List of Contributors

Yudong Ma
Mechanical Engineering University of California at Berkeley, CA 94720-1740, USA.

Stefan Richter
Automatic Control Laboratory ETH Zürich,Physikstrasse 3, ETL I11, 8092 Zürich, Switzerland

Francesco Borrelli
Mechanical Engineering University of California at Berkeley, CA 94720-1740, USA.

Chapter 1

Distributed Model Predictive Control for Building Temperature Regulation

Abstract

We study the problem of temperature regulation in a network of building thermal zones. The control objective is to keep zone temperatures within a comfort range while consuming the least energy using predictive knowledge of weather and occupancy. First, we present a simplified thermal zone dynamic model and energy consumption models for system components. Model identification and validation based on historical measured data are presented. Second, a distributed model-based predictive control (DMPC) is designed for optimal heating and cooling. The DMPC is implemented using sequential quadratic program, dual decomposition, and fast gradient method. Simulation results show good performance and computational tractability of the resulting scheme.

1.1 INTRODUCTION

The building sector consumes about 40% of the energy used in the United States and is responsible for nearly 40% of greenhouse gas emissions. It is therefore economically, socially, and environmentally significant to reduce the energy consumption of buildings.

This work focuses on the modeling and predictive control of networks of thermal zones. The system considered in this manuscript consists of an air handling unit (AHU) and a set of variable air volume (VAV) boxes which serves a network of thermal zones. The AHU is equipped with a cooling coil, a damper, and a fan. The damper mixes return air and outside air. The cooling coil cools down the mixed air, and the fan drives the air to the VAV boxes. Each VAV box has a damper controlling the mass flow rate of air supplied to thermal zones. A heating coil in each VAV box can reheat the supply air when necessary.

The paper is divided in two parts. The objective of the first part is to develop low-order models suitable for real-time predictive optimization. This part is extracted from [1]. In this work we model the system as a network of two-mass

nonlinear systems. We present identification and validation results based on historical data collected from Bancroft Library at the University of California, Berkeley. The results are promising and show that the models well capture thermal zone dynamics when the external load (due to occupancy, weather, and equipment) is minor. Historical data are then used to compute the envelope of the external load by comparing nominal models and measured data when external disturbances are not negligible.

In the second part, a distributed model-based predictive control (DMPC) is designed for regulating heating and cooling in order to minimize energy consumption while satisfying comfort constraints. The main idea of predictive control is to use the model of the plant to predict the future evolution of the system [2, 3, 4]. At each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon. The optimal command signal is applied to the process only during the following sampling interval. At the next time step a new optimal control problem based on new measurements of the state is solved over a shifted horizon. For complex constrained multivariable control problems, model predictive control has become the accepted standard in the process industries [5]: its success is largely due to its almost unique ability to simply and effectively handle hard constraints on control inputs and states.

The size of the centralized predictive control problem rapidly grows when a realistic number of rooms together with a meaningful control horizon are considered. Therefore the real-time implementation of an MPC scheme is a challenge for the low-cost embedded platforms currently used for HVAC control algorithms. The techniques presented in this paper enable the implementation of an MPC algorithm by distributing the computational load on a set of VAV box embedded controllers coordinated by the embedded controllers on the AHU system. Compared to existing DMPC schemes [6, 7, 8, 9], the proposed method is tailored to the specific class of problems considered in this work. In particular, it makes use of sequential quadratic programming (SQP) [10, 11], proximal minimization [12], and dual decomposition [13] to handle the system nonlinearities and the decentralization, respectively. The SQP and proximal minimization methods are used to derive a strictly convex Quadratic Program (QP) from the original nonlinear optimization problem. The dual decomposition scheme takes advantage of the separability of the dual Lagrangian QP problem. By doing so, the dual QP is solved iteratively by updating dual and primal variables in a distributed fashion. In this paper we show that if the centralized MPC problem is properly formulated, the resulting primal and dual update laws can be easily derived. Simulation results show good performance and computational tractability of the resulting scheme.

We remark that the evaluation of optimal controllers for building climate regulation has been studied in the past by several authors (see [14, 15, 16, 17] and references therein). Compared to existing literature, this paper focuses on distributing the computational load of the predictive controller on multiple, low cost, embedded platforms.

The paper is organized as follows. Section 1.2 introduces the system and the simplified thermal zone models to be used for the predictions in MPC schemes. In Section 1.3 the distributed MPC control algorithm is outlined. A numerical example

is presented in Section 1.4. Finally, conclusions are drawn in Section 1.5.

1.2 SYSTEM MODEL

The objective of this section is to introduce a simplified HVAC system architecture and develop a control oriented model for it. We consider an AHU and a fan serving multiple VAV boxes controlling air temperature and flows in a network of thermal zones (next called "rooms" for brevity). Figure 1.1 depicts the system architecture: the AHU uses a mixture of outside air and return air to generate cool air by using a cooling coil (usually driven by chilled water, see [18] for optimal generation of chilled water). The cool air then is distributed by a fan to VAV boxes connected with each room. The damper position in the VAV box controls the mass flow rate of air entering a room. In addition, a heating coil in the VAV box is used to warm up the supply air if needed.

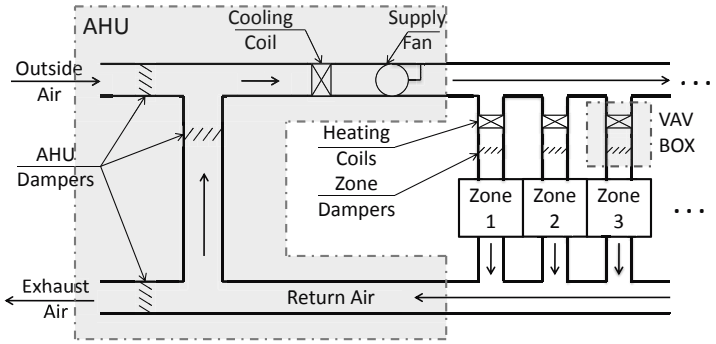


Figure 1.1. *HVAC System scheme*

In order to develop a simplified yet descriptive model, the following assumptions are introduced.

A1 The system pressure dynamics are not considered.

A2 The dynamics of each component (AHU, VAV boxes, and fan) are neglected. This implies that the supply air temperature and flow set points are tracked perfectly.

A3 Air temperature is constant through the ducts.

A4 The amount of air exiting the rooms is the same as the amount of air entering the rooms. This neglects the infiltration and exhalation of air through the building envelope.

1.2.1 Simplified System Model

We use an undirected graph structure to represent the rooms and their dynamic couplings in the following way. We associate the i-th room with the i-th node of a graph, and if an edge (i, j) connecting the i-th and j-th node is present, the rooms i and j are subject to direct heat transfer. The graph $\mathcal{G}$ will be defined as

$$\mathcal{G} = (\mathcal{V}, \mathcal{A}), \tag{1.1}$$

where $\mathcal{V}$ is the set of nodes (or vertices) $\mathcal{V} = \{1, \ldots, N_v\}$ and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges (i, j) with $i \in \mathcal{V},\ j \in \mathcal{V}$. We denote $\mathcal{N}^i$ the set of neighboring nodes of i, i.e., $j \in \mathcal{N}^i$ if and only if $(i, j) \in \mathcal{A}$.

Now consider a single room $j \in \mathcal{V}$. The air enters the room j with a mass flow rate $\dot{m}_s^j$. It is assumed that in the AHU, the outside air fully mixes with the return air without delay, and the mixing proportion δ between the return air and outside air is controlled by the damper configurations in the AHU system to obtain:

$$T_m = \delta T_r + (1 - \delta) T_{oa}, \tag{1.2}$$

where T_{oa} is the outside air temperature, and T_m is the temperature of the mixed air. T_r is the return air temperature calculated as weighted average temperature of return air from each room

$$T_r = \sum_{i \in \mathcal{V}} \dot{m}_s^i T^i / \sum_{i \in \mathcal{V}} \dot{m}_s^i. \tag{1.3}$$

The return air is not recirculated when $\delta = 0$, and no outside fresh air is used when $\delta = 1$. δ can be used to save energy through recirculation but it has to be strictly less than one to guarantee a minimal outdoor fresh air delivered to the rooms.

