasan, Su, and Poolla (2017) proposed a learning-based hierarchical distributed MPC protocol for HVAC systems with operational constraints. A learning algorithm was utilized to capture the occupancy pattern and user interactions. Yu, Xie, Jiang, Zou, and Wang (2017) designed a distributed real-time HVAC controller based on the Lyapunov optimization technique, with considerations of time-varying electricity price, outdoor temperature, and occupant comfort. The authors showed in simulations that the presented algorithm can achieve energy cost reduction with small sacrifice in thermal comfort. Xie, Yu, Jiang, and Zou (2018) investigated the distributed energy optimization problem for HVAC systems when there exist multiple buildings and each building has multiple zones. A MPC scheme based on alternating direction method of multipliers (ADMM) was used to obtain the optimal power input for each subsystem. However, a multi-building coordinator and a building message controller in each building were required. The multi-building coordinator and building message controllers function like centralized units to determine if the iterations of the ADMM algorithm satisfies the stopping criterion, which made the controllers not fully distributed. In addition, the thermal coupling between neighboring zones is not considered in Radhakrishnan et al. (2017), Xie et al. (2018) and Yu et al. (2017), which is an important factoring affecting zone temperatures. Although Moroşan et al. (2010) and Yang et al. (2020) considered the thermal couplings between neighboring zones, Yang et al. (2020) did not use the MPC framework, thus the optimization was over the whole time horizon, instead of a small prediction horizon, which will lead to large computational loads and inaccurate prediction information, such as outdoor temperatures, when the considered time horizon is large. Moroşan et al. (2010) used the prediction of neighboring zones' indoor temperatures when designing the distributed model predictive controller. However, only three zones are considered and any two of them are neighbors, and the authors did not show how to make predictions using only local information when the number of zones is large. In fact, when the MPC is used for the HVAC control problem in the presence of thermal couplings between neighboring zones, the larger the prediction horizon of controller is, the information of more zones is required. If a distributed control system is regarded as the one where each agent only requires the information of its neighbors, then the MPC algorithm is no longer distributed when the prediction horizon is larger than 1, since the information of neighbors of neighbors is required for a certain agent.

Convergence rate is a key consideration for the controllers of HVAC systems, since the temperature regulation should be in real time in order to minimize the thermal discomfort of occupants in practice. The distributed ADMM algorithm in Xie et al. (2018) can only achieve the convergence rate of $\mathcal{O}\left(\frac{1}{k}\right)$ with k being the iteration number (Wang & Ong, 2017). When using the stopping criterion based on the distributed average consensus and the total number of zones is large, the computational efficiency is relatively low. Nesterov's method is a strategy that can solve convex optimization problems with convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$ (Bertsekas, 2009; Nesterov, 2005). Since many MPC problem can be transferred to convex optimization problem (East & Cannon, 2019; Müller & Allgöwer, 2017; Wang & Boyd, 2009), there are some researches focusing on the accelerated gradient projection algorithms for MPC recently. Patrinos and Bemporad (2014) developed an accelerated dual gradient projection protocol based on Nesterov's method, and gave the conditions of primal suboptimality and feasibility. A convergence rate of $\mathcal{O}\left(\frac{1}{k^2}\right)$ was proved for both dual optimality and primal optimality. After rewriting the linear MPC problems with state-input constraints in the augmented Lagrangian framework, Nedelcu, Necoara, and Tran-Dinh (2014) investigated the inexact dual fast gradient augmented Lagrangian methods and demonstrated its superiority in computation complexity. Li, Wu, Wu, Long, and Wang (2016) considered the convex optimization problems when there exist separable objective functions with linear coupled constraints. By employing Lagrangian dual decomposition and a fast proximal-gradient method, an inexact dual accelerated gradient-projection strategy was proposed. In the distributed case, Wang and Ong (2018) developed an accelerated distributed MPC scheme for linear systems with global constraints and proved the convergence rate of $\mathcal{O}\left(\frac{1}{k^2}\right)$ based on the Nesterov's method.

In this study, we proposed an accelerated distributed MPC algorithm for HVAC systems with global constraints based on the Nesterov's method. The thermal couplings between neighboring zones and the unknown bounded thermal disturbances are considered. The main contributions of this paper are stated as follows. It should be noted that the contributions (ii) and (iii) fill the gap between the distributed MPC for HAVC systems and the Nesterov's gradient-based accelerated algorithm, which is essential for introducing the accelerated MPC algorithm into the case of distributed HVAC systems.

(i) The dynamics of indoor temperatures is incorporated into the distributed MPC framework, including the thermal couplings between immediate neighboring zones. Considering the thermal couplings between immediate neighbors only it is possible to have a prediction horizon larger than 1 in the distributed MPC algorithm. In addition, the unknown bounded thermal disturbances are handled by robust optimization strategy.
(ii) To demonstrate the applicability of the Nesterov's gradient-projection method that requires converting the primal optimization problem to the dual form, we strictly prove the strong duality of the primal and dual optimization problems for the considered HVAC systems.
(iii) The Nesterov's gradient-projection algorithm requires that the objective function has a Lipschitz continuous gradient. In this study, we convert the objective function of MPC for the HVAC systems into a positive definite quadratic form, which is μ_i-strongly convex for some μ_i, and then obtain a Lipschitz constant L_g for the gradient of the objective function of the Nesterov's gradient-projection algorithm.

In addition, comparing with the distributed MPC for HVAC systems in Xie et al. (2018), the advantages of this work are listed as follows.

(i) The thermal couplings between zones and unknown bounded disturbances are considered in the presented MPC algorithm, while they are neglected in Xie et al. (2018).
(ii) Xie et al. (2018) designed a distributed model predictive controller for HVAC systems with global constraint. However, a multi-building coordinator and building message controllers were required for the stopping criterion, which made the controller not fully distributed. In this work, the proposed algorithm is fully distributed in that the multi-building coordinator and building message controllers are not needed and each HVAC subsystem can compute its own power input based on the information of the corresponding zone and the neighboring HVAC systems.
(iii) The convergence rate of the algorithm is $\mathcal{O}\left(\frac{1}{k}\right)$ in Xie et al. (2018). When using the fully distributed stopping criterion, the

computational time will be large when the number of zones is large. In this work, the convergence rate of the algorithm is $\mathcal{O}\left(\frac{1}{k^2}\right)$. The numerical simulation study demonstrates that the computational speed of the algorithm in this paper is faster than that in Xie et al. (2018).

The remainder of this paper is organized as follows. Section 2 introduces the necessary preliminary information and formulates the problem. Section 3 transforms the optimization problem obtained in Section 2 into one with a quadratic objective function, and then shows that how the thermal disturbances and parameter uncertainties can be incorporated. Section 4 presents the designing procedure for the accelerated distributed MPC algorithm for HVAC systems. Numerical simulations are given in Section 5 to demonstrate the effectiveness and advantages of the proposed algorithm. Section 6 draws the conclusions of this work.

Some notations used in this paper are introduced as follows. $\boldsymbol{I}_n$ represents the n-dimensional identity matrix. $\mathbf{1}_n$ and $\mathbf{0}_n$ represent the n-dimensional column vectors with all elements being 1 and 0, respectively. $\mathbb{R}$ and $\mathbb{R}^n$ denote the set of real numbers and the set of n-dimensional real vectors, respectively. If we denote $\mathbb{Z}_0^+$ the set of nonnegative integers, then $\mathbb{Z}_\ell^h \triangleq \{\ell, \ell+1, \ldots, h\}$ for any $\ell, h \in \mathbb{Z}_0^+$ and $\ell < h$. $(\cdot)^{\mathrm{T}}$ represents the transpose of a vector or a matrix. $\max(\cdot)$ and $\min(\cdot)$ denote the maximum and minimum functions of the corresponding values, respectively. For two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\boldsymbol{a} < \boldsymbol{b}$ means that the elements of $\boldsymbol{a}$ is smaller than the corresponding elements of $\boldsymbol{b}$, respectively. The other comparison operators for vectors are also element-wise. The notation $[\boldsymbol{x}]_+ \triangleq \max(\mathbf{0}_{h-1}, \boldsymbol{x})$ with the element-wise maximum denotes the projection of $\boldsymbol{x} \in \mathbb{R}^n$ on the set $\{\boldsymbol{y} \mid \boldsymbol{y} \geq \mathbf{0}_{h-1}\}$. $\|\cdot\|_2$ denotes the Euclidean norm of a vector. For a matrix $\boldsymbol{A}$, $\boldsymbol{A} > 0$ means that $\boldsymbol{A}$ is positive definite. $\mathcal{U}(a, b)$ represents the continuous uniform distribution over the interval $[a, b]$.

2. Preliminaries and problem formulation

2.1. Graph theory

We consider M buildings and there are N connected zones in each building. An undirected graph $G = (\mathcal{V}, \mathcal{E})$ is used to describe the neighborhood relationships among the MN zones, where $\mathcal{V}$ denotes the set of vertexes, i.e., the zones, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes the set of edges, respectively. If subsystem i can exchange information with subsystem j, then $(i, j) \in \mathcal{E}$, otherwise $(i, j) \notin \mathcal{E}$. The adjacency matrix $\boldsymbol{A}$ of the graph G is an $MN \times MN$ matrix with the element a_{ij} satisfying $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. In addition, we assume $a_{ii} = 0$. All the subsystems which can communicate with subsystem i form a set, called the neighbor set, which is $\mathcal{N}_i \triangleq \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}, i \neq j\}$. The matrix $\boldsymbol{D}$ is a diagonal matrix with elements $d_i \triangleq \sum_{j=1}^{MN} a_{ij}$, then the Laplacian matrix of the network G can be defined as $\boldsymbol{L} \triangleq \boldsymbol{D} - \boldsymbol{A}$. An undirected path is a sequence of edges in a undirected graph G in the form of (i_1, i_2), (i_2, i_3), …. If there exists an undirected path between any two edges in G, then the undirected graph G is connected (Ren & Beard, 2008). We assume that the graph G is connected in this work. The diameter D of a graph is defined as the longest of the shortest paths between any two nodes.

2.2. Distributed HVAC systems

In this part, the model of distributed HVAC systems considered in this paper will be introduced.

We consider the ith zone of the systems, with power consumption $P_i(k)$ in kW at time slot k. The thermal dynamics of zone i with an HVAC system working in the cooling mode can be described as (Constantopoulos, Schweppe, & Larson, 1991; Thatte & Xie, 2012; Yang et al., 2020)

$$T_i(k+1) = \bar{a}_{ii} T_i(k) + \sum_{j \in \bar{\mathcal{N}}_i} \bar{a}_{ij} T_j(k) + \bar{a}_{io} T_i^{\mathrm{out}}(k) - \frac{\eta_i \Delta}{C_i} P_i(k) + d_i(k), \tag{1}$$

where $T_i(k)$ and $T_i^{\mathrm{out}}(k)$ denote the indoor and outdoor temperatures at time slot k in °F, $d_i(k)$ is the thermal disturbances from some other energy sources such as internal loads and solar gains, $\bar{a}_{ii}$, $\bar{a}_{ij}$, and $\bar{a}_{io}$ are three parameters whose definitions are explained in detail in Appendix A, η_i denotes the coefficient of performance of the cooling system, Δ denotes the duration of a time slot in h with h indicating the unit hour, C_i represents the thermal capacitance of zone i in kJ/K, and $\bar{\mathcal{N}}_i$ denotes the set of physically neighboring zones that have thermal couplings with zone i. The subscript i denotes that all the variables and parameters are with respect to the subsystem i.

Remark 1. This remark is to explain the similarity and differences between $\mathcal{N}_i$, a_{ij}, $i, j \in \mathbb{Z}_1^{MN}$ and $\bar{\mathcal{N}}_i$, $\bar{a}_{ij}$, $i, j \in \mathbb{Z}_1^{MN}$. The similarity is that they are both related to the concepts of neighbors. However, $\mathcal{N}_i$ denotes the set of zones whose HAVC systems can communicate with zone i, in order to guarantee that the total energy consumption limit is not violated. $\bar{\mathcal{N}}_i$ represents the set of physically neighboring zones that have thermal couplings with zone i. The use of both $\mathcal{N}_i$ and $\bar{\mathcal{N}}_i$ is necessary, since it is unreasonable to assume that there are thermal couplings between two zones at different buildings, while communications are required between some zones at different buildings to guarantee that the bound for total power input is not violated. Similarly, such differences exist between a_{ij} and $\bar{a}_{ij}$. For example, a_{ij} can only be either 1 or 0, for all $i, j \in \mathbb{Z}_1^{MN}$, and $a_{ii} = 0$, for all $i \in \mathbb{Z}_1^{MN}$. However, $\bar{a}_{ij}$ can be any values in $[0, 1]$, and $\bar{a}_{ii}$ should not be zero for $i \in \mathbb{Z}_1^{MN}$ since it represents the coefficient of thermal inertia.

We assume that the thermal disturbance $d_i(k)$ is bounded. That is

Assumption 1. The thermal disturbance $d_i(k)$ is bounded for all $i \in \mathbb{Z}_1^{MN}$ and $k \in \mathbb{Z}_0^+$, i.e., there exist constants $d_i^{\min}(k)$ and $d_i^{\max}(k)$ such that

$$d_i^{\min}(k) \leq d_i(k) \leq d_i^{\max}(k). \tag{2}$$

Remark 2. Assumption 1 provides the possibility to design a distributed MPC algorithm such that the bounds for indoor temperature and power input are not violated in the presence of unknown thermal disturbance $d_i(k)$. The method used to handle the unknown disturbance is robust optimization, which will be presented in Section 3.2. In addition, Assumption 1 is reasonable since the main factors affecting the indoor temperature, for example, the indoor and outdoor temperatures at the previous time slot, the neighboring zones' indoor temperatures, are already included in model (1). What Assumption 1 established is that the sum of all other thermal sources is bounded. In the numerical simulations in this paper, the maximum power of thermal disturbance is 424.4 W for each zone. In Remark 11, we also give a detailed explanation about the thermal disturbance $d_i(k)$ in the numerical simulations.

One of the control objectives is to keep the indoor temperature within the appropriate range so that the occupants in the room will not feel uncomfortable (Xie et al., 2018). Thus, we need

$$T_i^{\min}(k) \leq T_i(k) \leq T_i^{\max}(k), \quad \forall k, \tag{3}$$

where $T_i^{\min}(k)$ and $T_i^{\max}(k)$ denote the lower and upper bounds of the indoor temperature of zone i at time slot k, respectively. In this study, we assume that $T_i^{\min}(k)$ and $T_i^{\max}(k)$ are certain and known. The reason is that if $T_i^{\min}(k)$ and $T_i^{\max}(k)$ are in some ranges and uncertain, then we can just use the worst $T_i^{\min}(k)$ and $T_i^{\max}(k)$, i.e., the highest $T_i^{\min}(k)$ and lowest $T_i^{\max}(k)$ to guarantee that the indoor temperature $T_i(k)$ always satisfies (3).

Besides, the constraints of the power inputs $P_i(k)$ can be described as

$$P_i^{\min}(k) \leq P_i(k) \leq P_i^{\max}(k), \quad \forall k, \tag{4}$$

where $P_i^{\min}(k)$ and $P_i^{\max}(k)$ are the lower and upper bounds of the power input of zone i at time slot k, respectively.

Moreover, we consider that there exists a global constraint for the total power consumption of all zones (Radhakrishnan et al., 2017; Zhang, Deng, Yuan, & Qin, 2017b), which is

$$\sum_{i=1}^{MN} P_i(k) \le \bar{P}(k), \quad \forall k, \tag{5}$$

with $\bar{P}(k)$ being the upper bound of the total power consumption at time slot k.