We model the room as a two-mass system. C_1^j is the fast-dynamic mass that has lower thermal capacitance (e.g. air around VAV diffusers), and C_2^j represents the slow-dynamic mass that has higher thermal capacitance (e.g. solid parts which include floor, walls and furniture). We remark that the phenomenon of fast and slow dynamics has been observed in [19]. The thermal dynamic model of a room is:

$$\begin{aligned} C_1^j \frac{dT_1^j}{dt} &= \dot{m}_s^j c_p (T_m - \Delta T_c + \Delta T_h^j - T_1^j) + (T_2^j - T_1^j)/R^j \\ &\quad + (T_{oa} - T_1^j)/R_{oa}^j + \sum_{i \in \mathcal{N}^j} (T_1^i - T_1^j)/R_{ij} + P_d^j, \end{aligned} \tag{1.4a}$$

$$C_2^j \frac{dT_2^j}{dt} = (T_1^j - T_2^j)/R^j,\ \forall j \in \mathcal{V}, \tag{1.4b}$$

$$T_m = \delta T_r + (1 - \delta) T_{oa}, \tag{1.4c}$$

$$\sum_{i \in \mathcal{V}} \dot{m}_s^i T_r = \sum_{i \in \mathcal{V}} \dot{m}_s^i T^i, \tag{1.4d}$$

$$T^j = T_1^j,\ \forall j \in \mathcal{V}, \tag{1.4e}$$

where T_1^j and T_2^j are system states representing the temperature of the lumped masses C_1^j and C_2^j, respectively. The supply air temperature difference across the

Table 1.1. *Identification results for conference room model on July 4th, 2010*

Parameter	Value	Parameter	Value
C_1^1	9.163×10^3 kJ/K	R_{12}	2.000 K/kW
C_2^1	1.694×10^5 kJ/K	R_{oa}^1	57 K/kW
R^1	1.700 K/kW		

cooling coil in the AHU is denoted as ΔT_c, and ΔT_h^j is the air temperature difference across the heating coil in the j-th VAV box. T^j is the perceived temperature of room j, which is assumed to be equal to the temperature of the fast-dynamic mass C_1^j. R_{oa}^j is the thermal resistance between room j and outside air, and c_p is the specific heat capacity of room air. R^j models the heat resistance between C_1^j and C_2^j, $R_{ij} = R_{ji}$ models thermal resistances between room i and the adjacent room j, and P_d^j is an unmeasured load induced by external factors such as occupancy, equipment, and solar radiation.

The model (1.4) is tested to capture the temperature dynamics of a thermal zone in the Bancroft library located on the campus of University of California at Berkeley. By using historical data we have identified the model parameters for each thermal zones and validated the resulting model. The dimension of the conference room is $5 \times 4 \times 3$ m, and it has one door and no windows. As a result, the effect of solar radiation is negligible. The major source of load derives from occupants and electronic equipment. The conference room has one neighboring office room ($\mathcal{N}^1 = \{2\}$).

The model parameters ($[C_1^1,\ C_2^1,\ R^1,\ R_{12},\ R_{oa}^1]$) are identified by using a nonlinear regression algorithm using measured data collected on July 4th, 2010 over 24 hours. This corresponds to a Sunday when the conference room has no occupants ($P_d^1 = 0$). Measurements of room temperature (T^1), supply air temperature ($T_s^1 = T_m - \Delta T_c + \Delta T_h^1$), mass flow rate of the supply air ($\dot{m}_s^1$), the neighboring room temperature (T^2), and outside air temperature T_{oa} are used for the identification. The identified parameters values are reported in Table 1.1.

The identification results plotted in Figure 1.2 show that the proposed model successfully captures the thermal dynamics of the conference room without occupants. In Figure 1.2 the solid line depicts the measured room temperature trend and the dashed line is the room temperature predicted by model (1.4) when driven by the measured inputs.

The proposed model (1.4) with the identified parameters in Table 1.1 is validated against measurements during other weekends. Figure 1.3 plots the validation results for July 11th, 2010. One can observe that the predictions match well the experimental data.

The load prediction $P_d^1(t)$ is important for designing predictive feedback controllers and assessing potential energy savings. The disturbance load envelopes can be learned from historical data, shared calendars, and weather predictions. For instance the conference room discussed earlier has two regularly scheduled group meetings around 10:00 and 14:00 every Wednesday. By using historical data we can observe this from the data. Figure 1.4 depicts the envelope-bounded load during all Wednesdays in July, 2010 (Figure 1.4). The envelope is computed as pointwise

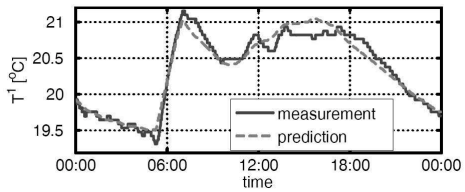


Figure 1.2. *Identification results of the thermal zone model (1.4)*

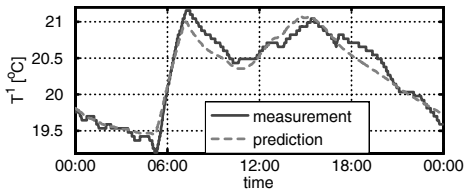


Figure 1.3. *Validation results of simplified room model (1.4)*

minimum and maximum difference between the measured data and the nominal model (i.e, model (1.4) with the identified parameters in Table 1.1 and $P_d^1 = 0$). Two peaks can be observed in the disturbance load envelope in Figure 1.4, which

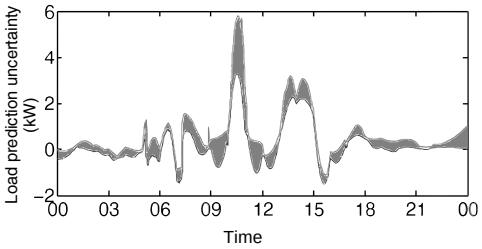


Figure 1.4. *Envelope bounds of disturbance load profile (kw) calculated using model (1.4) for Wednesdays in July 2010*

correspond to the two regularly scheduled group meetings. In the remainder of this paper, a nominal MPC is designed based on the load prediction profile with the highest probability. Ongoing research is focusing on stochastic MPC, where the load envelope with corresponding probability distribution function can be used at the controller design stage.

1.2.2 Simplified Energy Consumption Model

The components at the lower level of the architecture that use energy include dampers, supply fans, heating coils, and cooling coils as shown in Figure 1.1. The supply fan needs electrical power to drive the system while the heating and cooling coils consume the energy of the chilled and hot water. It is assumed that the power to drive the dampers is negligible. A simple energy consumption model for each component is presented next.

Fan Power The fan power can be approximated as a second order polynomial function of the total supply air mass flow rate[1] $\dot{m}_{\text{fan}}$ driven by the fan.

$$P_f = c_0 + c_1 \dot{m}_{\text{fan}} + c_2 \dot{m}_{\text{fan}}^2, \tag{1.5}$$

where c_0, c_1, c_2 are parameters to be identified by fitting recorded data. The

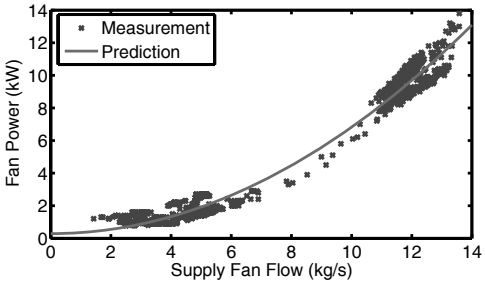


Figure 1.5. *Identification results for the simplified fan power consumption model (1.5).*

simplified fan model (1.5) is tested on the recorded data from the UC Berkeley Bancroft Library from October 1st to October 10th 2010. The identification results plotted in Figure 1.5 suggest that the polynomial function successfully predicts the electricity consumption of the fan.

Cooling and Heating Coils Cooling coils and heating coils are air-water heat exchangers. There has been extensive studies to develop simplified yet descriptive models of coil units [20, 21, 22]. The authors in [20] developed simple empirical equations with four parameters by using a finite difference method to capture the transient response of counterflow heat exchangers. In [21], the authors presented an improved simulation model based on ASHRAE Secondary HVAC Toolkit, and in [22] a simplified control oriented cooling coil unit model is presented based on energy and mass conservation laws.

In this work, we use a simple coil model with constant efficiency (η_c for the cooling coils and and η_h for the heating coils). With this simplification the energy consumption model is a static function of the load on the air-side

$$P_c = \frac{\sum_{j\in\mathcal{V}} \dot{m}_s^j c_p \Delta T_c}{\eta_c \,\text{COP}_c}, \;\; P_h^j = \frac{c_p \dot{m}_s^j \Delta T_h^j}{\eta_h \,\text{COP}_h}, \tag{1.6}$$

where P_c is the electrical power consumption related to the generation of chilled water consumed by the cooling coils in AHU and P_h^j is the power used to generate the hot water consumed by the heating coils in the VAV box connected with room j. ΔT_c is the temperature difference through the cooling coils, ΔT_h^j the temperature difference through the heating coils in the VAV box j, COP_c is the chilling coefficient

[1] The supply air mass flow rate by the fan is equal to the summation of air flow to each room $\dot{m}_{\text{fan}} = \sum_{j\in\mathcal{V}} \dot{m}_s^j$.

of performance, and COP_h is the heating coefficient of performance. The coefficient of performance (COP) is defined as

$$\mathrm{COP} = \frac{E_{\mathrm{thermal}}}{E_{\mathrm{input}}}. \tag{1.7}$$

COP captures the efficiency of the exchange system, i.e., the amount of thermal energy E_{thermal} (J) generated by the system with one Joule of energy consumed. The input energy E_{input} can be from different resources such as electricity, fuel, and gas for different systems. Model (1.6) is oversimplified as compared to the aforementioned literature. However, the model is adequate to capture the energy consumption if coils are operating in a narrow performance range.

1.2.3 Constraints

The states and control inputs of system (1.4) are subject to the following constraints (for all $j \in \mathcal{V}$):

1. $T^j \in [\underline{T},\ \overline{T}]$. Comfort range.
2. $\dot{m}_s^j \in [\underline{\dot{m}},\ \overline{\dot{m}}]$. The maximum mass flow rate of air supplied to a room is limited by the size of VAV boxes. The minimum mass flow rate is imposed to guarantee a minimal ventilation level.
3. $\Delta T_c \in [\underline{\Delta T_c},\ \overline{\Delta T_c}]$. The temperature decrement of the supply air (cooled by the cooling coil) is constrained by the capacity of the AHU.
4. $\Delta T_h^j \in [\underline{\Delta T_h},\ \overline{\Delta T_h}]$. The temperature increment of the supply air (heated by the heating coil) is constrained by the capacity of the VAV boxes.
5. $\delta \in [\underline{\delta},\ \overline{\delta}]$. The AHU damper position is positive and less than $\overline{\delta} < 1$ to make sure that there is always fresh outside air supplied to office rooms.

1.2.4 Model Summary

The thermal dynamic model (1.4) is compactly rewritten as

$$\frac{dx^j}{dt} = f_c(x^j, u^j, u^a, T_m, w^j) + \sum_{i \in \mathcal{N}^j} E_i^j x^i,\ \forall j \in \mathcal{V}, \tag{1.8a}$$

$$g_c(x^{j\in\mathcal{V}}, u^{j\in\mathcal{V}}, u^a, w^{j\in\mathcal{V}}, T_r, T_m) = 0, \tag{1.8b}$$

where $x^j = (T_1^j,\ T_2^j)$ is the state of the j-th room, $u^j = (\dot{m}_s^j,\ \Delta T_h^j)$ are the control inputs to the j-th VAV box, and $u^a = (\delta,\ \Delta T_c)$ collects the AHU control inputs. The vector $w^j = (P_d^j,\ T_{oa})$ is the disturbance assumed to be perfectly known. Equation (1.8b) lumps up the algebraic equations (1.4c) and (1.4d) that describe the static model for return air temperature and mixed air temperature, respectively. The set $x^{j\in\mathcal{V}}$ is defined as $\left\{x^1, x^2, \ldots, x^{\mathcal{N}_v}\right\}$. Note that the room dynamics in the

network are coupled through states (the second term in (1.8a)) and inputs (δ and ΔT_c are common to all rooms).

The continuous time dynamic system (1.8) is discretized. We consider the following control law with a uniform sampling time Δt,

$$u^j(t) = u^j_k,\ \forall j \in \mathcal{V}, \tag{1.9a}$$
$$u^a(t) = u^a_k, \tag{1.9b}$$
$$t \in [k\Delta t,\ (k+1)\Delta t).$$

With the control law defined by (1.9), the system equations (1.8) are discretized over $t \in [k\Delta t,\ (k+1)\Delta t)$ using the trapezoidal method to obtain:

$$\frac{x^j_{k+1} - x^j_k}{\Delta t} = \frac{1}{2} f_c(x^j_k, u^j_k, u^a_k, T_{mk}, w^j_k) + \frac{1}{2} f_c(x^j_{k+1}, u^j_k, u^a_k, T^+_{mk}, w^j_{k+1}) + \sum_{i \in \mathcal{N}^j} E^j_i \frac{x^i_k + x^i_{k+1}}{2},\ \forall j \in \mathcal{V}, \tag{1.10a}$$
$$g_c(x^{j\in\mathcal{V}}_k, u^{j\in\mathcal{V}}_k, u^a_k, w^{j\in\mathcal{V}}_k, T_{rk}, T_{mk}) = 0, \tag{1.10b}$$
$$g_c(x^{j\in\mathcal{V}}_{k+1}, u^{j\in\mathcal{V}}_k, u^a_k, w^{j\in\mathcal{V}}_{k+1}, T^+_{r\,k}, T^+_{m\,k}) = 0, \tag{1.10c}$$

where $T^+_{r\,k}$ and $T^+_{m\,k}$ are the temperature of return air and mixed air when time approaches $(k+1)\Delta t$, respectively. Note that $T^+_{r\,k}$ ($T^+_{m\,k}$) is generally different from T_{rk+1} (T_{mk+1}) due to the discontinuity introduced by the control law (1.9).

If $u^c_k = (u^a_k,\ T_{rk},\ T_{mk},\ T^+_{r\,k},\ T^+_{m\,k})$, then the discretized room model (1.10) can be compactly rewritten as

$$f(x^j_{k+1}, x^j_k, u^j_k, u^c_k, w^j_k, w^j_{k+1}) + \sum_{i \in \mathcal{N}^j} E^j_i \frac{x^i_k + x^i_{k+1}}{2} = 0,\ \forall j \in \mathcal{V}, \tag{1.11a}$$
$$g(x^{j\in\mathcal{V}}_{k+1}, x^{j\in\mathcal{V}}_k, u^{j\in\mathcal{V}}_k, u^c_k, w^{j\in\mathcal{V}}_k, w^{j\in\mathcal{V}}_{k+1}) = 0, \tag{1.11b}$$
$$u^c_k \in \mathcal{U}^c;\ x^j_k \in \mathcal{X},\ u^j_k \in \mathcal{U}^j,\ \forall j \in \mathcal{V}. \tag{1.11c}$$

The constraints (1.11c) are defined in Section 1.2.3.

In the next section we will also use the following linearized version of model (1.11) around the trajectory of states and control inputs $(\bar{x}^1_k, \ldots, \bar{x}^{N_v}_k,\ \bar{u}^1_k, \ldots, \bar{u}^{N_v}_k,\ \bar{u}^c_k),\ k = 0, 1, \ldots, N-1$:

$$T^j_k dx^j_{k+1} + A^j_k dx^j_k + B^j_k du^j_k + B^c_k du^c_k + f^{e\,j}_k + \sum_{i \in \mathcal{N}^j} E^j_i \frac{dx^i_{k+1} + dx^i_k}{2} = 0, \tag{1.12a}$$
$$\forall j \in \mathcal{V}, \tag{1.12b}$$
$$G^c_k du^c_k + \sum_{j \in \mathcal{V}} G^+_{j,k} dx^j_{k+1} + \sum_{j \in \mathcal{V}} G^x_{j,k} dx^j_k + \sum_{j \in \mathcal{V}} G^u_{j,k} du^j_k + g^e_k = 0, \tag{1.12c}$$
$$\bar{u}^c_k + du^c_k \in \mathcal{U}^c,\ \bar{x}^j_k + dx^j_k \in \mathcal{X},\ \ \bar{u}^j_k + du^j_k \in \mathcal{U}^j,\ \forall j \in \mathcal{V}, \tag{1.12d}$$
$$T^j_k = \left.\frac{\partial f}{\partial x^j_{k+1}}\right|_{\bar{x}^j_{k+1}},\ A^j_k = \left.\frac{\partial f}{\partial x^j_k}\right|_{\bar{x}^j_k},\ B^j_k = \left.\frac{\partial f}{\partial u^j_k}\right|_{\bar{u}^j_k},\ B^c_k = \left.\frac{\partial f}{\partial u^c_k}\right|_{\bar{u}^c_k}, \tag{1.12e}$$
$$G^+_{j,k} = \left.\frac{\partial g}{\partial x^j_{k+1}}\right|_{\bar{x}^j_{k+1}},\ G^x_{j,k} = \left.\frac{\partial g}{\partial x^j_k}\right|_{\bar{x}^j_k},\ G^u_{j,k} = \left.\frac{\partial g}{\partial u^j_k}\right|_{\bar{u}^j_k},\ G^c_k = \left.\frac{\partial g}{\partial u^c_k}\right|_{\bar{u}^c_k}, \tag{1.12f}$$

$$f_k^{e\,j} = f(\bar{x}_{k+1}^j, \bar{x}_k^j, \bar{u}_k^j, \bar{u}_k^c, w_k^j, w_{k+1}^j) + \sum_{i \in \mathcal{N}^j} E_i^j \frac{\bar{x}_{k+1}^i + \bar{x}_k^i}{2}, \ \forall j \in \mathcal{V}, \tag{1.12g}$$

$$g_k^e = g(\bar{x}_{k+1}^{j\in\mathcal{V}}, \bar{x}_k^{j\in\mathcal{V}}, \bar{u}_k^{j\in\mathcal{V}}, \bar{u}_k^c, w_k^{j\in\mathcal{V}}, w_{k+1}^{j\in\mathcal{V}}), \tag{1.12h}$$

where dx_k^j, du_k^j, and du_k^c are the deviations of states and control inputs around the trajectory. f_k^e and g_k^e are residuals of nonlinear equality constraints (1.11a) and (1.11b).