Remark 3. The local power constraint (4) is from the maximum power of an HVAC system. The motivation of the total power constraint (5) is stated as follows. Due to the power limit of the distribution infrastructure, the total power input for buildings should be constrained. We assume there is an aggregator that decides how much energy should be imported from the grid at each time slot for different power consumption systems of a building (Lampropoulos, Baghină, Kling, & Ribeiro, 2013; Nguyen & Le, 2013), such as HVAC system, water heater, and lighting system, etc. Thus, we can set total power limits for all power consumption systems in order to guarantee that the power limit of the grid will not be violated. Besides, this allocation can be time-varying for practical considerations. For example, in the numerical simulations in Section 5, we set $\bar{P}(k)$ to be smaller while the electricity price is higher and larger while the price is lower during a day.

2.3. Cost model

We consider two kinds of costs, which are the energy cost per hour of the cooling systems and the thermal discomfort cost associated with occupants. The cost function can be described as follows (Xie et al., 2018).

$$\ell_i(k) = S(k)P_i(k) + \sigma_i(k+1)\left(T_i(k+1) - T_i^{\text{ref}}(k+1)\right)^2, \quad \forall k, \tag{6}$$

where $S(k)$ denotes the electricity price at time slot k in \$/kWh, $T_i^{\text{ref}}(k+1)$ is the reference indoor temperature at time slot k reflecting the occupants' preference which is uncertain and in a range, and $\sigma_i(k+1)$ denotes the thermal coefficient cost in \$/(°F)^2 h at time slot $k+1$, which is assumed to be positive in this paper and serves as a trade-off coefficient between the energy saving and thermal comfort satisfaction. Note that $\sigma_i(k)$ is related to the occupancy status of zone i. For example, we can just set $\sigma_i(k)$ to be a small value at night when the building is closed and there are few occupants in zone i, and a large value in daytime when the building is open.

Remark 4. The reference indoor temperature $T_i^{\text{ref}}(k+1)$ can be determined by a comfort survey. Based on the PMV model, West et al. (2014) used an online thermal comfort survey form to inquire about the occupants' general thermal sensation and the satisfaction level for the indoor temperature. The calculated PMV can be used to tune the set-point temperature of zones in real time. Besides, we can also employ the reinforcement learning (RL) method to determine the reference indoor temperature (Fazenda, Veeramachaneni, Lima, & O'Reilly, 2014; Wei, Wang, & Zhu, 2017). When the occupants feel uncomfortable, they can give less rewards or more penalties to the corresponding state–action pairs of HVAC systems. The indoor temperature $T_i^{\text{ref}}(k+1)$ can be obtained and modified based on these related thermal feedbacks.

2.4. Problem formulation

In this paper, we would like to minimize total cost (6) of all MN zones over H slots, subject to constraints (3)–(5). To this end, the problem is formulated as

$$\min \sum_{i=1}^{MN}\sum_{k=1}^{H-1} \ell_i(k), \quad \text{s.t. (3)–(6)}, \tag{7}$$

with the decision variables being the power consumptions $P_i(k)$ for all $i \in \mathbb{Z}_1^{MN}$ and $k \in \mathbb{Z}_1^{H-1}$.

In order to solve (7), we need to know the outdoor temperature $T_i^{\text{out}}(k)$ and the reference indoor temperature over all $H-1$ time horizons in all MN zones. However, it is of large uncertainty to predict all these information at the initial time, which will deteriorate the control performance and enlarge the probabilities of feeling uncomfortable for occupants. To address this issue, we employ the MPC framework for which we only assume that the required information $T_i^{\text{out}}(k)$ and $T_i^{\text{ref}}(k+1)$ are accurate in the next h ($1 \le h \le H-1$) slots (Parisio, Rikos, & Glielmo, 2014; Xie et al., 2018). In this case, what we would like to minimize at the slot k_c is the cost $\sum_{i=1}^{MN}\sum_{k=k_c}^{k_c+h-2} \ell_i(k)$. After we obtain the power consumption $P_i(k_c)$, we can just substitute it into (1) to get the new state $T_i(k_c+1)$ and then we minimize $\sum_{i=1}^{MN}\sum_{k=k_c+1}^{k_c+h-1} \ell_i(k)$, subsequently. It should be noted that the minimization should repeat until $k_c = H-h+1$. For time slots from $H-h+2$ to $H-1$, we can directly use the remaining $h-2$ results computed at the slot $H-h+1$. Thus, the optimization problem can be written as

$$\min_{P_i(k)} \sum_{i=1}^{MN}\sum_{k=k_c}^{k_c+h-2} \ell_i(k), \quad \text{s.t. (3)–(6)}, \tag{8}$$

with k_c being from 1 to $H-h+1$.

2.5. Tightening the constraints

For optimization problem (8), we need to solve it numerically. However, little benefit can be gained when the numerical solution is really near optimal and the computing cost will be high if we continue the iteration (Wang & Ong, 2017). Besides, the subsequent optimization may be infeasible if we use an early termination condition in each step (Wang & Ong, 2018). To address this issue, we can tighten constraint (5) at each time slot $k_c \in \mathbb{Z}_1^{H-h+1}$ to account for errors arising from the premature termination, which is (Rubagotti, Patrinos, & Bemporad, 2014)

$$\sum_{i=1}^{MN} P_i(k) \le (1 - \epsilon MN(k - k_c + 1))\bar{P}(k), \quad \forall k \in \mathbb{Z}_{k_c}^{k_c+h-2}, \tag{9}$$

with ϵ being a predefined tolerance to the violation of the constraint (5) for a stopping criterion. The constraint (9) can be written in the vector form

$$\sum_{i=1}^{MN} \boldsymbol{P}_i(k_c) \le \boldsymbol{b}(k_c, \epsilon), \quad \forall k_c \in \mathbb{Z}_1^{H-h+1}, \tag{10}$$

with $\boldsymbol{P}_i(k_c) \triangleq [P_i(k_c), P_i(k_c+1), \ldots, P_i(k_c+h-2)]^{\mathrm{T}}$ and $\boldsymbol{b}(k_c,\epsilon) \triangleq [(1-\epsilon MN)\bar{P}(k_c), (1-2\epsilon MN)\bar{P}(k_c+1), \ldots, (1-(h-1)\epsilon MN)\bar{P}(k_c+h-2)]^{\mathrm{T}}$, respectively.

With global constraint (10), the optimization problem becomes

$$\min_{P_i(k)} \sum_{i=1}^{MN}\sum_{k=k_c}^{k_c+h-2} \ell_i(k), \quad \text{s.t. (3), (4), (6), (10)}. \tag{11}$$

Note that it will be complicated to directly solve optimization problem (11), due to the complicated objective function and thermal disturbances and parameter uncertainties existed in the problem. In Section 3, we will transform the objective function of (11) into a quadratic form and then show how the thermal disturbances and parameter uncertainties can be handled.

3. Transformation of optimization problem (11)

In this section, we firstly formulate problem (11) into one with a quadratic objective function with respect to $\boldsymbol{P}_i(k_c)$. Then, we will show how the additive thermal disturbances and uncertain parameters can be incorporated into the framework of distributed MPC in this work.

3.1. Transformation of problem (11)

In order to design a distributed MPC algorithm for HVAC systems considering thermal couplings between neighboring zones, we made the following assumption, with justification provided in Remark 13 of Appendix B.

Assumption 2. The impacts of initial indoor and outdoor temperatures, and power input for zone i, $i \in \mathbb{Z}_1^{MN}$, are only considered for zone i itself and its immediate neighbors. That is, they are neglected for the neighbors of neighbors of zone i.

If we denote

$$\boldsymbol{T}_i(k_c) \triangleq \left[T_i(k_c+1), T_i(k_c+2), \dots, T_i(k_c+h-1)\right]^{\mathrm{T}}, \tag{12a}$$

$$\boldsymbol{T}_i^{\mathrm{out}}(k_c) \triangleq \left[T_i^{\mathrm{out}}(k_c), T_i^{\mathrm{out}}(k_c+1), \dots, T_i^{\mathrm{out}}(k_c+h-2)\right]^{\mathrm{T}}, \tag{12b}$$

$$\boldsymbol{P}_i(k_c) \triangleq \left[P_i(k_c), P_i(k_c+1), \dots, P_i(k_c+h-2)\right]^{\mathrm{T}}, \tag{12c}$$

$$\boldsymbol{d}_i(k_c) \triangleq \left[d_i(k_c), d_i(k_c+1), \dots, d_i(k_c+h-2)\right]^{\mathrm{T}}, \tag{12d}$$

then the thermal dynamics of zone i can be formulated as

$$\boldsymbol{T}_i(k_c) = \boldsymbol{T}_{i0}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c), \tag{13}$$

where the matrices $\boldsymbol{F}_{ij}, \boldsymbol{\tau}_{ij}, \boldsymbol{G}_{ij} \in \mathbb{R}^{(h-1)\times(h-1)}$ and the vector $\boldsymbol{T}_{i0}(k_c) \in \mathbb{R}^{h-1}$ are constant matrices related to physical parameters and adjacency relationship of the zones in building systems, and the duration of each time slot. In addition, the vector $\boldsymbol{T}_{i0}(k_c)$ also depends on the indoor temperature $T_i(k_c)$ of zone i at time slot k_c. The definitions of $\boldsymbol{F}_{ij}, \boldsymbol{\tau}_{ij}, \boldsymbol{G}_{ij}, \boldsymbol{T}_{i0}(k_c)$, and the detailed formulation procedure of (13) can be found in Appendix B.

If we denote

$$\boldsymbol{T}_i^{\min}(k_c) \triangleq \left[T_i^{\min}(k_c+1), T_i^{\min}(k_c+2), \dots, T_i^{\min}(k_c+h-1)\right]^{\mathrm{T}}, \tag{14a}$$

$$\boldsymbol{T}_i^{\max}(k_c) \triangleq \left[T_i^{\max}(k_c+1), T_i^{\max}(k_c+2), \dots, T_i^{\max}(k_c+h-1)\right]^{\mathrm{T}}, \tag{14b}$$

$$\boldsymbol{P}_i^{\min}(k_c) \triangleq \left[P_i^{\min}(k_c), P_i^{\min}(k_c+1), \dots, P_i^{\min}(k_c+h-2)\right]^{\mathrm{T}}, \tag{14c}$$

$$\boldsymbol{P}_i^{\max}(k_c) \triangleq \left[P_i^{\max}(k_c), P_i^{\max}(k_c+1), \dots, P_i^{\max}(k_c+h-2)\right]^{\mathrm{T}}, \tag{14d}$$

the constraints (4) can be written as

$$\boldsymbol{T}_i^{\min}(k_c) \le \boldsymbol{T}_i(k_c) \le \boldsymbol{T}_i^{\max}(k_c), \tag{15}$$

$$\boldsymbol{P}_i^{\min}(k_c) \le \boldsymbol{P}_i(k_c) \le \boldsymbol{P}_i^{\max}(k_c), \tag{16}$$

with the inequalities being element-wise.

Substituting (13) into (15), and combining (16), we can obtain

$$\underbrace{\begin{bmatrix} \boldsymbol{I}_{h-1} \\ -\boldsymbol{I}_{h-1} \\ \boldsymbol{\tau}_{ii} \\ -\boldsymbol{\tau}_{ii} \end{bmatrix}}_{\boldsymbol{A}_i} \boldsymbol{P}_i(k_c) \le \underbrace{\begin{bmatrix} \boldsymbol{P}_i^{\max}(k_c) \\ -\boldsymbol{P}_i^{\min}(k_c) \\ \boldsymbol{T}_{i0}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c) - \boldsymbol{T}_i^{\min}(k_c) \\ -\boldsymbol{T}_{i0}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) - \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c) + \boldsymbol{T}_i^{\max}(k_c) \end{bmatrix}}_{\boldsymbol{B}_i}. \tag{17}$$

On the other hand, if we denote

$$\boldsymbol{S}(k_c) \triangleq \left[S(k_c), S(k_c+1), \dots, S(k_c+h-2)\right]^{\mathrm{T}}, \tag{18a}$$

$$\boldsymbol{\sigma}_i(k_c) \triangleq \begin{bmatrix} \sigma_i(k_c+1) & & & \\ & \sigma_i(k_c+2) & & \\ & & \ddots & \\ & & & \sigma_i(k_c+h-1) \end{bmatrix}, \tag{18b}$$

$$\boldsymbol{T}_i^{\mathrm{ref}}(k_c) \triangleq \left[T_i^{\mathrm{ref}}(k_c+1), T_i^{\mathrm{ref}}(k_c+2), \dots, T_i^{\mathrm{ref}}(k_c+h-1)\right]^{\mathrm{T}}, \tag{18c}$$

$$\boldsymbol{C}_i \triangleq \boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\boldsymbol{\tau}_{ii}, \tag{18d}$$

$$\boldsymbol{D}_i \triangleq \boldsymbol{S}(k_c) - 2\left(\left(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{T}_i^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\right)^{\mathrm{T}} \boldsymbol{\sigma}_i(k_c)\boldsymbol{\tau}_{ii}\right)^{\mathrm{T}}, \tag{18e}$$

$$\begin{aligned} E_i \triangleq{}& \left(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{T}_i^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\right)^{\mathrm{T}} \boldsymbol{\sigma}_i(k_c)\left(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{T}_i^{\mathrm{ref}}(k_c)\right. \\ &\left. + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\right), \end{aligned} \tag{18f}$$

then

$$\sum_{k=k_c}^{k_c+h-2} \ell_i(k) = \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c) + E_i. \tag{19}$$

The detailed information about how to get (19) is shown in Appendix C.

Thus, the optimization problem (11) can be formulated as

$$\min_{\boldsymbol{P}_i(k_c)} \sum_{i=1}^{MN} \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c) + E_i, \quad \text{s.t. (10), (17)}, \tag{20}$$

which is equivalent to

$$\min_{\boldsymbol{P}_i(k_c)} \sum_{i=1}^{MN} \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c), \quad \text{s.t. (10), (17)}, \tag{21}$$

since E_i is independent of $\boldsymbol{P}_i(k_c)$.

According to the form of $\boldsymbol{\tau}_{ii}$ indicated in (B.10), we know that $\boldsymbol{\tau}_{ii}$ is a nonsingular matrix. From (18b), we have that $\boldsymbol{\sigma}_i(k_c)$ is a diagonal matrix with all the diagonal elements $\sigma_i(k_c+1), \dots, \sigma_i(k_c+h-1)$ being positive, thus the matrix $\boldsymbol{C}_i \triangleq \boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\boldsymbol{\tau}_{ii}$ is positive definite. The positive definiteness of $\boldsymbol{C}_i$ is essential to prove that $\nabla_\lambda g_i(k_c, \lambda)$, which will be defined later and can be regarded as a function of λ, has Lipschitz continuous gradient. This is important for the accelerated distributed MPC algorithm design, in Section 4, which use the gradient projection method (see Section 6.10.1 of Bertsekas, 2009).

3.2. Formulation of uncertain optimization problem (21) to the certain counterpart

In Section 3.1, we formulated optimization problem (11) into the one with the objective function being quadratic. However, the obtained optimization problem (21) is uncertain, since the vector $\boldsymbol{D}_i$ in the objective and $\boldsymbol{B}_i$ in the constraint (17) are related to the uncertain parameter $T_i^{\mathrm{ref}}(k)$, $k \in \mathbb{Z}_{k_c+1}^{k_c+h-1}$, and the unknown thermal disturbances $d_i(k)$, $k \in \mathbb{Z}_{k_c}^{k_c+h-2}$, respectively. Thus, problem (21) cannot be directly solved. In this section, we transform (21) into a certain problem by using the concept of robust optimization (Ben-Tal, El Ghaoui, & Nemirovski, 2009).

The vector $\boldsymbol{T}_i^{\mathrm{ref}}(k_c)$ in (18c) is uncertain, since its corresponding elements are uncertain. In this study, we assume that $\boldsymbol{T}_i^{\mathrm{ref}}(k_c)$ is in the corresponding known ranges, i.e.,

$$\boldsymbol{\delta}_i^{\mathrm{ref}}(k_c) \le \boldsymbol{T}_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) \le \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c), \tag{22}$$

where $\boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c)$ is the nominal term, $\boldsymbol{\delta}_i^{\mathrm{ref}}(k_c)$ and $\boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c)$ are the corresponding lower and upper bounds of the true values' deviation from the nominal value $\boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c)$.