1.3 DISTRIBUTED MODEL PREDICTIVE CONTROL

In this section we formalize the MPC control problem [3] and provide details on the distributed MPC (DMPC) design. We are interested in solving at each time step t the following optimization problem:

$$\min_{\mathbf{U},\mathbf{X}} \ J(\mathbf{U},\mathbf{X}) = \sum_{k=0}^{N-1} \left\{ P_c + P_{\text{fan}} + \sum_{j\in\mathcal{V}} P_h^j \right\} \Delta t \tag{1.13a}$$

subj. to:

$$f(x_{k+1|t}^j, x_{k|t}^j, u_{k|t}^j, u_{k|t}^c, w_{k|t}^j, w_{k+1|t}^j) + \sum_{i\in\mathcal{N}^j} E_i^j (x_{k+1|t}^i + x_{k|t}^i)/2 = 0,$$

$$\forall j \in \mathcal{V}, \ k = 0, 1, \ldots, N-1, \tag{1.13b}$$

$$g(x_{k+1|t}^{j\in\mathcal{V}}, x_{k|t}^{j\in\mathcal{V}}, u_{k|t}^{j\in\mathcal{V}}, u_{k|t}^c, w_{k|t}^{j\in\mathcal{V}}, w_{k+1|t}^{j\in\mathcal{V}}) = 0,$$

$$k = 0, 1, \ldots, N-1, \tag{1.13c}$$

$$x_{k|t}^j \in \mathcal{X}^j, \ \forall j \in \mathcal{V}, \ k = 1, \ldots, N, \tag{1.13d}$$

$$u_{k|t}^j \in \mathcal{U}^j, \ u_{k|t}^c \in \mathcal{U}^c, \ \forall j \in \mathcal{V}, \ k = 0, \ldots, N-1, \tag{1.13e}$$

$$x_{0|t}^j = x^j(t), \ \forall j \in \mathcal{V}, \tag{1.13f}$$

where $\mathbf{U} = (u_{0|t}^1, \ldots, u_{N-1|t}^1, \ldots, u_{0|t}^{N_v}, \ldots, u_{N-1|t}^{N_v}, u_{0|t}^c, \ldots, u_{N-1|t}^c)$ lumps up u^c and $u^{j\in\mathcal{V}}$ over the prediction horizon, and $\mathbf{X} = (x_{0|t}^1, \ldots, x_{N|t}^1, \ldots x_{0|t}^{N_v}, \ \ldots, x_{N|t}^{N_v})$ is the vector of system state predictions over the prediction horizon. Let $\mathcal{T} = (\mathbf{X}, \mathbf{U})$ be the vector collecting all optimization variables. The cost function in (1.13) is the total energy consumed by all VAV boxes and the AHU system over the prediction horizon.

In (1.13) $x_{k|t}^j$ denotes the state of room j at time $t + k\Delta t$ predicted at time t starting from the current state $x_{0|t}^j = x^j(t), \ \forall j \in \mathcal{V}$.

Let $(\mathbf{U}^\star, \mathbf{X}^\star)$ be the optimal solution of problem (1.13). Then, only the first element of every control sequence is implemented to the system, i.e. $u^j(t) = u_{0|t}^{j\,\star}$, $u^a(t) = u_{0|t}^{a\,\star}$.

The optimization (1.13) is repeated at time $t + \Delta t$, with the updated state $x_{0|t+\Delta t} = x(t + \Delta t)$, yielding a *moving or receding horizon control* strategy.

The MPC problem (1.13) has a non-convex cost (1.13a) which includes bilinear terms for the energy consumption of cooling and heating coils, bilinear equality

constraints (1.13b) and (1.13c), and box constraints on system states and control inputs. The size of the nonlinear optimization problem rapidly grows when a realistic number of rooms and a meaningful horizon length N are considered. In order to solve the MPC problem (1.13) in a distributed fashion, we apply sequential quadratic programming (SQP), proximal minimization, and dual decomposition. Next we show the main idea of these techniques and implementation details for the specific class of problems considered in this paper.

The SQP procedure is an efficient method to solve nonlinear programming problems [11, 10]. The basic idea is to linearize the nonlinear constraints around a candidate solution and replace the objective with a quadratic function around this guess. The solution to the resulting QP is then used to update the candidate solution. The iterations are repeated until convergence is achieved [11]. Note that the hessian of the cost (1.13a) is, in general, indefinite since the cost is nonconvex.

The proximal minimization algorithm is adopted to solve the indefinite QP problem obtained from the SQP procedure by iteratively solving a set of subproblems whose cost functions have extra convex quadratic terms [12]. By doing so, the primal cost function of the subproblem is strictly convex, which implies differentiability of the dual cost function. This enables faster convergence in the procedure of dual decomposition [12].

The concept of dual decomposition traces back to the 70's [13], and it has been extensively studied since then [12, 23]. The QP derived from the proximal minimization algorithm is a separable convex optimization problems and therefore the gradient of the dual problem can be calculated in a distributed fashion, and the dual variables can be optimized separately by using gradient or subgradient based approaches. The primal optimal solution then can be reconstructed from the dual variables.

To summarize, the optimal solution to Problem (1.13) is obtained through three nested iterative algorithms. The outer iteration solves the original nonlinear optimization problem (1.13) by solving a sequence of QPs, the second and third levels of iteration solve the QP in a distributed fashion by using proximal minimization and dual decomposition, respectively.

The details of the proposed algorithms are described next. Problem (1.13) is time-variant because of the disturbance load profile w. With abuse of notation and for the sake of simplicity, in the rest of the paper we will remove the term "$|t$" from the lower indices.

1.3.1 Level 1: Modified Sequential Quadratic Programming

At the SQP iteration n_s, problem (1.13) is linearized by replacing the nonlinear system dynamics (1.13b) with the linearized ones (1.12) at a candidate solution $\mathcal{T}^{n_s} = (\mathbf{U}^{n_s}, \mathbf{X}^{n_s})$. The cost function (1.13a) is approximated by a quadratic function around $\mathcal{T}^{n_s}$ while neglecting the off-diagonal terms. The resultant optimization problem (1.14) is convex.

$$\min_{\mathbf{dU},\mathbf{dX}} dJ(\mathbf{dU},\mathbf{dX}) = \sum_{k=0}^{N-1} \left\{ \sum_{j\in\mathcal{V}} \left(\frac{1}{2} {du_k^j}^T Q_k^j du_k^j + {c_k^j}^T du_k^j \right) + \frac{1}{2} {du_k^c}^T Q_k^c du_k^c \right.$$

$$+c_k^{c\,T} du_k^c \Big\} \tag{1.14a}$$

subject to:

$$T_k^j dx_{k+1}^j + A_k^j dx_k^j + B_k^j du_k^j + B_k^c du_k^c + f_k^{e\,j} + \sum_{i\in\mathcal{N}^j} E_i^j \frac{dx_{k+1}^i + dx_k^i}{2} = 0,\ \forall j \in \mathcal{V},\ k = 0, 1, \ldots, N-1, \tag{1.14b}$$

$$G_k^c du_k^c + \sum_{j\in\mathcal{V}} G_{j,k}^+ dx_{k+1}^j + \sum_{j\in\mathcal{V}} G_{j,k}^x dx_k^j + \sum_{j\in\mathcal{V}} G_{j,k}^u du_k^j + g_k^e = 0,$$
$$k = 0, 1, \ldots, N-1, \tag{1.14c}$$

$$dx_k^j \in d\mathcal{X}^j,\ \forall j \in \mathcal{V},\ k = 1, \ldots, N, \tag{1.14d}$$

$$du_k^j \in d\mathcal{U}^j,\ du_k^c \in d\mathcal{U}^c,\ \forall j \in \mathcal{V},\ k = 0, \ldots, N-1, \tag{1.14e}$$

where Q_k^c and Q_k^j are diagonal matrices obtained by removing the off-diagonal terms of the hessian of (1.13a) at candidate solution $\mathcal{T}^{n_s}$. The vector $\mathbf{dU} = (du_0^1, \ldots, du_{N-1}^1, \ldots, du_0^{N_v}, \ldots, du_{N-1}^{N_v}, du_0^c, \ldots, du_{N-1}^c)$ collects control input difference from the candidate solution $\mathbf{U}^{n_s}$, and the vector $\mathbf{dX} = (dx_1^1, \ldots, dx_N^1, \ldots, dx_1^{N_v}, \ldots, dx_N^{N_v})$ collects system state deviations from the trajectory $\mathbf{X}^{n_s}$. The constraint sets $d\mathcal{X}^j = \mathcal{X}^j - x_k^{j\,n_s}, d\mathcal{U}^j = \mathcal{U}^j - u_k^{j\,n_s}$, and $d\mathcal{U}^c = \mathcal{U}^c - u_k^{c\,n_s}$ define the feasible state variations $\mathbf{dX}$ and control input variations $\mathbf{dU}$, respectively.

The optimal solution $d\mathcal{T}^\star = (\mathbf{dX}^\star,\ \mathbf{dU}^\star)$ to problem (1.14) is computed by the iterative algorithms at level 2 and level 3 described next. The vector $d\mathcal{T}^\star$ is used to update the candidate solution as

$$\mathcal{T}^{n_s+1} = \mathcal{T}^{n_s} + \alpha \cdot d\mathcal{T}^\star.$$

In this work, a constant step length α is applied. At the SQP iteration $n_s + 1$, the process of linearizing problem (1.13) and solving problem (1.14) is repeated. The SQP algorithm is terminated if

$$\|d\mathcal{T}^\star\| \le \kappa, \tag{1.15}$$

where κ is a predefined convergence tolerance.

Remark 1. *The convergence of the proposed SQP algorithm for general non-convex programs is not guaranteed. Extensive numerical tests have failed in finding an instance of the problem considered in this paper where the proposed algorithm would not converge. It is noted that the convergence rate of SQP procedure can be improved when an adaptive step length α is selected using an advanced line search algorithm [10].*

1.3.2 Level 2: Proximal Minimization

The iterative algorithm at the second level solves problem (1.14) by using the proximal minimization algorithm proposed in [12]. The quadratic objective in Problem (1.14) is positive semi-definite. Its solution is obtained by optimizing a sequence of subproblems obtained by adding a quadratic term to the original cost. At the

proximal minimization iteration n_p, we consider the subproblem

$$\min_{\mathbf{d\widehat{U}},\mathbf{d\widehat{X}}} d\widehat{J}^{n_p}(\mathbf{d\widehat{U}},\mathbf{d\widehat{X}}) = \sum_{k=0}^{N-1}\left\{\sum_{j\in\mathcal{V}}\left(\frac{1}{2}d\widehat{u}_k^{j\,T}Q_k^j d\widehat{u}_k^j + c_k^{j\,T}d\widehat{u}_k^j\right) + \frac{1}{2}d\widehat{u}_k^{c\,T}Q_k^c d\widehat{u}_k^c + c_k^{c\,T}d\widehat{u}_k^c\right\}$$
$$+\frac{\rho}{2}\left(\|\mathbf{d\widehat{U}} - \mathbf{d\widehat{U}}^{n_p-1}\|_2^2 + \|\mathbf{d\widehat{X}} - \mathbf{d\widehat{X}}^{n_p-1}\|_2^2\right) \quad (1.16a)$$

subj. to:

$$T_k^j d\widehat{x}_{k+1}^j + A_k^j d\widehat{x}_k^j + B_k^j d\widehat{u}_k^j + B_k^c d\widehat{u}_k^c + f_k^{e\,j}$$
$$+\sum_{i\in\mathcal{N}^j} E_i^j \frac{d\widehat{x}_{k+1}^i + d\widehat{x}_k^i}{2} = 0,\ \forall j\in\mathcal{V},\ k=0,1,\ldots,N-1, \quad (1.16b)$$

$$G_k^c d\widehat{u}_k^c + \sum_{j\in\mathcal{V}} G_{j,k}^+ d\widehat{x}_{k+1}^j + \sum_{j\in\mathcal{V}} G_{j,k}^x d\widehat{x}_k^j + \sum_{j\in\mathcal{V}} G_{j,k}^u d\widehat{u}_k^j + g_k^e = 0,$$
$$k=0,1,\ldots,N-1, \quad (1.16c)$$

$$d\widehat{x}_k^j \in d\mathcal{X}^j,\ \forall j\in\mathcal{V},\ k=1,\ldots,N, \quad (1.16d)$$

$$d\widehat{u}_k^j \in d\mathcal{U}^j,\ d\widehat{u}_k^c \in d\mathcal{U}^c,\ \forall j\in\mathcal{V},\ k=0,\ldots,N-1, \quad (1.16e)$$

where $\rho > 0$ is a strictly positive constant predetermined such that the resulting cost (1.16a) is positive definite. $\left(\mathbf{d\widehat{X}}^{n_p-1},\ \mathbf{d\widehat{U}}^{n_p-1}\right)$ is the optimal solution to Problem (1.16) at iteration n_p-1. When $n_p = 1$, let $\mathbf{d\widehat{X}}^0 = \mathbf{0},\ \mathbf{d\widehat{U}}^0 = \mathbf{0}$.

The optimal solution $(\mathbf{d\widehat{X}}^\star, \mathbf{d\widehat{U}}^\star)$ to problem (1.16) is computed by the algorithm at level 3. We set $\mathbf{d\widehat{X}}^{n_p} = \mathbf{d\widehat{X}}^\star$ and $\mathbf{d\widehat{U}}^{n_p} = \mathbf{d\widehat{U}}^\star$, and terminate the iterative algorithm if

$$\|\widehat{d\mathbf{U}}^{n_p} - \widehat{d\mathbf{U}}^{n_p-1}\|_2 \le \kappa,\ \|\widehat{d\mathbf{X}}^{n_p} - \widehat{d\mathbf{X}}^{n_p-1}\|_2 \le \kappa. \quad (1.17)$$

It has been proved in [12] that if the optimization problem (1.14) is convex, then the vector $(\mathbf{d\widehat{X}}^{n_p},\ \mathbf{d\widehat{U}}^{n_p})$ converges to an optimum $(\mathbf{dX}^\star,\ \mathbf{dU}^\star)$ of problem (1.14).

1.3.3 Level 3: Dual Decomposition and Fast Gradient Method

The dual decomposition algorithm is used to solve problem (1.16) by solving its dual problem. The dual problem of the QP (1.16) is formulated by assigning dual variables λ_k^j and μ_k to the constraints (1.16b) and (1.16c), respectively. The dual problem can be formulated as follows:

$$\max_{\lambda,\mu\ \text{free}} \min_{\mathbf{d\widehat{U}},\mathbf{d\widehat{X}}} d\widehat{J}^{n_p} + L^c + L^f \quad (1.18a)$$

subj. to

$$d\widehat{x}_k^j \in d\mathcal{X}^j,\ \forall j\in\mathcal{V},\ k=1,\ldots,N, \quad (1.18b)$$

$$d\widehat{u}_k^j \in d\mathcal{U}^j,\ \forall j\in\mathcal{V},\ k=0,\ldots,N-1, \quad (1.18c)$$

$$d\widehat{u}_k^c \in d\mathcal{U}^c,\ \forall k=0,\ldots,N-1, \quad (1.18d)$$

where $\widehat{dJ}^{n_p}$ is the cost defined in (1.16a). The term

$$L^c = \sum_{k=1}^{N} \mu_k^T \left(G_k^c d\widehat{u}_k^c + \sum_{j\in\mathcal{V}} G_{j,k}^{+} d\widehat{x}_{k+1}^j + \sum_{j\in\mathcal{V}} G_{j,k}^{x} d\widehat{x}_k^j + \sum_{j\in\mathcal{V}} G_{j,k}^{u} d\widehat{u}_k^j + g_k^e \right)$$

is the dual term corresponding to constraint (1.16c). The term

$$L^f = \sum_{j\in\mathcal{V}} \sum_{k=1}^{N} {\lambda_k^j}^T \left(T_k^j d\widehat{x}_{k+1}^j + A_k^j d\widehat{x}_k^j + B_k^j d\widehat{u}_k^j + B_k^c d\widehat{u}_k^c + {f_k^e}^j + \sum_{i\in\mathcal{N}^j} E_i^j \frac{d\widehat{x}_{k+1}^i + d\widehat{x}_k^i}{2} \right)$$

is the dual term for constraint (1.16b).

In our previous work [1], the dual problem is solved by a projected subgradient method with a constant step size, which suffers from relatively slow convergence. The algorithm convergence speed can be improved by applying the fast gradient method since the dual function becomes Lipschitz-smooth when a proximal terms is added [24, 25].

We note that the cost function (1.18a) and the constraints of the inner minimization problem (1.18) are separable. This special structure allows us to solve problem (1.18) using the fast gradient method in a distributed way as described next.

Three sets of variables are updated in the fast gradient method [25], namely the dual variables λ and μ, the auxiliary variables $\bar{\lambda}$ and $\bar{\mu}$ of the same dimension as the dual variables, and a parameter γ controlling the step size β. They are initialized as $\lambda^0 = {\lambda^\star}^{n_p-1}$, $\mu^0 = {\mu^\star}^{n_p-1}$, $\bar{\lambda}^0 = \lambda^0$, $\bar{\mu}^0 = \mu^0$, and $\gamma^0 = \frac{\sqrt{5}-1}{2}$. Note that for $n_p = 1$, we set ${\lambda^\star}^{n_p-1} = \mathbf{0}$ and ${\mu^\star}^{n_p-1} = \mathbf{0}$.

The optimal solution $\left({\lambda^\star}^{n_p-1},\ {\mu^\star}^{n_p-1}\right)$ to the dual problem at the second level iteration $n_p - 1$ is used as a warm start for faster convergence.