Besides, according to Assumption 1, we know that $\boldsymbol{d}_i(k_c)$ in (12d) is bounded, i.e.,

$$\tilde{\boldsymbol{d}}_i^{\min}(k_c) \le \boldsymbol{d}_i(k_c) \le \tilde{\boldsymbol{d}}_i^{\max}(k_c), \tag{23}$$

where $\tilde{\boldsymbol{d}}_i^{\min}(k_c) \triangleq \left[d_i^{\min}(k_c), d_i^{\min}(k_c+1), \dots, d_i^{\min}(k_c+h-2)\right]^{\mathrm{T}}$ and $\tilde{\boldsymbol{d}}_i^{\max}(k_c) \triangleq \left[d_i^{\max}(k_c), d_i^{\max}(k_c+1), \dots, d_i^{\max}(k_c+h-2)\right]^{\mathrm{T}}$. Since $\boldsymbol{G}(k_c)$ is a constant matrix, as defined in (B.10), there must exist vectors $\boldsymbol{d}_i^{\min}(k_c)$ and $\boldsymbol{d}_i^{\max}(k_c)$, such that

$$\boldsymbol{d}_i^{\min}(k_c) \le \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c) \le \boldsymbol{d}_i^{\max}(k_c). \tag{24}$$

It is shown in Appendix D that if we denote

$$\boldsymbol{B}_i^{\min} \triangleq \begin{bmatrix} \boldsymbol{P}_i^{\max}(k_c) \\ -\boldsymbol{P}_i^{\min}(k_c) \\ \boldsymbol{T}_{i0}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\mathcal{N}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\min}(k_c) - \boldsymbol{T}_i^{\min}(k_c) \\ -\boldsymbol{T}_{i0}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) - \boldsymbol{d}_i^{\max}(k_c) + \boldsymbol{T}_i^{\max}(k_c) \end{bmatrix}, \tag{25a}$$

$$\boldsymbol{D}_i^{\max} \triangleq \boldsymbol{S}(k_c) - 2\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i} \boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\min}(k_c)\Bigg), \tag{25b}$$

$$\mathcal{P}_i \triangleq \{\boldsymbol{P}_i(k_c) \mid \boldsymbol{A}_i\boldsymbol{P}_i(k_c) \le \boldsymbol{B}_i^{\min}, \forall k_c \in \mathbb{Z}_1^{H-h+1}\}, \tag{25c}$$

then the uncertain optimization problem (21) can be converted to the following certain optimization problem.

$$\min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \sum_{i=1}^{MN} \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c), \quad \text{s.t. (10)}. \tag{26}$$

Remark 5. The uncertain vectors $\boldsymbol{B}_i(k_c)$ and $\boldsymbol{D}_i(k_c)$ in (21) are replaced with their certain bounds $\boldsymbol{B}_i^{\min}(k_c)$ and $\boldsymbol{D}_i^{\max}(k_c)$ in (26). The intuitive reason of this replacement is that the worst case is considered when robust optimization procedure is used to transform the uncertain optimization problem to the certain counterpart. In Section 4, an accelerated distributed MPC algorithm will be proposed to solve the optimization problem (26).

4. Accelerated distributed model predictive control algorithm design

In this section, by using the Nesterov's gradient-projection algorithm, an accelerated distributed model predictive controller for HVAC systems with coupled constraints is designed.

4.1. The dual problem of (26)

Considering coupled constraints (10), the Lagrangian dual function of optimization problem (26) is

$$\begin{aligned}\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda) \triangleq{}& \sum_{i=1}^{MN} \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c) \\ &+ \lambda^{\mathrm{T}}\left(\sum_{i=1}^{MN}(\boldsymbol{P}_i(k_c) - \boldsymbol{b}(k_c,\epsilon))\right), \\ &\forall k_c \in \mathbb{Z}_1^{H-h+1}, \boldsymbol{P}_i(k_c) \in \mathcal{P}_i,\ i \in \mathbb{Z}_1^{MN},\end{aligned} \tag{27}$$

where $\lambda \ge \boldsymbol{0}_{h-1}$ is the dual variable of inequality constraint (10).

In the rest of this paper, we will omit $\forall k_c \in \mathbb{Z}_1^{H-h+1}$ and $\forall i \in \mathbb{Z}_1^{MN}$ for convenience if there is no confusion. We have the dual problem of (21) as

$$\max_{\lambda\ge\boldsymbol{0}_{h-1}} \min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \mathcal{L}(\boldsymbol{P}_i(k_c), \lambda), \tag{28}$$

and the primal optimization problem as

$$\min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \max_{\lambda\ge\boldsymbol{0}_{h-1}} \mathcal{L}(\boldsymbol{P}_i(k_c), \lambda), \tag{29}$$

with the primal function being $\max_{\lambda\ge\boldsymbol{0}_{h-1}} \mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$.

Lemma 1. *The objectives of dual optimization problem (28) and primal optimization problem (29) are equal, that is, the strong duality holds.*

Proof. Consider the Lagrangian dual function (27) $\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$. It is affine in λ, hence it is concave in λ. Besides, $\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$ is a continuous function with regard to λ, and $[0,\infty)^{h-1}$ is a convex and closed set, thus we know that $\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$ closed with regard to λ (see Section A.3.3 of Boyd & Vandenberghe, 2004). Thus, $-\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$ is convex and closed with regard to λ for all $\boldsymbol{P}_i(k_c) \in \mathcal{P}_i$.

From (19), we know that $\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c)+\boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)$ is a quadratic function of $\boldsymbol{P}_i(k_c)$, with $\boldsymbol{C}_i$ being positive definite. According to Example 3.2 of Boyd and Vandenberghe (2004), we have that $\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)$ is strictly convex with regard to $\boldsymbol{P}_i(k_c)$. Then, it is easy to see that $\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$ is also convex in $\boldsymbol{P}_i(k_c)$. Similarly, $\boldsymbol{P}_i(k_c)$ is a convex and closed set from (17), and $\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$ is continuous in $\boldsymbol{P}_i(k_c)$, thus $\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda)$ is convex and closed in $\boldsymbol{P}_i(k_c)$ for every $\lambda \ge \boldsymbol{0}_{h-1}$.

Now we consider the function

$$p(\boldsymbol{u}) = \min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \max_{\lambda\ge\boldsymbol{0}_{h-1}} \left(\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda) - \boldsymbol{u}^{\mathrm{T}}\lambda\right), \quad \boldsymbol{u} \in \mathbb{R}^{h-1}, \tag{30}$$

which is a continuous function in $\boldsymbol{u}$. Combining with the fact that $\mathbb{R}^{h-1}$ is a closed set, we know that $p(\boldsymbol{u})$ is a closed function (see Section 9.1 of Boyd & Vandenberghe, 2004). According to Proposition 1.1.2 in Bertsekas (2009), we have that $p(\boldsymbol{u})$ is lower semicontinuous. Since $p(\boldsymbol{0}) < \infty$, we can conclude

$$\max_{\lambda\ge\boldsymbol{0}_{h-1}} \min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \mathcal{L}(\boldsymbol{P}_i(k_c), \lambda) = \min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \max_{\lambda\ge\boldsymbol{0}_{h-1}} \mathcal{L}(\boldsymbol{P}_i(k_c), \lambda), \tag{31}$$

from Proposition 5.5.1 of Bertsekas (2009).

The dual problem (28) is equivalent to

$$\min_{\lambda\ge\boldsymbol{0}_{h-1}} \max_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} -\mathcal{L}(\boldsymbol{P}_i(k_c), \lambda) = \min_{\lambda\ge\boldsymbol{0}_{h-1}} \sum_{i=1}^{MN} g_i(k_c, \lambda), \tag{32}$$

with

$$\begin{aligned} g_i(k_c,\lambda) \triangleq{}& \max_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} -\left(\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)\right) \\ &- \lambda^{\mathrm{T}}\left(\boldsymbol{P}_i(k_c) - \frac{1}{MN}\boldsymbol{b}(k_c,\epsilon)\right). \end{aligned} \tag{33}$$

It is easy to see that $-\left(\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)\right) - \lambda^{\mathrm{T}}\left(\boldsymbol{P}_i(k_c) - \frac{1}{MN}\boldsymbol{b}(k_c,\epsilon)\right)$ is affine in λ, and thus it is a convex function with regard to λ. According to the extended pointwise maximum property (see Section 3.2.3 in Boyd & Vandenberghe, 2004), we can conclude that $g_i(k_c,\lambda)$ is a convex function with regard to λ.

Letting $\boldsymbol{P}_i(k_c,\lambda) \triangleq \arg\max_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} g_i(k_c,\lambda)$, we have

$$\begin{aligned} g_i(k_c,\lambda) ={}& -\left(\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)\right) - \lambda^{\mathrm{T}}\Big(\boldsymbol{P}_i(k_c,\lambda) \\ &- \frac{1}{MN}\boldsymbol{b}(k_c,\epsilon)\Big). \end{aligned} \tag{34}$$

According to Danskin's theorem for maximum functions (see Section 3.1.1 in Bertsekas, 2015), we have

$$\nabla_\lambda g_i(k_c,\lambda) = -\left(\boldsymbol{P}_i(k_c,\lambda) - \frac{1}{MN}\boldsymbol{b}(k_c,\epsilon)\right). \tag{35}$$

According to (19), we have

$$\nabla^2_{\boldsymbol{P}_i(k_c,\lambda)}\left(\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)\right) = 2\boldsymbol{C}_i > 0, \tag{36}$$

which indicates that there exists $\mu_i > 0$ such that $\nabla^2_{\boldsymbol{P}_i(k_c,\lambda)}\left(\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)\right) \ge \mu_i\boldsymbol{I}_{h-1}$. According to Theorem 1 in Nesterov

(2005), one has that $\nabla_\lambda g_i(k_c,\lambda)$ is Lipschitz continuous with constant $\frac{\sqrt{N}}{\mu_i}$. We denote $L_g \triangleq \max_{i\in\mathbb{Z}_1^{MN}} \frac{\sqrt{N}}{\mu_i}$ in the rest of this paper.

4.2. Distributed fast dual gradient algorithm for HVAC system

According to Section 6.10 of Bertsekas (2009) and Algorithm 1 in Patrinos and Bemporad (2014), optimization problem (32) can be solved by the following iterations.

$$\tilde{\lambda}^j = \lambda^j + \theta^j\left((\theta^{j-1})^{-1}-1\right)\left(\lambda^j-\lambda^{j-1}\right), \tag{37a}$$

$$\lambda^{j+1} = \left[\tilde{\lambda}^j - \frac{1}{L_g}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}^j)\right]_+, \tag{37b}$$

$$\theta^{j+1} = \frac{\sqrt{\left(\theta^j\right)^4+4\left(\theta^j\right)^2}-\left(\theta^j\right)^2}{2}. \tag{37c}$$

with the initial values $\lambda^{-1}=\lambda^0=\mathbf{0}_{h-1}$ and $\theta^{-1}=\theta^0=1$. It is easy to check that (37c) satisfies the following relations.

$$\frac{1-\theta^{j+1}}{\left(\theta^{j+1}\right)^2} = \frac{1}{\left(\theta^j\right)^2}, \frac{1}{\left(\theta^j\right)^2} = \sum_{\ell=0}^{j}\left(\theta^\ell\right)^{-1}, \theta^j \le \frac{2}{j+2}, \tag{38}$$

for $j>0$.

The variable λ^j in (37a) and (37b) is a global variable. Since what we need to design is a distributed MPC controller, we can make a copy of λ^j, denoted by λ_i^j, for subsystem i, $i\in\mathbb{Z}_1^{MN}$. The result is

$$\tilde{\lambda}_i^j = \lambda_i^j + \theta^j\left((\theta^{j-1})^{-1}-1\right)\left(\lambda_i^j-\lambda_i^{j-1}\right), \tag{39a}$$

$$\lambda_i^{j+1} = \left[\tilde{\lambda}_i^j - \frac{1}{L_g}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)\right]_+, \tag{39b}$$

where $\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j) = -\left(\tilde{\boldsymbol{P}}_i^j(k_c)-\frac{1}{MN}\boldsymbol{b}(k_c,\epsilon)\right)$ with

$$\begin{aligned}\tilde{\boldsymbol{P}}_i^j(k_c) &\triangleq \arg\min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i}\ \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c)+\boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)\\ &\qquad + \tilde{\lambda}_i^{j\mathrm{T}}\left(\boldsymbol{P}_i(k_c)-\frac{1}{MN}\boldsymbol{b}(k_c,\epsilon)\right)\\ &= \arg\min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i}\sum_{k=k_c}^{k_c+h-2}\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c)+(\boldsymbol{D}_i^{\max}+\tilde{\lambda}_i^j)^{\mathrm{T}}\boldsymbol{P}_i(k_c).\end{aligned} \tag{40}$$

Remark 6. We use iterations (39a) and (39b) instead of (37a) and (37b) in the aim of making the controller distributed. As long as λ_i^{-1} and λ_i^0 are the same for all $i\in\mathbb{Z}_1^{MN}$ respectively, we can obtain $\lambda_1^j=\lambda_2^j=\cdots=\lambda_{MN}^j$ for all iteration step j. Thus, the consensus of λ_i, $i\in\mathbb{Z}_1^{MN}$ can be guaranteed at all iteration steps. However, it should be noted that we still need the information of $\frac{1}{L_g}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)$ in (39b), which is related to all MN subsystems and thus makes the controller not fully distributed. In Section 4.3, a distributed averaging consensus algorithm will be introduced to handle this problem.

Define

$$\bar{\boldsymbol{P}}_i^j(k_c) \triangleq (\theta^j)^2\sum_{\ell=0}^{j}(\theta^\ell)^{-1}\tilde{\boldsymbol{P}}_i^\ell(k_c) = (1-\theta^j)\bar{\boldsymbol{P}}_i^{j-1}(k_c)+\theta^j\tilde{\boldsymbol{P}}_i^j(k_c), \tag{41}$$

for all $i\in\mathbb{Z}_1^{MN}$ and with the initial condition $\bar{\boldsymbol{P}}_i^{-1}(k_c)=\mathbf{0}_{h-1}$.

According to the results of Wang and Ong (2018), supposing we terminate the iteration of (41) at step $\bar{j}$, then we can get $\bar{\boldsymbol{P}}_i^{\bar{j}}(k_c)$ which can be written as $\bar{\boldsymbol{P}}_i^{\bar{j}}(k_c)\triangleq\left[\bar{P}_{i,1}^{\bar{j}}(k_c),\dots,\bar{P}_{i,h-1}^{\bar{j}}(k_c)\right]^{\mathrm{T}}$. The MPC algorithm applied to the ith subsystem at time slot k_c can be designed as

$$P_i(k_c) = \bar{P}_{i,1}^{\bar{j}}(k_c), \forall k_c\in\mathbb{Z}_1^{H-h+1}, \forall i\in\mathbb{Z}_1^{MN}. \tag{42}$$

4.3. Distributed average consensus

The aim of this subsection is to compute the term $\frac{1}{L_g}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)$ in a distributed manner. Note that we can rewrite $\frac{1}{L_g}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)$ as

$$\frac{1}{L_g}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j) = \frac{MN}{L_g}\frac{1}{MN}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j), \tag{43}$$

with $\frac{1}{MN}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)$ being the average of $\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)$ over MN subsystems. Thus, we can employ a distributed average consensus algorithm to compute $\frac{1}{MN}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c,\tilde{\lambda}_i^j)$ distributedly.

To that end, we introduce a distributed average consensus method in this subsection. Besides, another distributed protocol is also used to determine whether the average consensus is achieved with a predefined tolerance at certain iteration step.