At the third level iteration n_d, the dual and auxiliary variables are updated by using the fast gradient method as follows [25]:

$$\lambda_k^{j\,n_d} = \bar{\lambda}_k^{j\,n_d-1} + \frac{1}{L} h_{\lambda_k^j}(\bar{\lambda}_k^{j\,n_d-1}),\ \forall j \in \mathcal{V},\quad \forall k = 1, 2, \ldots, N, \tag{1.19a}$$

$${\mu_k}^{n_d} = {\bar{\mu}_k}^{n_d-1} + \frac{1}{L} h_{\mu_k}(\bar{\mu}_k^{n_d-1}),\ \forall k = 0, 1, \ldots, N-1, \tag{1.19b}$$

$$\gamma^{n_d} = \frac{\gamma^{n_d-1}}{2}\left(\sqrt{(\gamma^{n_d-1})^2 + 4} - \gamma^{n_d-1}\right), \tag{1.19c}$$

$$\beta = \frac{\gamma^{n_d-1}(1-\gamma^{n_d-1})}{(\gamma^{n_d-1})^2 + \gamma^{n_d}}, \tag{1.19d}$$

$$\bar{\lambda}_k^{j\,n_d} = \lambda_k^{j\,n_d} + \beta(\lambda_k^{j\,n_d} - \lambda_k^{j\,n_d-1}),\ \forall j \in \mathcal{V},\quad \forall k = 1, 2, \ldots, N, \tag{1.19e}$$

$${\bar{\mu}_k}^{n_d} = {\mu_k}^{n_d} + \beta({\mu_k}^{n_d} - {\mu_k}^{n_d-1}),\ \forall k = 0, 1, \ldots, N-1, \tag{1.19f}$$

where L is the Lipschitz constant of the dual gradient calculated as in [25]. $h_{\lambda_k^j}(\bar{\lambda}_k^{j\,n_d-1})$ and $h_{\mu_k}(\bar{\mu}_k^{n_d-1})$ are the gradients of the dual cost function (1.18a) at $\bar{\lambda}_k^{j\,n_d-1}$ and ${\bar{\mu}_k}^{n_d-1}$, respectively, The fast gradient method algorithm is terminated if

$$\|h_{\lambda_k^j}(\bar{\lambda}_k^{j\,n_d})\|_2 \le \kappa,\ \forall j \in \mathcal{V},\ k = 1, 2, \ldots, N, \tag{1.20a}$$

$$\|h_{\mu_k}(\bar{\mu}_k^{n_d})\|_2 \le \kappa,\ k = 1, 2, \ldots, N. \tag{1.20b}$$

The computation of the gradients $h_{\lambda_k^j}(\bar{\lambda}_k^{j\,n_d})$ and $h_{\mu_k}(\bar{\mu}_k^{n_d})$ is the most time consuming step of the algorithm. The proposed algorithm uses the following approach.

First, we compute the primal variables in (1.18) associated to the set of dual variables $\left(\bar{\lambda}^{n_d}, \bar{\mu}^{n_d}\right)$:

$$d\widehat{x}_k^{\star\, j\, n_d} = \Pi_{d\mathcal{X}^j}\left(d\widehat{x}_k^{j\, n_p-1} + \frac{1}{\rho}(-G_{j,k}^{+\,T}\bar{\mu}_{k-1}^{n_d} - G_{j,k}^{x\,T}\bar{\mu}_k^{\,n_d} - A_k^{j\,T}\bar{\lambda}_k^{j\,n_d} - T_k^{j\,T}\bar{\lambda}_{k-1}^{j\,n_d} - \sum_{i\in\mathcal{N}^j} E_j^{i\,T}\frac{\bar{\lambda}_{k-1}^{i\,n_d} + \bar{\lambda}_k^{i\,n_d}}{2})\right),\ \forall j\in\mathcal{V},\quad \forall k = 1,2,\ldots,N, \tag{1.21a}$$

$$d\widehat{u}_k^{\star\, j\, n_d} = \Pi_{d\mathcal{U}^j}\left((\rho I + Q_k^j)^{-1}(\rho d\widehat{u}_k^{j\, n_p-1} - c_k^j - G_{j,k}^{u\,T}\bar{\mu}_k^{\,n_d} - B_k^{j\,T}\bar{\lambda}_k^{j\,n_d})\right),$$
$$\forall j\in\mathcal{V},\quad \forall k = 0,1,\ldots,N-1, \tag{1.21b}$$

$$d\widehat{u}_k^{\star\, c\, n_d} = \Pi_{d\mathcal{U}^c}\left((\rho I + Q_k^c)^{-1}(\rho d\widehat{u}_k^{c\, n_p-1} - c_k^c - G_{c,k}^{u\,T}\bar{\mu}_k^{\,n_d} - \sum_{j\in\mathcal{V}} B_k^{c\,T}\bar{\lambda}_k^{j\,n_d})\right),$$
$$\forall k = 0,1,\ldots,N-1, \tag{1.21c}$$

where $\Pi_{\mathcal{S}}(\star)$ is the operation of projecting $\star$ onto the convex set $\mathcal{S}$.[2] Then, the gradients of the dual cost function (1.18a) at $(\bar{\mu}^{n_d},\ \bar{\lambda}^{n_d})$ can be computed as follows [25]:

$$h_{\lambda_k^j}(\bar{\lambda}_k^{j\,n_d}) = T_k^j d\widehat{x}_{k+1}^{\star\, j\ n_d} + A_k^j d\widehat{x}_k^{\star\, j\, n_d} + B_k^j d\widehat{u}_k^{\star\, j\, n_d} + B_k^c d\widehat{u}_k^{\star\, c\, n_d} + f_k^{e\,j} + \sum_{i\in\mathcal{N}^j} E_i^j \frac{d\widehat{x}_{k+1}^{\star\, i\ \ n_d} + d\widehat{x}_k^{\star\, i\, n_d}}{2}, \tag{1.22a}$$
$$\forall j\in\mathcal{V},\quad \forall k = 0,1,\ldots,N-1,$$

$$h_{\mu_k}(\bar{\mu}_k^{n_d}) = G_k^c d\widehat{u}_k^{\star\, c\, n_d} + \sum_{j\in\mathcal{V}} G_{j\,k}^{+} d\widehat{x}_{k+1}^{\star\, j\, n_d} + \sum_{j\in\mathcal{V}} G_{j\,k}^{x} d\widehat{x}_k^{\star\, j\, n_d} + \sum_{j\in\mathcal{V}} G_{j,k}^{u} d\widehat{u}_k^{\star\, j\, n_d} + g_k^e, \tag{1.22b}$$
$$\forall k = 0,1,\ldots,N-1.$$

In summary, the proposed MPC problem is solved locally by three nested levels of iterations: the outer iteration solves the original nonlinear optimization problem (1.13) by solving a sequence of QPs (1.14). The second and third levels of iteration solve the QP (1.14) in a distributed fashion by using proximal minimization and dual decomposition. Algorithm 1.1 summarizes the main steps of the proposed distributed model predictive control (DMPC) scheme.

Algorithm 1.1. DMPC for building control.

[2]Note that as Q_k^j and Q_k^c are diagonal matrices, the matrix inversion in (1.21b) and (1.21c) can be easily evaluated. The projection is easy to calculate as $d\mathcal{X}^j$, $d\mathcal{U}^j$, and $d\mathcal{U}^c$ are box constraints.

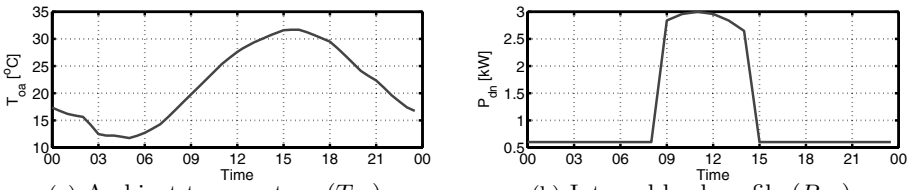
Initial:	Let $\mathbf{U}^1$, $\mathbf{X}^1$ be initial guesses for primal variables for Problem (1.13). Set the SQP iteration index to $n_s = 1$.
Step 1:	Let n_s be the current SQP iteration index. Linearize the system model (1.11) at $(\mathbf{U}^{n_s}, \mathbf{X}^{n_s})$ to obtain the coefficients in (1.14). Let $n_p = 1$, and set $\widehat{\mathbf{dU}}^0 = \mathbf{0}$ and $\widehat{\mathbf{dX}}^0 = \mathbf{0}$ in (1.16). Also let $\lambda^{\star 0} = \mathbf{0}$ and $\mu^{\star 0} = \mathbf{0}$.
Step 2:	Let n_p be the current proximal minimization iteration index, and set $n_d = 1$. Initialize $\widehat{\lambda}^1 = \lambda^1 = \lambda^{\star\, n_p - 1}$ and $\widehat{\mu}^1 = \mu^1 = \mu^{\star\, n_p - 1}$
Step 3:	Let n_d be the current dual decomposition iteration index. Update primal variables $(\widehat{d\mathbf{U}}^\star, \widehat{d\mathbf{X}}^\star)$ as in the distributed algorithm (1.21).
Step 4:	Exchange the updated primal variables and calculate the gradients (1.22) for the dual function.
Step 5:	Update the dual sequences λ^{n_d+1}, μ^{n_d+1}, $\widehat{\lambda}^{n_d+1}$, $\widehat{\lambda}^{n_d+1}$, and γ^{n_d+1} as in the distributed algorithm (1.19).
Step 6:	If condition (1.20) is satisfied, go to the next step. Otherwise, set $n_d = n_d + 1$ and go to Step 3.
Step 7:	Update $\widehat{d\mathbf{U}}^{n_p} = \widehat{d\mathbf{U}}^{\star\, n_d}$ and $\widehat{d\mathbf{X}}^{n_p} = \widehat{d\mathbf{X}}^{\star\, n_d}$. If condition (1.17) is satisfied, go to the next step. Otherwise, set $n_p = n_p + 1$ and go to Step 2.
Step 8	Set $d\mathbf{U}^\star = \widehat{d\mathbf{U}}^{n_p}$ and $d\mathbf{X}^\star = \widehat{d\mathbf{X}}^{n_p}$. If condition (1.15) is satisfied, terminate and the optimal control sequence is $\mathbf{U}^{n_s}$. Otherwise, update $\mathbf{U}^{n_s+1} = \mathbf{U}^{n_s} + \alpha \cdot d\mathbf{U}^\star$ and $\mathbf{X}^{n_s+1} = \mathbf{X}^{n_s} + \alpha \cdot d\mathbf{X}^\star$, set $n_s = n_s + 1$, and go to Step 1.