Consider the average consensus of MN vectors $\boldsymbol{x}_1,\dots,\boldsymbol{x}_{MN}$, with initial values $\boldsymbol{x}_1(0),\dots,\boldsymbol{x}_{MN}(0)$, respectively. A distributed average consensus algorithm can be formulated as (Olfati-Saber, Fax, & Murray, 2007; Saber & Murray, 2003)

$$\boldsymbol{x}_i(k'+1) = \boldsymbol{x}_i(k')+\bar{\epsilon}\sum_{j\in\mathcal{N}_i}a_{ij}\left(\boldsymbol{x}_j(k')-\boldsymbol{x}_i(k')\right), \tag{44}$$

for all $i\in\mathbb{Z}_1^{MN}$, and $0<\bar{\epsilon}<\frac{1}{\max_i d_i}$ being the step-size. According to the related analysis in Olfati-Saber et al. (2007) and Saber and Murray (2003) (the difference is that $\boldsymbol{x}_i$ is a vector-valued in our case while it is a scalar in Olfati-Saber et al., 2007; Saber & Murray, 2003), we know that

$$\lim_{k'\to\infty}\boldsymbol{x}_i(k') = \frac{1}{MN}\sum_{i=1}^{MN}\boldsymbol{x}_i(0), \quad \forall i\in\mathbb{Z}_1^{MN}, \tag{45}$$

as long as the undirected graph G is connected.

It is indicated in (45) that the distributed average consensus of vectors $\boldsymbol{x}_1,\dots,\boldsymbol{x}_{MN}$ can be achieved by (44) asymptotically. However, when putting (44) into implementation, we should have a criterion to determine whether the average consensus is achieved with a predefined tolerance at certain iteration step k''. To this end, we firstly introduce the maximum consensus and minimum consensus protocols as follows (Yadav & Salapaka, 2007).

Consider MN vectors $\boldsymbol{y}_1,\dots,\boldsymbol{y}_{MN}$, with initial values $\boldsymbol{y}_1(0),\dots,\boldsymbol{y}_{MN}(0)$, respectively. The algorithm

$$\boldsymbol{y}_i(k+1) = \max_{j\in\mathcal{N}_i}\boldsymbol{y}_j(k), \tag{46}$$

with max being the element-wise operator, can achieve the finite-time distributed maximum consensus. That is

$$\boldsymbol{y}_i(k') = \max_i \boldsymbol{y}_i(0), \quad \forall i\in\mathbb{Z}_1^{MN}, \tag{47}$$

for some $\bar{k}'$ such that $k'\ge\bar{k}'$.

Similarly, the algorithm

$$\boldsymbol{y}_i(k+1) = \min_{j\in\mathcal{N}_i}\boldsymbol{y}_j(k), \tag{48}$$

with min being the element-wise operator, can achieve the finite-time distributed minimum consensus. That is

$$\boldsymbol{y}_i(k') = \min_i \boldsymbol{y}_i(0), \quad \forall i\in\mathbb{Z}_1^{MN}, \tag{49}$$

for some $\underline{k}'$ such that $k'\ge\underline{k}'$.

According to Yadav and Salapaka (2007), we have that both $\bar{k}'$ and $\underline{k}'$ are less than or equal to the diameter D of the graph. Thus, we can just let $\bar{k}'=\underline{k}'=D$. Then Algorithm 1 can be used to compute the average of $\boldsymbol{x}_1(0),\dots,\boldsymbol{x}_{MN}(0)$ with the prescribed margin error ρ.

In Algorithm 1, the termination criterion of distributed average consensus is $\left\|\max_i\boldsymbol{x}_i(\ell)-\min_i\boldsymbol{x}_i(\ell)\right\|_2<\rho$ after the maximum and minimum consensus are both realized. We just use the value of the first

Algorithm 1 Distributed average consensus with marginal error ρ.

1: **Input:** $\boldsymbol{x}_i(0)$, $i \in \mathbb{Z}_1^{MN}$, the predefined marginal error ρ, the diameter D and the adjacency matrix $\boldsymbol{A} \triangleq [a_{ij}]$, $i, j \in \mathbb{Z}_1^{MN}$, of graph G, the step size $\bar{\epsilon}$
2: **Output:** a vector $\bar{\boldsymbol{x}}$ such that $\left\|\bar{\boldsymbol{x}} - \frac{1}{MN}\sum_{i=1}^{MN}\boldsymbol{x}_i(0)\right\|_2 < \rho$
3: *Initialization:* set $\ell = 0$, $k = 0$, $\bar{\boldsymbol{x}} = \boldsymbol{0}_{h-1}$, $\delta = \rho + 1$;
4: **while** $\delta \geq \rho$ (in parallel) **do**
5: $\quad \boldsymbol{x}_i(\ell+1) \leftarrow \boldsymbol{x}_i(\ell) + \bar{\epsilon}\sum_{j=1}^{MN} a_{ij}\left(\boldsymbol{x}_j(\ell) - \boldsymbol{x}_i(\ell)\right)$
6: $\quad \boldsymbol{y}_i(0) \leftarrow \boldsymbol{x}_i(\ell+1)$, $\boldsymbol{z}_i(0) \leftarrow \boldsymbol{x}_i(\ell+1)$, $i \in \mathbb{Z}_1^{MN}$;
7: $\quad$ **while** $k < D$ **do**
8: $\qquad \boldsymbol{y}_i(k+1) \leftarrow \max_{j\in\mathcal{N}_i}\boldsymbol{y}_j(k)$;
9: $\qquad \boldsymbol{z}_i(k+1) \leftarrow \min_{j\in\mathcal{N}_i}\boldsymbol{z}_j(k)$;
10: $\qquad k \leftarrow k+1$;
11: $\quad$ **end while**
12: $\quad \delta \leftarrow \|\boldsymbol{y}_1(D) - \boldsymbol{z}_1(D)\|_2$;
13: $\quad \ell \leftarrow \ell + 1$;
14: **end while**
15: $\bar{\boldsymbol{x}} \leftarrow \boldsymbol{x}_1(\ell)$

subsystem to compute $\|\max_i \boldsymbol{x}_i(\ell) - \min_i \boldsymbol{x}_i(\ell)\|_2 < \rho$ in Line 12 since we know that $\boldsymbol{y}_1((j+1)D) = \max_i \boldsymbol{x}_i(\ell+1)$ and $\boldsymbol{z}_1((j+1)D) = \min_i \boldsymbol{x}_i(\ell+1)$. Note that

$$\left|\max_i \boldsymbol{x}_i(\ell) - \min_i \boldsymbol{x}_i(\ell)\right| \geq \left|\boldsymbol{x}_1(\ell) - \frac{1}{MN}\sum_{i=1}^{MN}\boldsymbol{x}_i(0)\right|, \tag{50}$$

where both the absolute operator and the inequality are element-wise. Thus, one has

$$\begin{aligned}\left\|\bar{\boldsymbol{x}} - \frac{1}{MN}\sum_{i=1}^{MN}\boldsymbol{x}_i(0)\right\|_2 &= \left\|\boldsymbol{x}_1(\ell) - \frac{1}{MN}\sum_{i=1}^{MN}\boldsymbol{x}_i(0)\right\|_2 \\ &\leq \left\|\max_i \boldsymbol{x}_i(\ell) - \min_i \boldsymbol{x}_i(\ell)\right\|_2 \\ &< \rho. \end{aligned}\tag{51}$$

It should be noted that we can actually let $\bar{\boldsymbol{x}} = \boldsymbol{x}_i(\ell)$ for any $i \in \mathbb{Z}_1^{MN}$ and we just pick $\bar{\boldsymbol{x}} = \boldsymbol{x}_1(\ell)$ without loss of generality.

Now we consider to compute $\frac{1}{MN}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c, \tilde{\lambda}_i^j)$. We can just let $\boldsymbol{x}_i(0) \leftarrow \nabla_\lambda g_i(k_c, \tilde{\lambda}_i^j)$ and then use Algorithm 1 to compute $\frac{1}{MN}\sum_{i=1}^{MN}\nabla_\lambda g_i(k_c, \tilde{\lambda}_i^j)$ distributedly. Thus, we can use iterations (39a), (39b), and (37c) to solve optimization problem (32) in a distributed manner. According to the strong duality indicated in Lemma 1, we actually have solved constrained optimization problem (11) in a distributed way. From Theorem 1 in Wang and Ong (2018) and Proposition 6.10.3 in Bertsekas (2009), we have the following statement.

Let the feasible domain of optimization problem (21) be $\mathcal{D}_\epsilon \triangleq \{\boldsymbol{T}_{i0} \in \mathbb{R}^h : \text{(21) is feasible}\}$. For any $\boldsymbol{T}_{i0} \in \mathcal{D}_\epsilon$, we denote the optimal solution of (21) as $\{\boldsymbol{P}_i^*(k_c)\}_{i=1}^{MN}$ and one of the optimal dual variable as λ^*. Further, let $\bar{J}^j(k_c) \triangleq \sum_{i=1}^{MN}\bar{\boldsymbol{P}}_i^{j\mathrm{T}}(k_c)\boldsymbol{C}_i\bar{\boldsymbol{P}}_i^j(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\bar{\boldsymbol{P}}_i^j(k_c)$ and $\boldsymbol{J}^*(k_c) \triangleq \sum_{i=1}^{MN}\boldsymbol{P}_i^{*\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i^*(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i^*(k_c)$. Then, for any $\boldsymbol{T}_{i0} \in \mathcal{D}_\epsilon$, $j \geq 0$, we have

$$-\frac{4MN(\sqrt{MN}+1)L_g\|\lambda^*\|_2^2}{(j+2)^2} \leq \bar{J}^j(k_c) - \boldsymbol{J}^*(k_c) \leq 0. \tag{52}$$

Remark 7. Eq. (52) indicates that the designed algorithm can achieve a convergence rate of $O\left(\frac{1}{j^2}\right)$, which is accelerated compared with those whose convergence rate is of $O\left(\frac{1}{j}\right)$, such as the one in Xie et al. (2018) (the rigorous proof can be found in Wang & Ong, 2017). However, although the convergence rate is accelerated, we still need a stopping criterion to determine when we should terminate the iterations (39a), (39b), and (37c). In the next section, we will show how we can handle this issue in a distributed manner.

4.4. Stopping criterion

Since the stopping criterion is with respect to whether we should stop the iterations when obtaining $\bar{\boldsymbol{P}}_i^j(k_c)$ at step j, we only consider the MPC slot $k = k_c$ in this section without loss of generality. Let us firstly introduce the definitions about ϵ-solution and (ϵ, δ)-suboptimal solution of (21).

Definition 1. Given any $\epsilon > 0$, the set $\{\boldsymbol{P}_i(k_c)\}_{i=1}^{MN}$ is an ϵ-relaxed solution of (21) if $\boldsymbol{P}_i(k_c) \in \mathcal{P}_i$, $\forall i \in \mathbb{Z}_1^{MN}$, and

$$\sum_{i=1}^{MN}\boldsymbol{P}_i(k_c) - \boldsymbol{b}(k_c, \epsilon) \leq \epsilon MN\bar{\rho}(k_c), \tag{53}$$

where $\bar{\rho}(k_c) \triangleq [\bar{P}(k_c), \bar{P}(k_c+1), \ldots, \bar{P}(k_c+h-2)]^{\mathrm{T}}$.

Given any $\epsilon, \delta > 0$, $\{\boldsymbol{P}_i(k_c)\}_{i=1}^{MN}$ is an (ϵ, δ)-suboptimal solution of (21) if it is an ϵ-related solution and

$$J(k_c) - \boldsymbol{J}^*(k_c) \leq \delta, \tag{54}$$

where $J(k_c) \triangleq \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c)$.

Similar to property (iii) of Lemma 1 in Wang and Ong (2018), we have

$$\sum_{i=1}^{MN}\bar{\boldsymbol{P}}_i^j(k_c) - \boldsymbol{b}(k_c, \epsilon) \leq \frac{4MN(\sqrt{MN}+1)L_g\|\lambda^*\|_2}{(j+2)^2}\bar{\rho}(k_c). \tag{55}$$

Thus, there always exists j such that $\{\bar{\boldsymbol{P}}_i^j(k_c)\}_{i=1}^{MN}$ is an ϵ-relaxed solution of (21). Besides, according to (52), we know that such a solution $\{\bar{\boldsymbol{P}}_i^j(k_c)\}_{i=1}^{MN}$ must be an $(\epsilon, 0)$-suboptimal solution. Thus, a stopping criterion could be $\sum_{i=1}^{MN}\bar{\boldsymbol{P}}_i^j(k_c) - \boldsymbol{b}(k_c, \epsilon) \leq \epsilon MN\bar{\rho}(k_c)$. Since $\boldsymbol{b}(k_c, \epsilon) \triangleq [(1-\epsilon MN)\bar{P}(k_c), (1-2\epsilon MN)\bar{P}(k_c+1), \ldots, (1-(h-1)\epsilon MN)\bar{P}(k_c+h-2)]^{\mathrm{T}}$, this stopping condition could be written as

$$\sum_{i=1}^{MN}\bar{\boldsymbol{P}}_i^j(k_c) \leq \begin{bmatrix}\bar{P}(k_c) \\ (1-\epsilon MN)\bar{P}(k_c+1) \\ \vdots \\ (1-(h-2)\epsilon MN)\bar{P}(k_c+h-2)\end{bmatrix}. \tag{56}$$

Remark 8. We now illustrate the function of tightening the constraints in Section 2.5. By (9), we can guarantee that $P_i(k_c)$, namely the first element of $\boldsymbol{P}_i(k_c)$, will be no larger than $\bar{P}(k_c)$. From (42), we know that only the first element of $\boldsymbol{P}_i(k_c)$ will be implemented during the MPC framework, thus the coupled constraints (5) will not be violated during the control process.

Remark 9. We can use Algorithm 1 to compute $\sum_{i=1}^{MN}\bar{\boldsymbol{P}}_i^j(k_c)$ for the stopping criterion $\sum_{i=1}^{MN}\bar{\boldsymbol{P}}_i^j(k_c) - \boldsymbol{b}(k_c, \epsilon) \leq \epsilon MN\bar{\boldsymbol{\rho}}$.

The accelerated distributed model predictive controller for HVAC systems with coupled constraints is summarized in Algorithm 2.

Remark 10. This remark is to illustrate how the possible conflict between the power and indoor temperature constraints can be addressed when implementing the proposed algorithm. The possible conflict of these two constraints is a characteristic of MPC when we consider them both (see Section 2.5.4 of the book Wang, 2009). (1) When the constraints of local input bound and the indoor temperature are conflicting, the quadratic programming problem, i.e., the line 13 of Algorithm 2, will be infeasible. In this case, we can slack the bound of the indoor temperature at the time slot when the conflict occurs. Thus, the thermal comfort level might be sacrificed. (2) When the constraints of the total power input and indoor temperature are conflicting, the stopping criterion will never be satisfied. In order to step out of the while loop of Algorithm 2, the maximum iteration number can be predefined. In this case, the global power constraints might be violated, which means more power would be allocated to the HVAC systems in the building system. However, the MPC algorithm will try its best to avoid conflicts of constraints, since the optimization is over a finite future time horizon instead of only the current time slot. This constitutes one advantage of the model predictive controller.

Algorithm 2 The accelerated distributed MPC for HVAC systems.

Input: C_i, R_{io}, R_{ij}, η_i, $d_i(k)$, Δ, $T_i(0)$, $T_i^{\text{out}}(k)$, $T_i^{\text{ref}}(k)$, $S(k)$, $\sigma_i(k)$, $T_i^{\min}(k)$, $T_i^{\max}(k)$, $P_i^{\min}(k)$, $P_i^{\max}(k)$, $\boldsymbol{A} \triangleq [a_{ij}]$, ϵ, $\bar{P}(k)$, h, L_g, $i, j \in \mathbb{Z}_1^{MN}$, $k \in \mathbb{Z}_1^H$
Output: $\bar{\boldsymbol{P}}_i^{\bar{j}}(k)$, $\forall i \in \mathbb{Z}_1^{MN}$, $\forall k \in \mathbb{Z}_1^{H-h+1}$
Initialization: $k_c = 1$, $\bar{\boldsymbol{P}}_i^{-1}(k) = \mathbf{0}_{h-1}$, $\forall i \in \mathbb{Z}_1^{MN}$, $\forall k \in \mathbb{Z}_1^{H-h+1}$
for all $i \in \mathbb{Z}_1^{MN}$ (in parallel) **do**
  while $k_c \le H - h + 1$ **do**
    $j \leftarrow 0$;
    $\lambda_i^{-1} = \lambda_i^0 \leftarrow \mathbf{0}_{h-1}$;
    $\theta^{-1} = \theta^0 \leftarrow 1$;
    $\delta \leftarrow \epsilon M N \bar{\rho} + \mathbf{1}_{h-1}$;
    while $\delta > \epsilon M N \bar{\rho}$ **do**
      $\tilde{\lambda}_i^j \leftarrow \lambda_i^j + \theta^j \left((\theta^{j-1})^{-1} - 1\right)\left(\lambda_i^j - \lambda_i^{j-1}\right)$;
      $\lambda_i^{j+1} \leftarrow \left[\tilde{\lambda}_i^j - \frac{1}{L_g} \sum_{i=1}^{MN} \nabla_\lambda g_i(k_c, \tilde{\lambda}_i^j)\right]_+$;
      $\tilde{\boldsymbol{P}}_i^j(k_c) \leftarrow \arg\min_{\boldsymbol{P}_i(k_c) \in \mathcal{P}_i} \sum_{k=k_c}^{k_c+h-2} \boldsymbol{P}_i^{\mathrm{T}}(k_c) \boldsymbol{C}_i \boldsymbol{P}_i(k_c) + (\boldsymbol{D}_i^{\max} + \tilde{\lambda}_i^j)^{\mathrm{T}} \boldsymbol{P}_i(k_c)$;
      $\bar{\boldsymbol{P}}_i^j(k_c) \leftarrow (1-\theta^j)\bar{\boldsymbol{P}}_i^{j-1}(k_c) + \theta^j \tilde{\boldsymbol{P}}_i^j(k_c)$;
      $\delta \leftarrow \sum_{i=1}^{MN} \bar{\boldsymbol{P}}_i^j(k_c) - \boldsymbol{b}(k_c, \epsilon)$;
      $\theta^{j+1} \leftarrow \frac{\sqrt{(\theta^j)^4 + 4(\theta^j)^2} - (\theta^j)^2}{2}$;
      $j \leftarrow j + 1$;
    end while
    $\bar{j} \leftarrow j$;
    $P_i(k_c) \leftarrow \bar{P}_{i,1}^{\bar{j}}(k_c)$;
    $T_i(k_c+1) \leftarrow \bar{a}_{ii} T_i(k) + \sum_{j \in \bar{\mathcal{N}}_i} \bar{a}_{ij} T_j(k_c) + \bar{a}_{io} T_i^{\text{out}}(k_c) - \frac{\eta_i \Delta}{C_i} P_i(k_c) + d_i(k_c)$;
    $k_c \leftarrow k_c + 1$
  end while
end for

5. Numerical simulations

In this section, the numerical simulations will be used to demonstrate the effectiveness and advantages of the algorithms in this paper. We use the similar simulation framework as that in Xie et al. (2018) to compare the results and show the advantages. We consider 5 buildings with each building comprising 10 distinct zones. In the distributed cases, there is an independent HVAC system in each zone and the neighborhood relationship of these independent systems is shown in Fig. 1. The physical neighboring relationship, i.e., the thermal coupling relationship, of these 50 zones are assumed to be the same as the neighborhood relationship of the HVAC systems, expect that there are no thermal couplings between two zones at different buildings, such as zones 10 and 15 in Fig. 1. Note that we do not need the building message controllers and the multi-building coordinator since we use a fully distributed method to determine when to stop the iteration. The MATLAB R2019a, running on a laptop with an Intel Core i7-4770 and 32.0 GB RAM, is used.

5.1. Simulation setup and results of the proposed distributed algorithm

We consider that there are $M = 5$ buildings, and each building has $N = 10$ zones. The parameters of zones are shown in Table 1, for all $i \in \mathbb{Z}_1^{MN}$ and $k \in \mathbb{Z}_1^H$ (Constantopoulos et al., 1991; Yang et al., 2020). The duration of a time slot is selected as $\Delta = 0.2$h, and the simulation lasts for 48 h, i.e., $H = \frac{48}{0.2} = 240$.

Table 1
System parameters.

Parameter	Value	Unit
C_i	1.375×10^3	kJ/K
R_{io}	50	K/kW
R_{ij}	14	K/kW
η_i	4.5	–
$d_i(k)$	$\mathcal{U}(-0.2, 0.2)$	°F

The other simulation parameters are set as $h = 8$, $\epsilon = \frac{0.001}{MN}$, $\bar{\epsilon} = 0.25$, $\rho = 0.03$, and $L_g = 200$. According to Fig. 1, we have $D = M\left(\frac{N}{2} + 1\right) - 1 = 29$. Besides, we have $T_i^{\text{ref}}(k) = (71 + \mathcal{U}(-0.5, 0.5))$ °F, $P_i^{\min}(k) = 0$, $P_i^{\max}(k) = 1$ kW, and

$$\bar{P}(k) = \begin{cases} 0.5MN \text{ kW}, & \text{if } S(k) = \$0.0808/\text{kWh}, \\ 0.3MN \text{ kW}, & \text{if } S(k) = \$0.1692/\text{kWh}, \end{cases}$$

for all $k \in \mathbb{Z}_1^H$, $i \in \mathbb{Z}_1^{MN}$. We assume the buildings 1 and 2 open at 9 am and close at 5 pm and the buildings 3 to 5 open at 8 am and close at 8 pm, respectively. When the buildings are open, we set $\sigma_i(k) = 3$, $T_i^{\min}(k) = 65$ °F, $T_i^{\max}(k) = 78$ °F, and when the buildings are closed, we set $\sigma_i(k') = \frac{0.04}{(k'+2)^2}$, $T_i^{\min}(k) = 65$ °F, $T_i^{\max}(k) = 85$ °F, with k' being the time since the nearest closing. We assume that the initial indoor temperature of the zones labeled with odd numbers is 73 °F while the initial indoor temperature of the zones labeled with even numbers is 74°F. We use the outdoor temperature in Atlanta from July 3 to July 4, 2020 (Atlanta, 0000). Note that we can only get the outdoor temperature information every hour, while the duration of one time slot is $\Delta = 0.2$h. To address this problem, the spline interpolation is used to generate the data between adjacent hours. Thus, the outdoor temperature used in this section is shown in Fig. 2. We consider that the buildings are utilized commercially and the electricity price $S(k)$, $k \in \mathbb{Z}_1^H$ is shown in Fig. 3. In order to compare the simulation results with those in Xie et al. (2018) and of the centralized algorithm, we also obtain the simulation results by using the distributed ADMM in Xie et al. (2018) with Algorithm 1 to determine when to stop the iterations, and using the same accelerating scheme in the centralized algorithm. The common parameters of three algorithms are set as the same. The simulation results are shown in Figs. 4–6.

Remark 11. We choose the disturbance $d_i(k)$ to be uniformly distributed over $[-0.2, 0.2]$, as shown in Table 1. (1) The reason the negative disturbance is allowed is due to Assumption 2. According to the analysis in Remark 13, heat transfers between zones that are not immediate neighbors are omitted in order to design a distributed model predictive controller. This is equivalent to introduce negative heat gains to the thermal dynamics. (2) When $d_i(k) = 0.2$, the thermal disturbances will cause the indoor temperature to raise 0.2°F within 12 min. From the system model (A.3), this is equivalent to a power of thermal disturbances being 424.4 W for each zone, which is reasonable. (3) In this study, we use the robust optimization method to handle the unknown thermal disturbances. It is assumed that the minimum and maximum thermal disturbances are known. Thus, from the perspective of the algorithm in this paper, it does not matter what distribution the thermal disturbance has, as long as it is over a small finite interval. Thus, without loss of generality, we assume that it is uniformly distributed in the numerical simulations.

We take zone 1, opening from 9 am to 5 am, and zone 22, opening from 8 am to 10 pm, as examples for discussion. Fig. 4 shows the indoor temperature of both zones. The indoor temperature is around 72°F when the zones are open. When the zones are closed, for example between 5 pm and 9 am for zone 1, since the outdoor temperature is relatively high during this interval, the indoor temperature is typically higher than the reference temperature. In addition, the indoor temperatures for both zones are within the comfortable range defined by $T_i^{\min}(k)$

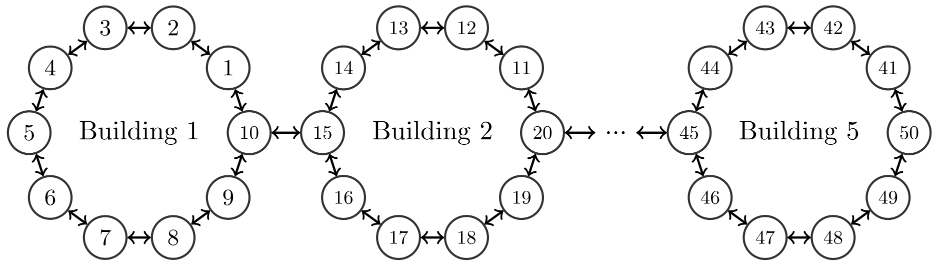

Fig. 1. The neighborhood relationship of all 50 zones.

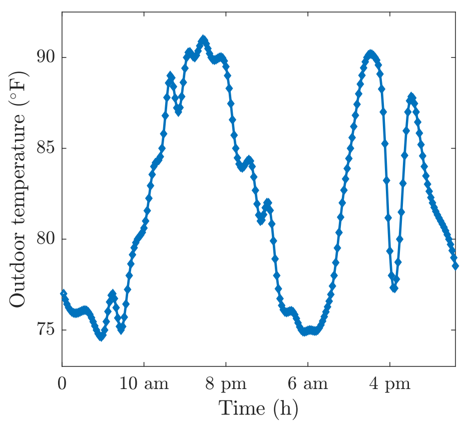

Fig. 2. The outdoor temperature in Atlanta from July 3 to July 4, 2020.

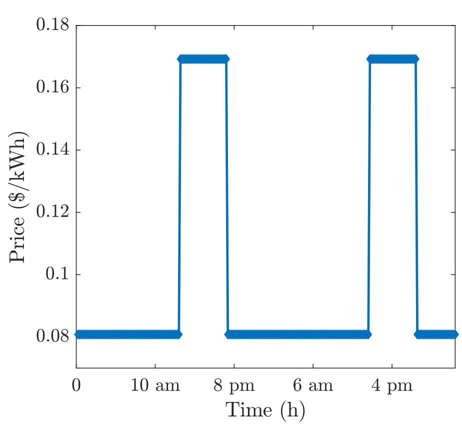

Fig. 3. The electricity price for business in Georgia (An explanation, 0000).

and $T_i^{\max}(k)$. Fig. 5 presents the HVAC power inputs in the two zones, which shows that all the input powers satisfy the local constraint. Fig. 6 shows the total HVAC power inputs of all 50 zones. The total power consumptions do not exceed the global bound for all three algorithms. Besides, when the buildings are closed, there is also nonzero power inputs, which are caused by the nonzero weights on the temperature deviation from the reference. This can avoid the indoor temperature being too high when the buildings are closed.

Remark 12. This remark is to explain why we assume $T_i^{\mathrm{ref}}(k) = (71+\mathcal{U}(-0.5, 0.5))$ °F while the indoor temperature is around 72 °F when the buildings are open. In Section 3.2, we replace the vectors $\boldsymbol{B}_i$ and $\boldsymbol{D}_i$ in the optimization problem with $\boldsymbol{B}_i^{\min}$ and $\boldsymbol{D}_i^{\max}$. Note that using $\boldsymbol{B}_i^{\min}$ can guarantee that the local power inputs never exceed the local bound in the presence of unknown bounded thermal disturbances, since $\boldsymbol{B}_i^{\min}$ serves as a tighter bound for the local power inputs than $\boldsymbol{B}_i$. However, the vector $\boldsymbol{D}_i$ appears originally in the objective function. When we use robust optimization to handle the uncertainties, the results guarantee the minimum of the defined objective no matter what the exact disturbances are. Thus, the algorithm actually considers the worst case. From the mathematical point of view, the reference temperature $\boldsymbol{T}_i^{\mathrm{ref}}(k_c)$ in $\boldsymbol{D}_i$ is replaced with $\boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c)+\boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c)+\boldsymbol{d}_i^{\min}(k_c)$ in $\boldsymbol{D}_i^{\max}$. Thus, the indoor temperature when the buildings are open is slightly higher than the predefined one. However, this will not affect the practical significance of the proposed algorithm, since we can always set the reference temperature to be slightly lower than the occupants' preferred temperatures, and tune it when necessary. In the future work, we plan to investigate how to determine the reference temperature by using the occupants' thermal feedback and machining learning method, which will be incorporated into the control framework presented herein.

5.2. Advantages of the proposed algorithm over the one in Xie et al. (2018) and the centralized algorithm

Figs. 4–6 show that the distributed algorithm in this paper, the distributed ADMM algorithm in Xie et al. (2018), and the centralized algorithm can all achieve indoor temperature regulation. However, the original algorithm in Xie et al. (2018) requires the building message controllers and the multi-building coordinator for the stopping criteria, making the HVAC systems not fully distributed. To evaluate the computing speed of these algorithms, a series of simulation studies are conducted for 5 buildings. The number of zones of each building increases from 5 to 15. Each simulation is over 120 time slots, i.e., 24 h when $\Delta = 0.2$h. The total running time of each simulation is tallied.

When using the distributed method to determine when to stop the iterations, the algorithm in Xie et al. (2018) requires significantly more time to compute the results, as is shown in Fig. 7. In contrast, the algorithm presented in this paper can accelerate the computing speed when the HVAC systems are fully distributed. The proposed distributed algorithm also presents higher computing speed compared with the centralized algorithm using the Nesterov's accelerating method when a building has a large number of zones. In addition, the total energy cost of all zones is shown in Fig. 8. The results demonstrate that the total energy cost is almost the same for the proposed algorithm and that in Xie et al. (2018), and is slightly lower for the centralized algorithm. However, considering that the running time of the centralized algorithm increases significantly with increasing of the number of zones, as shown in Fig. 7, it is beneficial to use the proposed distributed algorithm for buildings with a large number of zones. Since a commercial building typically has more than 15 zones, the proposed algorithm can achieve the distributed temperature regulation in real time, while the distributed ADMM in Xie et al. (2018) or the centralized algorithm might not, as shown in Fig. 7.

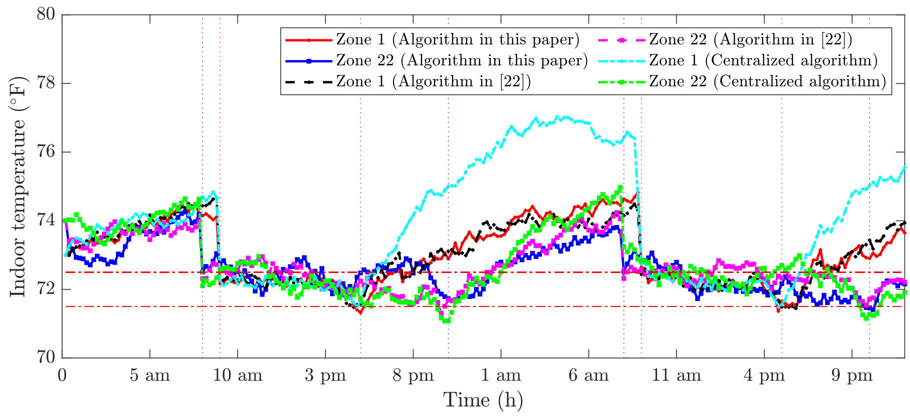

Fig. 4. The indoor temperature of zones 1 and 22 by the algorithms in this paper and Xie et al. (2018), and the centralized algorithm.