1.4 SIMULATION RESULTS

This section presents a numerical example to show the effectiveness of the proposed controller design methodology.

We compare the proposed control methodology with a baseline control logic (BC), which is a simplified version of a production control logic. The BC works as follows. When all the room temperatures are within the comfort range, the mass flow rate of the supply air ($\dot{m}_s^j$) is set to its minimum and the valves of cooling and heating coils are closed. When a room temperature hits the lower bound, the air mass flow rate to the room is maintained at its minimum, and the supply air temperature will be adjusted by the heating coil in the corresponding VAV box so that the room temperature stays at the lower bound value. When a room temperature violates the upper constraints, the AHU supply air temperature is set to its minimum, and the mass flow rate of the supply air is controlled so that the room temperature is within the comfort range.

The numerical example considers a network of 10 rooms. All the rooms have the same model parameters as in Table 1.1 identified for the conference room in the Bancroft library in Section 1.2. The undirected graph describing the topology of the room network is $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{1, 2, \ldots, 10\}$, and $\mathcal{A} = \{(1,2), (2,3), \ldots, (9,10)\}$. The weather information is downloaded from July 2nd to July 3rd, 2009 at UC Berkeley, and the temperature profile is plotted in Figure 1.6(a). Because of the warm weather, only cooling is critical in the considered scenario. Figure 1.6(b) depicts the nominal internal load profile (P_{dn}) in our simulations. We assume that during 15:00 and 8:00 the next day, the rooms are empty



(a) Ambient temperature (T_{oa}) (b) Internal load profile (P_{dn})

Figure 1.6. *Simulation setups*

Table 1.2. *Parameters for the numerical example*

param	value	param	value	param	value	param	value
$\underline{\dot{m}}$	0.005 kg/s	$\overline{\dot{m}}$	5 kg/s	η_c	0.7	η_h	0.8
$\underline{\Delta T_c}$	0 K	$\overline{\Delta T_c}$	8 K	P_f	0.08	α	0.25
$\underline{\Delta T_h}$	0 K	$\overline{\Delta T_h}$	8 K	$\underline{\delta}$	0	$\overline{\delta}$	0.8
$\overline{COP_c}$	5	COP_h	0.9	N	48	κ	1×10^{-3}

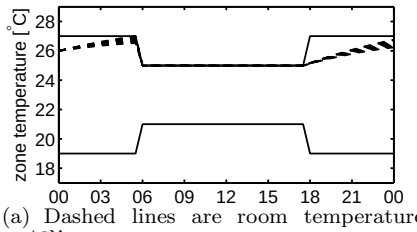
without occupancy, leaving a minimum internal load of 0.01 kW due to lighting or other electrical devices. We use an internal load profile different for each room. In particular, we compute the internal load for room j as

$$P_d{}^j = (1 + 0.2j)P_{dn}, \quad j = 1, 2, \ldots, 10.$$

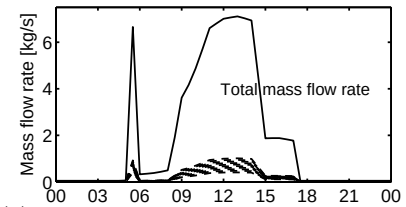
In our simulations, the parameters for controllers in Section 1.3 are listed in Table 1.2. The sampling time Δt is chosen to be thirty minutes, and the prediction horizon is one day ($N = 48$). The comfort constraints are defined as $[21,\ 25]^o$C from 6:00 to 18:00 when there is occupancy in the buildings, and the comfort set is relaxed to $[19,\ 27]^o$C when offices are unoccupied.

Figure 1.7 shows the simulation results for the network of rooms controlled by the baseline controller. Figure 1.7(a) shows that all the room temperatures are within the comfort range defined by the dotted lines. During early morning till 04:00, all zone temperatures are within the comfort range. As a result, the supply fan only maintains the minimum required air mass flow rate to each zone, and the valve of the cooling coils in the AHU is fully closed. The occupancy load at noon results in a peak total air mass flow rate of 7.2 kg/s. The cooling coils are operating at maximum capacity as soon as one of the zone temperatures hits the upper constraints so that the thermal comfort can be guaranteed. The return air damper position is fully closed to take advantage of free cooling when the ambient temperature is lower than the zone temperature.

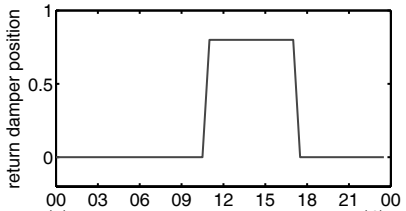
The performance of the proposed DMPC controller is reported in Figure 1.8. It cools down the room temperature to the lower bounds of the comfort range during the early morning (Figure 1.8(a)) while the baseline controller remains inactive until the room temperature hits the upper bounds around 4:00 (Figure 1.7(a)). This *precooling* saves energy, since during the early morning the lower ambient temperature enables free cooling. The free cooling is illustrated in Figure 1.8(d).



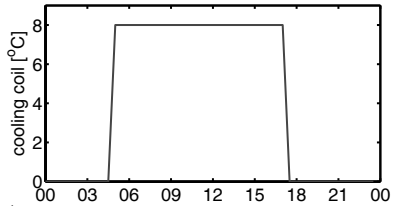
(a) Dashed lines are room temperatures ($T^{j\in\mathcal{V}}$), and solid lines define the comfort sets.



(b) Dashed lines are mass flow rate of the supply air to each zone ($\dot{m}_s^{j\in\mathcal{V}}$), and the solid line is the total mass flow rate $\dot{m}_{\text{fan}}$.

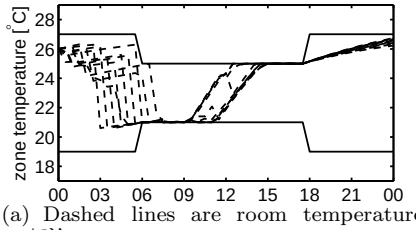


(c) Return air damper position (δ)

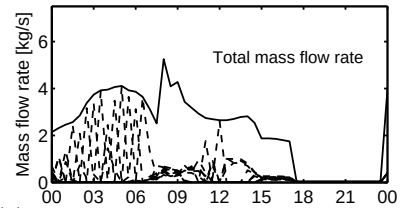


(d) Temperature difference across cooling coil (ΔT_h)

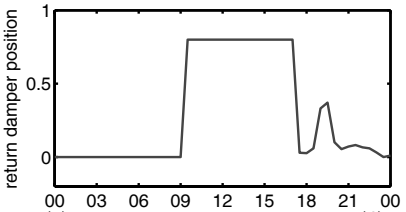
Figure 1.7. *System behavior for simplified baseline control logics*



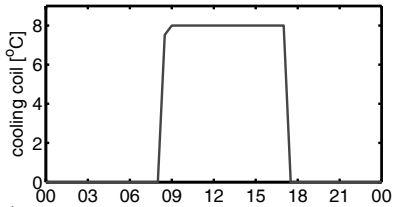
(a) Dashed lines are room temperatures ($T^{j\in\mathcal{V}}$), and solid lines define the comfort sets.



(b) Dashed lines are mass flow rate of the supply air to each zone ($\dot{m}_s^{j\in\mathcal{V}}$), and the solid line is the total mass flow rate $\dot{m}_{\text{fan}}$.



(c) Return air damper position (δ)



(d) Temperature difference across cooling coil (ΔT_h)

Figure 1.8. *System behavior for distributed model predictive control*

The MPC algorithm decides to open the cooling coil valve from 8:00, which is three hours later than the schedule proposed by baseline control logics in Figure 1.7(d).

Moreover, it is noted that instead of cooling all zones simultaneously, MPC cools down zones consecutively as Figure 1.8(a) illustrates. This feature significantly

reduces the peak total air flow rate from 7.2 kg/s of BC to 5.8 kg/s (Figure 1.8(b)), and thus saves fan energy consumption (note that we use a quadratic penalty of total supply air mass flow rate in (1.13a)). The simulation results suggested that the pre-cooling and consecutive cooling strategies induced by DMPC enable a 43.5% energy saving compared to the baseline.

DMPC Algorithm Complexity

The proposed DMPC Algorithm 1.1 can be implemented in a network of embedded processors with low computational capacity since Step 3 and Step 4 of Algorithm 1.1 require only a few algebraic operations and simple projections. Figure 1.9 shows that a large number of iterations is required. This imposes a requirement for high network communication speed.