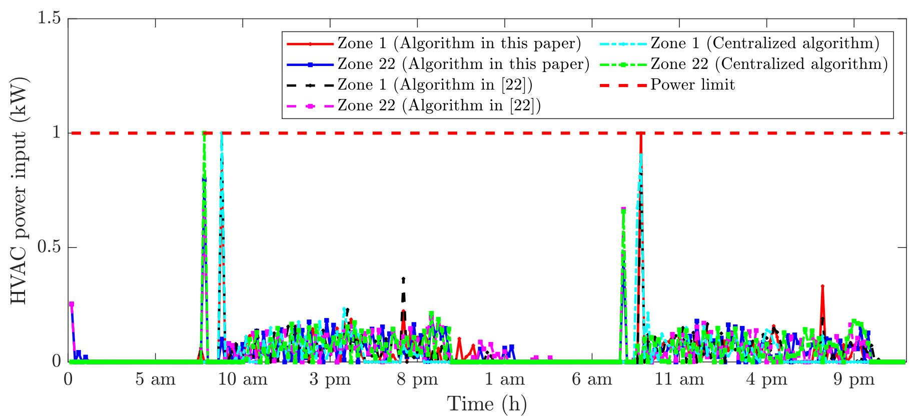

Fig. 5. The HVAC power input of zones 1 and 22 by the algorithms in this paper and Xie et al. (2018), and the centralized algorithm.

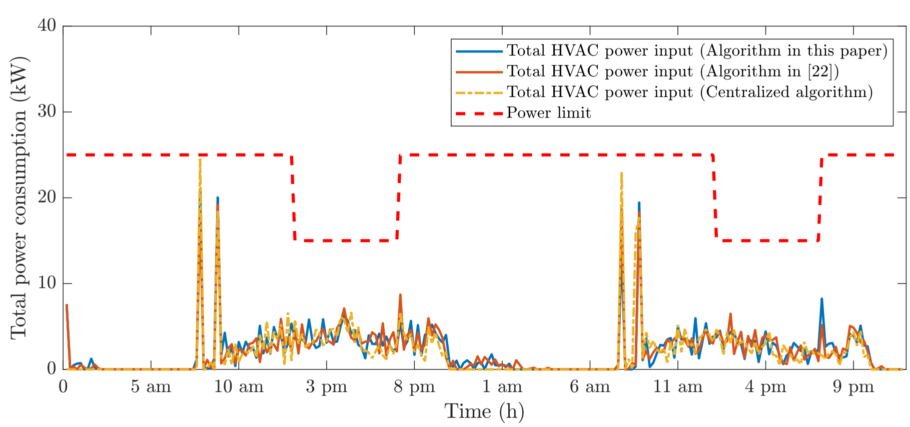

Fig. 6. The total power consumption by the algorithms in this paper and Xie et al. (2018), and the centralized algorithm.

6. Conclusions

In this paper, we investigated an accelerated distributed MPC strategy for HVAC systems with global constraints. Firstly, we incorporated the dynamics of indoor temperatures into MPC framework with the assumption that only thermal couplings between the immediate neighboring zones need to be considered. After presenting the system and cost models, we converted the constrained optimization problem into an quadratic programming problem and used robust optimization to handle the potential unknown bounded thermal disturbances. Then, based on the accelerated dual gradient-projection method, a distributed fast MPC protocol was designed for HVAC systems. In order to determine when to stop the iterations, a distributed stopping criterion was used based on a distributed average consensus algorithm. Besides,

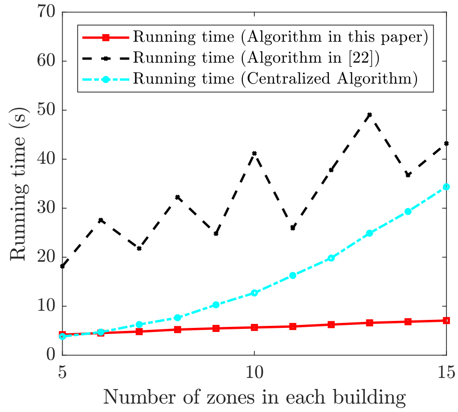

Fig. 7. The running time comparison between the algorithms in this paper and Xie et al. (2018), and the centralized algorithm.

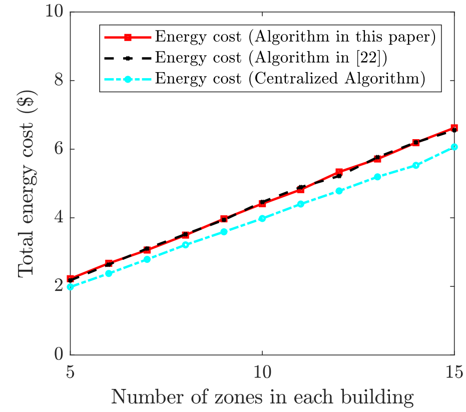

Fig. 8. The total energy cost comparison between the algorithms in this paper and Xie et al. (2018), and the centralized algorithm.

the tightening of coupled constraints was firstly implemented to compensate for the influence of early termination. Numerical simulations demonstrated the effectiveness of the proposed distributed MPC algorithm, and the comparisons with the MPC algorithm in Xie et al. (2018) and the centralized counterpart illustrated that the proposed algorithm can indeed accelerate the computation speed.

Our future research will focus on the algorithm for learning individuals' thermal preferences and how to incorporate the real-time thermal feedbacks into controller design.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Science Foundation, USA under Cyber–Physical Systems Grant No. 1837021.

Appendix A. Zone thermal dynamics model formulation

This appendix formulates the thermal dynamics of a zone including neighboring zones' thermal impact. The thermal dynamics of zones can be modeled similarly to a resistance–capacitance (RC) network.

For zone i, $i \in \mathbb{Z}_1^{MN}$, one has (Constantopoulos et al., 1991; Thatte & Xie, 2012; Wang, Hu, & Spanos, 2017; Yang et al., 2020)

$$C_i \frac{T_i(k+1)-T_i(k)}{\Delta} = \sum_{j\in\bar{\mathcal{N}}_i} \frac{T_j(k)-T_i(k)}{R_{ij}} + \frac{T_o(k)-T_i(k)}{R_{io}} - \eta_i P_i(k), \quad \text{(A.1)}$$

where R_{ij} denotes the thermal resistance between zones i and j in K/kW, and R_{io} represents the thermal resistance between zone i and the outside in K/kW, respectively. The meanings of other notations are the same as those given in Section 2.2.

By some mathematical operations, (A.1) can be reformulated as

$$T_i(k+1) = \bar{a}_{ii}T_i(k) + \sum_{j\in\bar{\mathcal{N}}_i} \bar{a}_{ij}T_j(k) + \bar{a}_{io}T_i^{\text{out}}(k) - \frac{\eta_i \Delta}{C_i}P_i(k), \quad \text{(A.2)}$$

where $\bar{a}_{ii} \triangleq 1 - \sum_{j\in\bar{\mathcal{N}}_i} \frac{\Delta}{R_{ij}C_i} - \frac{\Delta}{R_{io}C_i}$, $\bar{a}_{ij} \triangleq \frac{\Delta}{R_{ij}C_i}$, and $\bar{a}_{io} \triangleq \frac{\Delta}{R_{io}C_i}$.

Eq. (A.2) indicates that the indoor temperature $T_i(k+1)$ of zone i at time slot $k+1$ is related to the indoor temperatures of zone i and its neighbors, and the power input of zone i at time slot k. However, there might be some other energy sources that may influence the indoor temperature, such as some internal loads and solar gains. Thus, we include an additive thermal disturbance term $d_i(k)$ in the model (A.2), i.e.,

$$T_i(k+1) = \bar{a}_{ii}T_i(k) + \sum_{j\in\bar{\mathcal{N}}_i} \bar{a}_{ij}T_j(k) + \bar{a}_{io}T_i^{\text{out}}(k) - \frac{\eta_i \Delta}{C_i}P_i(k) + d_i(k). \quad \text{(A.3)}$$

Appendix B. Detailed formulation procedure of (13)

We suppose that there is no thermal couplings between two zones at different buildings. Thus, it is enough to consider all N zones in a building $m \in \mathbb{Z}_1^M$ when we try to get $\boldsymbol{T}_i(k_c)$ in the form of (13). In this case, we may write the system dynamics (1) in the following compact form.

$$\underbrace{\begin{bmatrix} T_1(k+1) \\ T_2(k+1) \\ \vdots \\ T_N(k+1) \end{bmatrix}}_{\bar{\boldsymbol{T}}_m(k+1)} = \underbrace{\begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} & \cdots & \bar{a}_{1N} \\ \bar{a}_{21} & \bar{a}_{22} & \cdots & \bar{a}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{a}_{N1} & \bar{a}_{N2} & \cdots & \bar{a}_{NN} \end{bmatrix}}_{\boldsymbol{A}_m} \underbrace{\begin{bmatrix} T_1(k) \\ T_2(k) \\ \vdots \\ T_N(k) \end{bmatrix}}_{\bar{\boldsymbol{T}}_m(k)} + \underbrace{\begin{bmatrix} \bar{a}_{1o} & & & \\ & \bar{a}_{2o} & & \\ & & \ddots & \\ & & & \bar{a}_{No} \end{bmatrix}}_{\boldsymbol{A}_{mo}} \underbrace{\begin{bmatrix} T_1^{\text{out}}(k) \\ T_2^{\text{out}}(k) \\ \vdots \\ T_N^{\text{out}}(k) \end{bmatrix}}_{\bar{\boldsymbol{T}}_m^{\text{out}}(k)} - \underbrace{\begin{bmatrix} \frac{\eta_1\Delta}{C_1} & & & \\ & \frac{\eta_2\Delta}{C_2} & & \\ & & \ddots & \\ & & & \frac{\eta_N\Delta}{C_N} \end{bmatrix}}_{\bar{\boldsymbol{\tau}}_m} \underbrace{\begin{bmatrix} P_1(k) \\ P_2(k) \\ \vdots \\ P_N(k) \end{bmatrix}}_{\bar{\boldsymbol{P}}_m(k)} + \underbrace{\begin{bmatrix} d_1(k) \\ d_2(k) \\ \vdots \\ d_N(k) \end{bmatrix}}_{\bar{\boldsymbol{d}}_m(k)}, \quad \text{(B.1)}$$

Thus,

$$\begin{aligned} \bar{\boldsymbol{T}}_m(k_c+k) &= \boldsymbol{A}_m\bar{\boldsymbol{T}}_m(k_c+k-1) + \boldsymbol{A}_{mo}\bar{\boldsymbol{T}}_m^{\text{out}}(k_c+k-1) - \bar{\boldsymbol{\tau}}_m\bar{\boldsymbol{P}}_m \\ &\quad (k_c+k-1) \\ &= \boldsymbol{A}_m^k\bar{\boldsymbol{T}}_m(k_c) + \sum_{k'=0}^{k-1} \boldsymbol{A}_m^{k'}\boldsymbol{A}_{mo}\bar{\boldsymbol{T}}_m^{\text{out}}(k_c+k-k'-1) \\ &\quad - \sum_{k'=0}^{k-1} \boldsymbol{A}_m^{k'}\bar{\boldsymbol{\tau}}_m\bar{\boldsymbol{P}}_m(k_c+k-k'-1) \\ &\quad + \sum_{k'=0}^{k-1} \boldsymbol{A}_m^{k'}\bar{\boldsymbol{d}}_m(k_c+k-k'-1). \end{aligned} \quad \text{(B.2)}$$

Similar to the Appendix of Xie et al. (2018), the indoor temperatures from time slot k_c+1 to k_c+h-1 for the ith zone can be described as

$$
\begin{aligned}
\underbrace{\begin{bmatrix} \bar{\boldsymbol{T}}_m(k_c+1) \\ \bar{\boldsymbol{T}}_m(k_c+2) \\ \vdots \\ \bar{\boldsymbol{T}}_m(k_c+h-1) \end{bmatrix}}_{\tilde{\boldsymbol{T}}_m(k_c)} = {}& \underbrace{\begin{bmatrix} \boldsymbol{A}_m & & & \\ & \boldsymbol{A}_m^2 & & \\ & & \ddots & \\ & & & \boldsymbol{A}_m^{h-1} \end{bmatrix}}_{\tilde{\boldsymbol{\eta}}_m^0} \underbrace{\begin{bmatrix} \bar{\boldsymbol{T}}_m(k_c) \\ \bar{\boldsymbol{T}}_m(k_c) \\ \vdots \\ \bar{\boldsymbol{T}}_m(k_c) \end{bmatrix}}_{\tilde{\boldsymbol{T}}_{m0}(k_c)} \\
&+ \underbrace{\begin{bmatrix} \boldsymbol{A}_{mo} & \boldsymbol{0}_{N\times N} & \cdots & \boldsymbol{0}_{N\times N} \\ \boldsymbol{A}_m\boldsymbol{A}_{mo} & \boldsymbol{A}_{mo} & \cdots & \boldsymbol{0}_{N\times N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_m^{h-2}\boldsymbol{A}_{mo} & \boldsymbol{A}_m^{h-3}\boldsymbol{A}_{mo} & \cdots & \boldsymbol{A}_{mo} \end{bmatrix}}_{\tilde{\boldsymbol{F}}_m^0} \\
&\times \underbrace{\begin{bmatrix} \bar{\boldsymbol{T}}_m^{\text{out}}(k_c) \\ \bar{\boldsymbol{T}}_m^{\text{out}}(k_c+1) \\ \vdots \\ \bar{\boldsymbol{T}}_m^{\text{out}}(k_c+h-2) \end{bmatrix}}_{\tilde{\boldsymbol{T}}_m^{\text{out}}(k_c)} \\
&- \underbrace{\begin{bmatrix} \bar{\boldsymbol{\tau}}_m & \boldsymbol{0}_{N\times N} & \cdots & \boldsymbol{0}_{N\times N} \\ \boldsymbol{A}_m\bar{\boldsymbol{\tau}}_m & \bar{\boldsymbol{\tau}}_m & \cdots & \boldsymbol{0}_{N\times N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_m^{h-2}\bar{\boldsymbol{\tau}}_m & \boldsymbol{A}_m^{h-3}\bar{\boldsymbol{\tau}}_m & \cdots & \bar{\boldsymbol{\tau}}_m \end{bmatrix}}_{\tilde{\boldsymbol{\tau}}_m^0} \\
&\times \underbrace{\begin{bmatrix} \bar{\boldsymbol{P}}_m(k_c) \\ \bar{\boldsymbol{P}}_m(k_c+1) \\ \vdots \\ \bar{\boldsymbol{P}}_m(k_c+h-2) \end{bmatrix}}_{\tilde{\boldsymbol{P}}_m(k_c)} \\
&+ \underbrace{\begin{bmatrix} \boldsymbol{I}_{N\times N} & \boldsymbol{0}_{N\times N} & \cdots & \boldsymbol{0}_{N\times N} \\ \boldsymbol{A}_m & \boldsymbol{I}_{N\times N} & \cdots & \boldsymbol{0}_{N\times N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_m^{h-2} & \boldsymbol{A}_m^{h-3} & \cdots & \boldsymbol{I}_{N\times N} \end{bmatrix}}_{\tilde{\boldsymbol{G}}_m^0} \underbrace{\begin{bmatrix} \bar{\boldsymbol{d}}_m(k_c) \\ \bar{\boldsymbol{d}}_m(k_c+1) \\ \vdots \\ \bar{\boldsymbol{d}}_m(k_c+h-2) \end{bmatrix}}_{\tilde{\boldsymbol{d}}_m(k_c)},
\end{aligned}
\tag{B.3}
$$

which has the compact form

$$
\tilde{\boldsymbol{T}}_m(k_c) = \tilde{\boldsymbol{\eta}}_m^0 \tilde{\boldsymbol{T}}_{m0}(k_c) + \tilde{\boldsymbol{F}}_m^0 \tilde{\boldsymbol{T}}_m^{\text{out}}(k_c) - \tilde{\boldsymbol{\tau}}_m^0 \tilde{\boldsymbol{P}}_m(k_c) + \tilde{\boldsymbol{G}}_m^0 \tilde{\boldsymbol{d}}_m(k_c). \tag{B.4}
$$

Note that $\tilde{\boldsymbol{\eta}}_m^0, \tilde{\boldsymbol{F}}_m^0, \tilde{\boldsymbol{\tau}}_m^0, \tilde{\boldsymbol{G}}_m^0 \in \mathbb{R}^{N(h-1)\times N(h-1)}$ are constant matrices for certain parameters of zone dynamics, the duration of each time slot Δ, and time horizon h. Thus, these three matrices can be precomputed and stored in computer before the iterations of MPC. To facilitate the analysis, we firstly provide the justification and meaning of Assumption 2.

Remark 13. The justification of Assumption 2 is stated as follows. The coefficient of thermal inertia is relatively large for a zone. For example, with $\Delta = 1$h, the coefficient of thermal inertia is 0.9608 in Constantopoulos et al. (1991). In Wang et al. (2017) and Yang et al. (2020), the coefficient of thermal inertia used in the simulation is 0.9147 with $\Delta = 0.2$h. We consider building 1 consisting of 10 zones in Fig. 1. The coefficients of thermal inertia is 0.9147, and the thermal coupling coefficient between the neighboring zones is 0.0374. That is,

$$
\boldsymbol{A}_1 \triangleq \begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} & \cdots & \bar{a}_{110} \\ \bar{a}_{21} & \bar{a}_{22} & \cdots & \bar{a}_{210} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{a}_{101} & \bar{a}_{102} & \cdots & \bar{a}_{1010} \end{bmatrix}
$$

$$
= \begin{bmatrix} 0.9147 & 0.0374 & & & 0.0374 \\ 0.0374 & 0.9147 & 0.0374 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.0374 & 0.9147 & 0.0374 \\ 0.0374 & & & 0.0374 & 0.9147 \end{bmatrix}. \tag{B.5}
$$

Then, one has

$$
\boldsymbol{A}_1^2 = \begin{bmatrix} 0.8395 & 0.0684 & 0.0014 & & & 0.0014 & 0.0684 \\ 0.0684 & 0.8395 & 0.0684 & 0.0014 & & & 0.0014 \\ 0.0014 & 0.0684 & 0.8395 & 0.0684 & 0.0014 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0.0014 & 0.0684 & 0.8395 & 0.0684 & 0.0014 \\ 0.0014 & & & 0.0014 & 0.0684 & 0.8395 & 0.0684 \\ 0.0684 & 0.0014 & & & 0.0014 & 0.0684 & 0.8395 \end{bmatrix}. \tag{B.6}
$$

Compared with 0.8395 and 0.0684, 0.0014 is very small. Assumption 2 indicates that we can neglect 0.0014 and only retain the elements 0.8395 and 0.0684. That is, we can approximate the matrix $\boldsymbol{A}_1^2$ by

$$
\boldsymbol{A}_1^2 \approx \begin{bmatrix} 0.8395 & 0.0684 & & & 0.0684 \\ 0.0684 & 0.8395 & 0.0684 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.0684 & 0.8395 & 0.0684 \\ 0.0684 & & & 0.0684 & 0.8395 \end{bmatrix}, \tag{B.7}
$$

and add the elements 0.0014 to the disturbance term.

By approximating the matrices $\boldsymbol{A}_m^i$, $\forall m \in \mathbb{Z}_1^M$, $i \in \mathbb{Z}_2^{h-1}$, in the similar way, Assumption 2 makes it possible to design a distributed MPC controller for the HVAC systems. Without this assumption, the information of immediate neighbors is not sufficient to compute the local control input. In particular, when the time horizon h is large, the global information of the distributed HVAC systems is required to design the control algorithm for each zone.

Under Assumption 2 and the approximating method illustrated in Remark 13, we denote the corresponding approximated matrices of $\tilde{\boldsymbol{\eta}}_m^0$, $\tilde{\boldsymbol{F}}_m^0$, $\tilde{\boldsymbol{\tau}}_m^0$, and $\tilde{\boldsymbol{G}}_m^0$ as $\tilde{\boldsymbol{\eta}}_m$, $\tilde{\boldsymbol{F}}_m$, $\tilde{\boldsymbol{\tau}}_m$, and $\tilde{\boldsymbol{G}}_m$. Note that the matrices $\tilde{\boldsymbol{\eta}}_m$, $\tilde{\boldsymbol{F}}_m$, $\tilde{\boldsymbol{\tau}}_m$, and $\tilde{\boldsymbol{G}}_m$ can be precomputed and stored in the computer, and do not need to be computed in the iterations of MPC. Thus, we can regard them as three constant matrices. From (B.4), one has

$$
\tilde{\boldsymbol{T}}_m(k_c) = \tilde{\boldsymbol{\eta}}_m \tilde{\boldsymbol{T}}_{m0}(k_c) + \tilde{\boldsymbol{F}}_m \tilde{\boldsymbol{T}}_m^{\text{out}}(k_c) - \tilde{\boldsymbol{\tau}}_m \tilde{\boldsymbol{P}}_m(k_c) + \tilde{\boldsymbol{G}}_m \tilde{\boldsymbol{d}}_m(k_c). \tag{B.8}
$$

From (B.3) and (B.8), we know that

$$
\begin{aligned}
T_i(k_c+k+1) = {}& \sum_{j\in\bar{\mathcal{N}}_i\cup i} \eta_{(kN+i)(kN+j)} T_j(k_c) \\
&+ \sum_{j\in\bar{\mathcal{N}}_i\cup i} \sum_{k'=0}^{k} F_{(kN+i)(k'N+j)} T_j^{\text{out}}(k_c+k') \\
&- \sum_{j\in\bar{\mathcal{N}}_i\cup i} \sum_{k'=0}^{k} \tau_{(kN+i)(k'N+j)} P_j(k_c+k') \\
&+ \sum_{j\in\bar{\mathcal{N}}_i\cup i} \sum_{k'=0}^{k} G_{(kN+i)(k'N+j)} d_j(k_c+k'),
\end{aligned}
\tag{B.9}
$$

for $k \in \mathbb{Z}_0^{h-2}$, and $i \in \mathbb{Z}_1^M$. It should be noted that η_{ij}, F_{ij}, τ_{ij}, and G_{ij} denote the (i,j)th elements of the matrices $\tilde{\boldsymbol{\eta}}_{ij}$, $\tilde{\boldsymbol{F}}_{ij}$, $\tilde{\boldsymbol{\tau}}_{ij}$, and $\tilde{\boldsymbol{G}}_{ij}$, respectively.

Thus,

$$
\underbrace{\begin{bmatrix} T_i(k_c+1) \\ T_i(k_c+2) \\ \vdots \\ T_i(k_c+h-1) \end{bmatrix}}_{\boldsymbol{T}_i(k_c)} = \underbrace{\begin{bmatrix} \sum_{j\in\bar{\mathcal{N}}_i\cup i} \eta_{ij} T_j(k_c) \\ \sum_{j\in\bar{\mathcal{N}}_i\cup i} \eta_{(N+i)(N+j)} T_j(k_c) \\ \vdots \\ \sum_{j\in\bar{\mathcal{N}}_i\cup i} \eta_{((h-2)N+i)((h-2)N+j)} T_j(k_c) \end{bmatrix}}_{\boldsymbol{T}_{i0}(k_c)}
$$

$$+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\underbrace{\begin{bmatrix} F_{ij} & 0 & \cdots & 0\\ F_{(N+i)j} & F_{(N+i)(N+j)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ F_{((h-2)N+i)j} & F_{((h-2)N+i)(N+j)} & \cdots & F_{((h-2)N+i)((h-2)N+j)}\end{bmatrix}}_{\boldsymbol{F}_{ij}}$$
$$\times\underbrace{\begin{bmatrix} T_j^{\mathrm{out}}(k_c)\\ T_j^{\mathrm{out}}(k_c+1)\\ \vdots\\ T_j^{\mathrm{out}}(k_c+h-2)\end{bmatrix}}_{\boldsymbol{T}_j^{\mathrm{out}}(k_c)}$$
$$-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\underbrace{\begin{bmatrix} \tau_{ij} & 0 & \cdots & 0\\ \tau_{(N+i)j} & \tau_{(N+i)(N+j)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \tau_{((h-2)N+i)j} & \tau_{((h-2)N+i)(N+j)} & \cdots & \tau_{((h-2)N+i)((h-2)N+j)}\end{bmatrix}}_{\boldsymbol{\tau}_{ij}}$$
$$\times\underbrace{\begin{bmatrix} P_j(k_c)\\ P_j(k_c+1)\\ \vdots\\ P_j(k_c+h-2)\end{bmatrix}}_{\boldsymbol{P}_j(k_c)}$$
$$+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\underbrace{\begin{bmatrix} G_{ij} & 0 & \cdots & 0\\ G_{(N+i)j} & G_{(N+i)(N+j)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ G_{((h-2)N+i)j} & G_{((h-2)N+i)(N+j)} & \cdots & G_{((h-2)N+i)((h-2)N+j)}\end{bmatrix}}_{\boldsymbol{G}_{ij}}$$
$$\times\underbrace{\begin{bmatrix} d_j(k_c)\\ d_j(k_c+1)\\ \vdots\\ d_j(k_c+h-2)\end{bmatrix}}_{\boldsymbol{d}_j(k_c)} \tag{B.10}$$

can be written in the compact form

$$\boldsymbol{T}_i(k_c)=\boldsymbol{T}_{i0}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c). \tag{B.11}$$

Appendix C. Transform of the objective of (21)

We firstly consider only the ith zone, with $i\in\mathbb{Z}_1^{MN}$. From (6), the cost function $\sum_{k=k_c}^{k_c+h-2}\ell_i(k)$ for zone i can be simplified as follows.

$$\begin{aligned}\sum_{k=k_c}^{k_c+h-2}\ell_i(k) &= \sum_{k=k_c}^{k_c+h-2}\left(S(k)P_i(k)+\sigma_i(k+1)\left(T_i(k+1)-T_i^{\mathrm{ref}}(k+1)\right)^2\right)\\ &=\sum_{k=k_c}^{k_c+h-2}S(k)P_i(k)+\sum_{k=k_c}^{k_c+h-2}\left(\sigma_i(k+1)\left(T_i^2(k+1)\right.\right.\\ &\quad\left.\left.+T_i^{\mathrm{ref}2}(k+1)-2T_i(k+1)T_i^{\mathrm{ref}}(k+1)\right)\right)\\ &=\boldsymbol{S}^{\mathrm{T}}(k_c)\boldsymbol{P}_i(k_c)+\boldsymbol{T}_i^{\mathrm{T}}(k_c)\boldsymbol{\sigma}_i(k_c)\boldsymbol{T}_i(k_c)-2\boldsymbol{T}_i^{\mathrm{refT}}(k_c)\boldsymbol{\sigma}_i(k_c)\\ &\quad\times\boldsymbol{T}_i(k_c)+\boldsymbol{T}_i^{\mathrm{refT}}(k_c)\boldsymbol{\sigma}_i(k_c)\boldsymbol{T}_i^{\mathrm{ref}}(k_c).\end{aligned} \tag{C.1}$$

Substituting (13) into (C.1), one has

$$\begin{aligned}&\sum_{k=k_c}^{k_c+h-2}\ell_i(k)\\ &=\boldsymbol{S}^{\mathrm{T}}(k_c)\boldsymbol{P}_i(k_c)+\Bigg(\boldsymbol{T}_{i0}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)\\ &\quad+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)\\ &\quad-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)-2\boldsymbol{T}_i^{\mathrm{refT}}(k_c)\boldsymbol{\sigma}_i(k_c)\\ &\quad\times\Bigg(\boldsymbol{T}_{i0}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)\\ &\quad+\boldsymbol{T}_i^{\mathrm{refT}}(k_c)\boldsymbol{\sigma}_i(k_c)\boldsymbol{T}_i^{\mathrm{ref}}(k_c)\\ &=\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\boldsymbol{\tau}_{ii}\boldsymbol{P}_i(k_c)\\ &\quad+\Bigg(\boldsymbol{S}^{\mathrm{T}}(k_c)-2\Bigg(\boldsymbol{T}_{i0}(k_c)-\boldsymbol{T}_i^{\mathrm{ref}}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)\\ &\quad-\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\boldsymbol{\tau}_{ii}\Bigg)\boldsymbol{P}_i(k_c)\\ &\quad+\Bigg(\boldsymbol{T}_{i0}(k_c)-\boldsymbol{T}_i^{\mathrm{ref}}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)\\ &\quad+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c)-\boldsymbol{T}_i^{\mathrm{ref}}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)\\ &\quad-\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)\\ &=\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c)+\boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c)+E_i.\end{aligned} \tag{C.2}$$

Appendix D. Converting uncertain optimization problem (21) to the certain counterpart (26)

This appendix presents how uncertain optimization problem (21) can be transformed to its certain counterpart (26).

Note that both the objective function $\sum_{i=1}^{MN}\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c)+\boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c)$ in (21) and the constraint (17) are uncertain in this case, since the vectors $\boldsymbol{B}_i$ and $\boldsymbol{D}_i$ are uncertain. In order to make the problem easy to analyze, we firstly perform the following standardization procedure.

It is easy to check that the constrained optimization problem (21) is equivalent to the following problem with decision variables being $\boldsymbol{P}_i(k_c)$ and t_i, $\forall i\in\mathbb{Z}_1^{MN}$.

$$\min_{\boldsymbol{P}_i(k_c),t_i}\sum_{i=1}^{MN}t_i, \tag{D.1a}$$
$$\text{s.t. }\boldsymbol{A}_i\boldsymbol{P}_i(k_c)-\boldsymbol{B}_i\le\boldsymbol{0}_{4(h-1)}, \tag{D.1b}$$
$$\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c)+\boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c)-t_i\le 0,\quad\forall i\in\mathbb{Z}_1^{MN}, \tag{D.1c}$$
$$\sum_{i=1}^{MN}\boldsymbol{P}_i(k_c)\le\boldsymbol{b}(k_c,\epsilon),\quad\forall k_c\in\mathbb{Z}_1^{H-h+1}. \tag{D.1d}$$

We now consider the uncertainties of vectors $\boldsymbol{B}_i$ and $\boldsymbol{D}_i$ in (D.1b) and (D.1c).