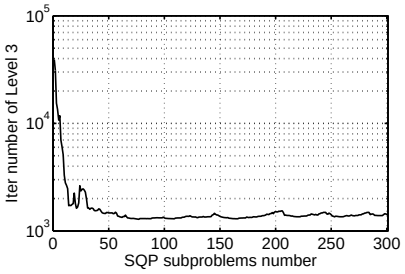


Figure 1.9. *Number of dual decomposition iterations at time $t = 0$.*

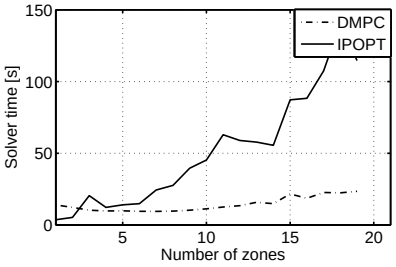


Figure 1.10. *Comparison between CPU time for DMPC and IPOPT.*

The DMPC Algorithm 1.1 was coded in Matlab® and runs on a single PC with Intel Core Duo CPU 3.00GHz. The runtime of the DMPC algorithm is estimated based on the assumption that the computation of Step 3 and Step 4 in Algorithm 1.1 are executed in parallel on $N_v + 1$ units including the controller units equipped on each VAV box and the AHU unit, and that the communication time is neglected. The results are reported in Figure 1.10 for different numbers of thermal zones considered. The dashed line shows the runtime of DMPC algorithm *when implemented on $N_v + 1$ CPUs in parallel*, and the solid line depicts the time required to solve Problem (1.13) by the Interior Point OPTimizer (IPOPT) interfaced via AMPL *on one CPU*. Both the DMPC algorithm and the IPOPT start from the same initial guess of optimal solutions. One can notice that when the number of zones is less than three, IPOPT is faster than DMPC on a single PC. As the number of zones and the size of problem (1.13) increase, one can notice that DMPC could be implemented with a faster control sampling rate than IPOPT.

Figure 1.11 plots the variations of the optimal cost $J^{\star}_{DMPC}$ obtained by Algorithm 1.1 relative to the optimal cost $J^{\star}_{IPOPT}$ by IPOPT when solving MPC problem (1.13) with different numbers of zones N_v. It is noted that DMPC results in slightly higher optimal costs than IPOPT. The reasons for this include the selection of convergence tolerance κ in Table 1.2.

Fast Gradient Method Improvement

The advantage of applying the fast gradient method instead of the projected subgradient method in level 3 of the DMPC scheme is illustrated in Figure 1.12. We focus on the MPC problem (1.13) at $t = 0$. The modified SQP algorithm converged in 301 iterations. The dashed line in Figure 1.12 depicts the total number of iterations $\sum_{i=1}^{n_p} n_d^i$ required to solve the subproblems generated from the SQP algorithm when the fast gradient method in Section 1.3.3 is applied. The solid line depicts the number of iterations for the projected subgradient method with constant step size.

In this work, n_d^i is the number of iterations to solve the i-th subproblem (1.16) obtained in the proximal minimization level. If the fast gradient method is used, the number of iterations required to satisfy the stopping criterion is, on average, about four times less than for the projected subgradient method.

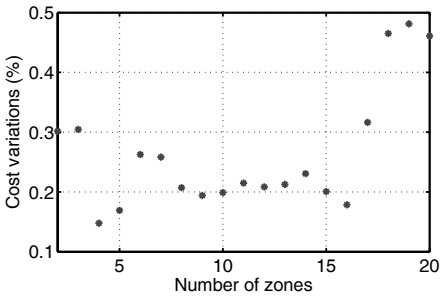


Figure 1.11. *Cost variations* $(J^\star_{DMPC} - J^\star_{IPOPT})/J^\star_{IPOPT}$.

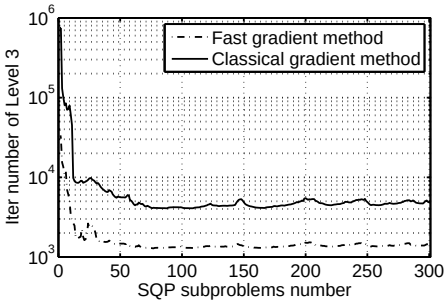


Figure 1.12. *Number of dual decomposition iterations at time* $t = 0$.

1.5 CONCLUSIONS

In this study a simplified two-mass room model is presented. Validation results show that the model captures the thermal dynamics of a thermal zone with negligible external load. Based on this model and predictive information of thermal loads and weather, a distributed model predictive control is designed to regulate thermal comfort while minimizing energy consumption. We have presented a three-level iterative algorithm for the solution of the nonlinear MPC problem. The key advance of this approach is the ability to solve the nonlinear MPC problem in a distributed manner and also in parallel. The resulting scheme is suitable for being implemented on a set of distributed low-cost processors. Simulation results show interesting behavior and short computation time. Future research will focus on analyzing stability issues of the resultant nonlinear MPC controller and improving the convergence rate of the proposed algorithm using adaptive step length and parameters.

Bibliography

[1] Y. Ma, G. Anderson, and F. Borrelli. A Distributed Predictive Control Approach to Building Temperature Regulation. In *2011 American Control Conference*, June 2011.

[2] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. *Automatic*, 36(6):789–814, June 2000.

[3] E. Camacho and C. Bordons. Nonlinear Model Predictive Control: An Introductory Review. In Rolf Findeisen, Frank Allgöwer, and Lorenz Biegler, editors, *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, pages 1–16. Springer Berlin / Heidelberg, 2007.

[4] J.H. Lee, M. Morari, and C.E. Garcia. State-Space Interpretation of Model Predictive Control. *IFAC*, 1992.

[5] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290. Springer-Verlag, 2003.

[6] F. Borrelli, T. Keviczky, and G.E. Stewart. Decentralized Constrained Optimal Control Approach to Distributed Paper Machine Control. In *44th IEEE Conference on Decision and Control and European Control Conference. CDC-ECC '05*, pages 3037 – 3042, December 2005.

[7] J.B. Rawlings and B.T. Stewart. Coordinating Multiple Optimization-Based Controllers: New Opportunities and Challenges. *Journal of Process Control*, 18(9):839 – 845, 2008.

[8] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, and G. Pannocchia. Cooperative Distributed Model Predictive Control. *Systems & Control Letters*, July 2010.

[9] F. Borrelli and T. Keviczky. Distributed LQR Design for Identical Dynamically Decoupled Systems. *Automatic Control, IEEE Transactions on*, 53(8):1901–1912, Septempber 2008.

[10] J. Nocedal and S. J. Wright. *Numerical Optimization*, chapter 18. Springer-Verlag, 1999.

[11] S.P. Han. A Globally Convergent Method for Nonlinear Programming. *Journal of Optimization Theory and Applications*, 22(3):297–309, July 1977.

[12] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*, volume 290. Springer-Verlag, Englewood Cliffs, NJ, 1989.

[13] L.S. Lasdon. Duality and Decomposition in Mathematical Programming. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2), July 1968.

[14] F. Oldewurtel, A. Parisio, C.N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy Efficient Building Climate Control using Stochastic Model Predictive Control and Weather Predictions. In *2010 American Control Conference*, pages 5100–5105, June 2010.

[15] G.P. Henze, C. Felsmann, and G. Knabe. Evaluation of optimal control for active and passive building thermal storage. *International Journal of Thermal Sciences*, 43(2):173 – 183, 2004.

[16] G.P. Henze, J. Pfafferott, S. Herkel, and C. Felsmann. Impact of Adaptive Comfort Criteria and Heat Waves on Optimal Building Thermal Mass Control. *Energy and Buildings*, 39(2):221 – 235, 2007.

[17] G.P. Henze, M. Krarti, and M.J. Brandemuehl. Guidelines for Improved Performance of Ice Storage Systems. *Energy and Buildings*, 35(2):111 – 127, 2003.

[18] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model Predictive Control for the Operation of Building Cooling Systems. In *2010 American Control Conference*, pages 5106 –5111, June 2010.

[19] J. Jang. *System Design and Dynamic Signature Identification for Intelligent Energy Management in Residential Buildings*. PhD thesis, University of California at Berkeley, 2008.

[20] F.E. Romie. Transient Response of the Counterflow Heat Exchanger. *Journal of Heat Transfer*, 106(3):620–626, 1984.

[21] R.J. Chillar and R.J. Liesen. Improvement of the ASHRAE Secondary HVAC Toolkit Simple Cooling Coil Model for Simulation. *Proceedings of the SimBuild 2004 Conference, Boulder, Colorado.*, August 2004.

[22] Y.W. Wang, W.J. Cai, Y.C. Soh, S.J. Li, L. Lu, and L. Xie. A Simplified Modeling of Cooling Coils for Control and Optimization of HVAC Systems. *Energy Conversion and Management*, 45(18-19):2915 – 2930, 2004.

[23] A. Rantzer. Dynamic Dual Decomposition for Distributed Control. In *American Control Conference, 2009*, pages 884 –888, June 2009.

[24] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[25] S. Richter, M. Morari, and C.N. Jones. Towards Computational Complexity Certification for Constrained MPC Based on Lagrange Relaxation and the Fast Gradient Method. In *Proc. 50th IEEE Conf. on Decision and Control*, December 2011.