For vector $\boldsymbol{B}_i$, by defining

$$\boldsymbol{B}_i^{\min}\triangleq\begin{bmatrix}\boldsymbol{P}_i^{\max}(k_c)\\ -\boldsymbol{P}_i^{\min}(k_c)\\ \boldsymbol{T}_{i0}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\boldsymbol{d}_i^{\min}(k_c)-\boldsymbol{T}_i^{\min}(k_c)\\ -\boldsymbol{T}_{i0}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)-\boldsymbol{d}_i^{\max}(k_c)+\boldsymbol{T}_i^{\max}(k_c)\end{bmatrix}, \tag{D.2a}$$

$$\boldsymbol{B}_i^{\max}\triangleq\begin{bmatrix}\boldsymbol{P}_i^{\max}(k_c)\\ -\boldsymbol{P}_i^{\min}(k_c)\\ \boldsymbol{T}_{i0}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)+\boldsymbol{d}_i^{\max}(k_c)-\boldsymbol{T}_i^{\min}(k_c)\\ -\boldsymbol{T}_{i0}(k_c)-\sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)+\sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)-\boldsymbol{d}_i^{\min}(k_c)+\boldsymbol{T}_i^{\max}(k_c)\end{bmatrix}, \tag{D.2b}$$

one has

$$\boldsymbol{B}_i^{\min} \le \boldsymbol{B}_i \le \boldsymbol{B}_i^{\max}. \tag{D.3}$$

We consider $\boldsymbol{D}_i = \boldsymbol{S}(k_c) - 2\left(\left(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{T}_i^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i} \boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\right)^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\boldsymbol{\tau}_{ii}\right)^{\mathrm{T}}$. Since $\delta_i^{\mathrm{ref}}(k_c) + \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) \le \boldsymbol{T}_i^{\mathrm{ref}}(k_c) \le \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c) + \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c)$ from (22), we have

$$\begin{aligned}
&\boldsymbol{T}_{i0}(k_c) - \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)\\
&\quad + \boldsymbol{d}_i^{\min}(k_c)\\
&\le \boldsymbol{T}_{i0}(k_c) - \boldsymbol{T}_i^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)\\
&\quad + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\\
&\le \boldsymbol{T}_{i0}(k_c) - \delta_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)\\
&\quad + \boldsymbol{d}_i^{\max}(k_c).
\end{aligned} \tag{D.4}$$

According to the fact that $\boldsymbol{\sigma}_i(k_c)$ is a diagonal matrix with positive diagonal entries, we have

$$\begin{aligned}
&\boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)\\
&\quad - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\min}(k_c)\Bigg)\\
&\le \boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{T}_i^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c)\\
&\quad + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{G}_{ij}\boldsymbol{d}_j(k_c)\Bigg)\\
&\le \boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c) - \delta_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)\\
&\quad - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\max}(k_c)\Bigg).
\end{aligned} \tag{D.5}$$

From the construction of matrix $\boldsymbol{\tau}_{ii}$ in (B.10), we can find that all the elements of $\boldsymbol{\tau}_{ii}$ are non-negative, which indicates that $\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{u} \le \boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{v}$ for any vectors $\boldsymbol{u} \le \boldsymbol{v}$. Thus, one has

$$\begin{aligned}
&\boldsymbol{S}(k_c) - 2\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\Bigg(\boldsymbol{T}_{i0}(k_c) - \delta_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c)\\
&- \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\max}(k_c)\Bigg) \le \boldsymbol{D}_i \le \boldsymbol{S}(k_c) - 2\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\Big(\boldsymbol{T}_{i0}(k_c)\\
&- \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\min}(k_c)\Big),
\end{aligned} \tag{D.6}$$

according to the definition of $\boldsymbol{D}_i$ in (18e).

If we denote $\boldsymbol{D}_i^{\min} \triangleq \boldsymbol{S}(k_c) - 2\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\left(\boldsymbol{T}_{i0}(k_c) - \delta_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\max}(k_c)\right)$ and $\boldsymbol{D}_i^{\max} \triangleq \boldsymbol{S}(k_c) - 2\boldsymbol{\tau}_{ii}^{\mathrm{T}}\boldsymbol{\sigma}_i(k_c)\left(\boldsymbol{T}_{i0}(k_c) - \boldsymbol{\Delta}_i^{\mathrm{ref}}(k_c) - \boldsymbol{T}_{in0}^{\mathrm{ref}}(k_c) + \sum_{j\in\bar{\mathcal{N}}_i\cup i}\boldsymbol{F}_{ij}\boldsymbol{T}_j^{\mathrm{out}}(k_c) - \sum_{j\in\bar{\mathcal{N}}_i}\boldsymbol{\tau}_{ij}\boldsymbol{P}_j(k_c) + \boldsymbol{d}_i^{\min}(k_c)\right)$, then (D.6) can be written as

$$\boldsymbol{D}_i^{\min} \le \boldsymbol{D}_i \le \boldsymbol{D}_i^{\max}. \tag{D.7}$$

Now we consider the robust counterpart of the uncertain programming problem without the global constraint as follows.

$$\min_{\boldsymbol{P}_i(k_c),t_i} \; t_i, \tag{D.8a}$$

$$\text{s.t. } \boldsymbol{A}_i\boldsymbol{P}_i(k_c) - \boldsymbol{B}_i \le \boldsymbol{0}_{4(h-1)}, \tag{D.8b}$$

$$\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c) - t_i \le 0, \quad \forall k_c \in \mathbb{Z}_1^{H-h+1}, \tag{D.8c}$$

$$\boldsymbol{B}_i^{\min} \le \boldsymbol{B}_i \le \boldsymbol{B}_i^{\max}, \tag{D.8d}$$

$$\boldsymbol{D}_i^{\min} \le \boldsymbol{D}_i \le \boldsymbol{D}_i^{\max}. \tag{D.8e}$$

Firstly, it is easy to see that $\boldsymbol{P}_i(k_c)$ satisfies $\boldsymbol{A}_i\boldsymbol{P}_i(k_c) - \boldsymbol{B}_i \le \boldsymbol{0}_{4(h-1)}$ for all $\boldsymbol{B}_i^{\min} \le \boldsymbol{B}_i \le \boldsymbol{B}_i^{\max}$ if and only if $\boldsymbol{P}_i(k_c)$ satisfies $\boldsymbol{A}_i\boldsymbol{P}_i(k_c) - \boldsymbol{B}_i^{\min} \le \boldsymbol{0}_{4(h-1)}$. Similarly, $\boldsymbol{P}_i(k_c)$ and t_i satisfies $\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c) - t_i \le 0$ for all $\boldsymbol{D}_i^{\min} \le \boldsymbol{D}_i \le \boldsymbol{D}_i^{\max}$ if and only if $\boldsymbol{P}_i(k_c)$ and t_i satisfies $\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\mathrm{T}}\boldsymbol{P}_i(k_c) - t_i \le 0$, for all $\boldsymbol{D}_i^{\min} \le \boldsymbol{D}_i \le \boldsymbol{D}_i^{\max}$, since we only consider the cooling mode and thus $\boldsymbol{P}_i(k_c) \ge \boldsymbol{0}_{h-1}$.

Denote $\mathcal{P}_i \triangleq \{\boldsymbol{P}_i(k_c) \mid \boldsymbol{A}_i\boldsymbol{P}_i(k_c) \le \boldsymbol{B}_i^{\min}, \forall k_c \in \mathbb{Z}_1^{H-h+1}\}$. The constraints (15) and (16), namely (3) and (4), can be written as

$$\boldsymbol{P}_i(k_c) \in \mathcal{P}_i. \tag{D.9}$$

Thus, we have

$$\min_{\boldsymbol{P}_i(k_c),t_i} \; t_i, \tag{D.10a}$$

$$\text{s.t. } \boldsymbol{A}_i\boldsymbol{P}_i(k_c) - \boldsymbol{B}_i^{\min} \le \boldsymbol{0}_{4(h-1)}, \tag{D.10b}$$

$$\boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c) - t_i \le 0, \quad \forall k_c \in \mathbb{Z}_1^{H-h+1}. \tag{D.10c}$$

which is equivalent to

$$\min_{\boldsymbol{P}_i(k_c)\in\mathcal{P}_i} \sum_{i=1}^{MN} \boldsymbol{P}_i^{\mathrm{T}}(k_c)\boldsymbol{C}_i\boldsymbol{P}_i(k_c) + \boldsymbol{D}_i^{\max\mathrm{T}}\boldsymbol{P}_i(k_c), \quad \text{s.t. (10)}. \tag{D.11}$$

References

Afram, A., & Janabi-Sharifi, F. (2014). Theory and applications of hvac control systems–a review of model predictive control (mpc). *Building and Environment*, *72*, 343–355.

(0000). An explanation of Georgia power's tariff for business, https://www.georgiapower.com/content/dam/georgia-power/pdfs/business-pdfs/rates-schedules/small-business/MKT_Time_of_use_Rates_brochure.pdf.

Atlanta, (0000). GA Weather History, https://www.wunderground.com/history/daily/us/ga/atlanta/KATL/date.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization, Vol. 28*. Princeton University Press.

Bertsekas, D. P. (2009). *Convex optimization theory*. Athena Scientific Belmont.

Bertsekas, D. P. (2015). *Convex optimization algorithms*. Athena Scientific Belmont.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Camacho, E. F., & Alba, C. B. (2013). *Model predictive control*. Springer Science & Business Media.

Camponogara, E., Jia, D., Krogh, B. H., & Talukdar, S. (2002). Distributed model predictive control. *IEEE Control Systems Magazine*, *22*(1), 44–52.

Cao, X., Chen, J., Xiao, Y., & Sun, Y. (2009). Building-environment control with wireless sensor and actuator networks: Centralized versus distributed. *IEEE Transactions on Industrial Electronics*, *57*(11), 3596–3605.

Constantopoulos, P., Schweppe, F. C., & Larson, R. C. (1991). Estia: A real-time consumer control scheme for space conditioning usage under spot electricity pricing. *Computers & Operations Research*, *18*(8), 751–765.

East, S., & Cannon, M. (2019). Energy management in plug-in hybrid electric vehicles: Convex optimization algorithms for model predictive control. *IEEE Transactions on Control Systems Technology*, *28*(6), 2191–2203.

Fazenda, P., Veeramachaneni, K., Lima, P., & O'Reilly, U.-M. (2014). Using reinforcement learning to optimize occupant comfort and energy usage in hvac systems. *Journal of Ambient Intelligence and Smart Environments*, *6*(6), 675–690.

Goetzler, W., Shandross, R., Young, J., Petritchenko, O., Ringo, D., & McClive, S. (2017). *Energy savings potential and rd & d opportunities for commercial building hvac systems*: *Tech. rep.*, Burlington, MA (United States): Navigant Consulting, URL https://www.energy.gov/sites/prod/files/2017/12/f46/bto-DOE-Comm-HVAC-Report-12-21-17.pdf.

Humphreys, M. A., & Nicol, J. F. (2002). The validity of iso-pmv for predicting comfort votes in every-day thermal environments. *Energy and Buildings*, *34*(6), 667–684.

Hussain, S., Gabbar, H. A., Bondarenko, D., Musharavati, F., & Pokharel, S. (2014). Comfort-based fuzzy control optimization for energy conservation in hvac systems. *Control Engineering Practice*, *32*, 172–182.

Kim, J., Zhou, Y., Schiavon, S., Raftery, P., & Brager, G. (2018). Personal comfort models: predicting individuals' thermal preference using occupant heating and cooling behavior and machine learning. *Building and Environment*, *129*, 96–106.

Klein, L., Kwak, J.-y., Kavulya, G., Jazizadeh, F., Becerik-Gerber, B., Varakantham, P., et al. (2012). Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Automation in Construction*, *22*, 525–536.

Koebrich, S., Bowen, T., & Sharpe, A. (2020). *2018 renewable energy data book*: *Tech. rep.*, Golden, CO (United States): National Renewable Energy Lab. (NREL), URL https://www.nrel.gov/docs/fy20osti/75284.pdf.

Lampropoulos, I., Baghină, N., Kling, W. L., & Ribeiro, P. F. (2013). A predictive control scheme for real-time demand response applications. *IEEE Transactions on Smart Grid*, *4*(4), 2049–2060.

Lee, D., Ooka, R., Ikeda, S., Choi, W., & Kwak, Y. (2020). Model predictive control of building energy systems with thermal energy storage in response to occupancy variations and time-variant electricity prices. *Energy and Buildings*, Article 110291.

Li, J., Wu, Z., Wu, C., Long, Q., & Wang, X. (2016). An inexact dual fast gradient-projection method for separable convex optimization with linear coupled constraints. *Journal of Optimization Theory and Applications*, *168*(1), 153–171.

Mei, J., & Xia, X. (2019). Distributed control for a multi-evaporator air conditioning system. *Control Engineering Practice*, *90*, 85–100.

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

Müller, M. A., & Allgöwer, F. (2017). Economic and distributed model predictive control: Recent developments in optimization-based control. *SICE Journal of Control, Measurement, and System Integration*, *10*(2), 39–52.

Nedelcu, V., Necoara, I., & Tran-Dinh, Q. (2014). Computational complexity of inexact gradient augmented lagrangian methods: application to constrained mpc. *SIAM Journal on Control and Optimization*, *52*(5), 3109–3134.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, *103*(1), 127–152.

Nguyen, D. T., & Le, L. B. (2013). Joint optimization of electric vehicle and home energy scheduling considering user comfort preference. *IEEE Transactions on Smart Grid*, *5*(1), 188–199.

O'Dwyer, E., De Tommasi, L., Kouramas, K., Cychowski, M., & Lightbody, G. (2017). Prioritised objectives for model predictive control of building heating systems. *Control Engineering Practice*, *63*, 57–68.

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215–233.

Parisio, A., Rikos, E., & Glielmo, L. (2014). A model predictive control approach to microgrid operation optimization. *IEEE Transactions on Control Systems Technology*, *22*(5), 1813–1827.

Patrinos, P., & Bemporad, A. (2014). An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, *59*(1), 18–33.

Radhakrishnan, N., Srinivasan, S., Su, R., & Poolla, K. (2017). Learning-based hierarchical distributed hvac scheduling with operational constraints. *IEEE Transactions on Control Systems Technology*, *26*(5), 1892–1900.

Ren, W., & Beard, R. W. (2008). *Distributed consensus in multi-vehicle cooperative control*. Springer.

Rubagotti, M., Patrinos, P., & Bemporad, A. (2014). Stabilizing linear model predictive control under inexact numerical optimization. *IEEE Transactions on Automatic Control*, *59*(6), 1660–1666.

Saber, R. O., & Murray, R. M. (2003). Consensus protocols for networks of dynamic agents. In *Proceedings of the 2003 American control conference, Vol. 2* (pp. 951–956).

Serale, G., Fiorentini, M., Capozzoli, A., Bernardini, D., & Bemporad, A. (2018). Model predictive control (mpc) for enhancing building and hvac system energy efficiency: Problem formulation, applications and opportunities. *Energies*, *11*(3), 631.

Thatte, A. A., & Xie, L. (2012). Towards a unified operational value index of energy storage in smart grid environment. *IEEE Transactions on Smart Grid*, *3*(3), 1418–1426.

Valenzuela, P. E., Ebadat, A., Everitt, N., & Parisio, A. (2019). Closed-loop identification for model predictive control of hvac systems: From input design to controller synthesis. *IEEE Transactions on Control Systems Technology*, 1–15.

Wang, L. (2009). *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media.

Wang, Y., & Boyd, S. (2009). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, *18*(2), 267–278.

Wang, Z., Hu, G., & Spanos, C. J. (2017). Distributed model predictive control of bilinear hvac systems using a convexification method. In *2017 11th Asian control conference (ASCC)* (pp. 1608–1613). IEEE.

Wang, Z., & Ong, C. J. (2017). Distributed model predictive control of linear discrete-time systems with local and global constraints. *Automatica*, *81*, 184–195.

Wang, Z., & Ong, C.-J. (2018). Accelerated distributed mpc of linear discrete-time systems with coupled constraints. *IEEE Transactions on Automatic Control*, *63*(11), 3838–3849.

Wei, T., Wang, Y., & Zhu, Q. (2017). Deep reinforcement learning for building hvac control. In *Proceedings of the 54th annual design automation conference 2017* (pp. 1–6).

West, S. R., Ward, J. K., & Wall, J. (2014). Trial results from a model predictive control and optimisation system for commercial building hvac. *Energy and Buildings*, *72*, 271–279.

Xie, D., Yu, L., Jiang, T., & Zou, Y. (2018). Distributed energy optimization for hvac systems in university campus buildings. *IEEE Access*, *6*, 59141–59151.

Yadav, V., & Salapaka, M. V. (2007). Distributed protocol for determining when averaging consensus is reached. In *45th annual allerton conference on communication, control, and computing* (pp. 715–720).

Yang, Y., Hu, G., & Spanos, C. J. (2020). Hvac energy cost optimization for a multizone building via a decentralized approach. *IEEE Transactions on Automation Science and Engineering*, *17*(4), 1950–1960.

Yu, L., Xie, D., Jiang, T., Zou, Y., & Wang, K. (2017). Distributed real-time hvac control for cost-efficient commercial buildings under smart grid environment. *IEEE Internet of Things Journal*, *5*(1), 44–55.

Zhang, Z., Deng, R., Yuan, T., & Qin, S. J. (2017a). Distributed optimization of multi-building energy systems with spatially and temporally coupled constraints. In *2017 American control conference (ACC)* (pp. 2913–2918). IEEE.

Zhang, Z., Deng, R., Yuan, T., & Qin, S. J. (2017b). Sliding window games for cooperative building temperature control using a distributed learning method. *Frontiers of Engineering Management*, *4*(3), 304–314.