

An internet of things upgrade for smart and scalable heating, ventilation and air-conditioning control in commercial buildings

Ethan Png^a, Seshadhri Srinivasan^{b,*}, Korkut Bekiroglu^c, Jiang Chaoyang^d, Rong Su^a, Kameshwar Poolla^e

^a School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore

^b Berkeley Education Alliance for Research in Singapore (BEARS), Singapore 138609, Singapore

^c Electrical Engineering Technology, SUNY Polytechnic Institute, Utica, NY, USA

^d School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

^e University of California at Berkeley, 5105 Etcheverry Hall, USA

HIGHLIGHTS

- A scalable and smart HVAC control approach.
- An IoT architecture and prototype for implementing the control.
- Deployment results in a test-bed.
- Implementation aspects with IoT and legacy BMS.

ARTICLE INFO

Keywords:

Internet of Things (IoT)
Smart Token Based Scheduling Algorithm (Smart-TBSA)
Heating, Ventilation and Air-Conditioning (HVAC)
Building Automation System (BAS)

ABSTRACT

Scalability of control algorithms used for savings energy in commercial building Heating, Ventilation and Air-Conditioning (HVAC) system and their implementation on low cost resource constrained hardware is a challenging problem. This paper presents the Internet of Things (IoT) prototype which implements a smart and scalable control approach called the *Smart-Token Based Scheduling Algorithm* (Smart-TBSA) to minimize energy in commercial building HVAC systems. The IoT prototype is formalized with an architecture that encapsulates the different components (hardware, software, networking, and their integration) along with their interactions. A detailed description of the different components, hardware design, deployment issues, and their integration with legacy systems as well as cloud-connectivity is presented. In addition, simple modifications required for transforming the optimization models to an active control technique is also presented. While scalability is provided by the decentralized control, recursive zone thermal model identification, prediction occupant's thermal sensation, and embedding them within the optimization models enhances the smartness. Consequently, due to the implementation of Smart-TBSA using IoT devices, an otherwise centralized control architecture of the legacy building automation system is transformed to a more scalable and smart decentralized one. The proposed Smart-TBSA and IoT prototype are illustrated on a pilot building in Nanyang Technological University, Singapore having 85 zones. Our results shows that by combining IoT with decentralized control, energy savings up to 20% can be derived. Moreover, we show that legacy building automation system can be transformed into a more smart, adaptable, scalable, and decentralized control by deploying IoT devices without incurring significant costs.

1. Introduction

Building energy consumption accounts for about 51% of the total energy in Singapore and space cooling contributes to a significant

portion of it [1,2]. Therefore, significant attention has been devoted to conserve energy in heating, ventilation, and air-conditioning (HVAC) systems using information (e.g., weather, occupancy etc.). This requires approaches to integrate such information into building thermal models

* Corresponding author.

E-mail addresses: seshadhri.srinivasan@bears-berkeley.sg (S. Srinivasan), cjiang@bit.edu.cn (J. Chaoyang), rsu@ntu.edu.sg (R. Su), poolla@berkeley.edu (K. Poolla).

Nomenclature	
Variable	
A_i	duct cross-sectional area for zone i valve opening (m^2)
Ar_i	floor area of zone i (m^2)
c_i	thermal capacitance for zone i [$\text{kJ} \cdot \text{C}^{-1}$]
c_p	specific heat of air [$\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]
A_i, \bar{A}_i	minimum and Maximum cross sectional area of duct [m^2]
d_r	return to total air ratio
E_v	system ventilation efficiency
g_i	cooling energy supply to zone i [kJ]
H_p	prediction horizon in time epochs
i	zone index
k	sample index
k_f	parameter capturing fan efficiency and duct pressure losses
\dot{m}_i	cool air mass flow rate into zone i [kg s^{-1}]
\dot{m}_{il}	lower limit on air mass flow rate [kg s^{-1}]
\dot{m}_{ih}	upper limit on air mass flow rate [kg s^{-1}]
\dot{m}_{OA}	mass flow rate of outside air [kg s^{-1}]
\dot{m}_{SA}	mass flow rate of fan supply air [kg s^{-1}]
n_z	number of zones
p_0	pressure at supply fan i [Pa]
p_i	pressure at entrance of zone i [Pa]
P_c	chiller power consumption [kJ]
P_f	fan power consumption [kJ]
P_i	population of zone i
\dot{Q}	cooling load forecast for zone i [kJ]
Q_{ch}	constant for various amounts of cooling loads
R_i	thermal resistance between zone i and the environment [kW K^{-1}]
T_c	temperature of cool air supply [$^\circ\text{C}$]
T_i	temperature of zone i [$^\circ\text{C}$]
T_{il}	lower thermal comfort bound [$^\circ\text{C}$]
T_{ih}	upper thermal comfort bound [$^\circ\text{C}$]
T_{oa}	temperature of outside air [$^\circ\text{C}$]
Zp_i	primary outdoor air fraction for zone i
δ	sampling time [s]
η	reciprocal of chiller COP

and perform optimization to reduce energy consumption. In this backdrop, model predictive control (MPC) is being widely used as the control approach due to its ability to include information and perform optimization. This is a complex task due to distributed nature of HVAC devices and nonlinear relationships between flow and pressure. In addition, in commercial buildings, different types of HVAC components are deployed, e.g., the chiller, air-handling units (AHU), variable-air-volume (VAV) boxes, pumps and others. To handle the heterogeneity and interoperability issues with these devices, usually a building automation system (BAS) is used. Albeit their ability to integrate different technologies, getting a payback from BAS has been a daunting task and a major concern for automation industries, service providers, and building owners. In addition, functionally, BAS are less flexible due to their centralized architecture and proprietary communication protocols. Further, they do not have many intelligent designs or technologies that support decentralized control, smartness, and energy optimization. Consequently, building owners are looking at cost-efficient upgrades to BAS that can also save energy and provide enhanced smartness.

In this scenario, it is meaningful to use emerging technologies such as the “Internet of Things” (IoT) to transform a legacy BAS into a more flexible, smart and energy efficient system. By exploiting the low-power, open networking, and compact hardware, the centralized BAS control architecture can be transformed into a more flexible and smart architecture to save energy in buildings. This helps in overcoming interoperability constraints, ubiquitous sensing through low power sensors, and greater data visualization. Still, the IoT’s capability need to be complimented by control approaches that can transform the centralized architecture to a more scalable, smart, and energy efficient application. However, there are not many studies that have investigated the role of IoT as an upgrade to legacy BAS to provide smart and scalable HVAC control.

The available literature can be discerned into two broad categories: (i) the use of IoT for building control, and (ii) the HVAC control approaches. D. Minoli et al. [3] studied the technical challenges and opportunities provided by IoT integration in commercial buildings were. Further, they argued that the IoT offers strong business opportunities in energy savings, demand response and cost savings for tenants and building owners. Further, the technical challenges such as lack of reliability, standardization and cloud integration were discussed. Coates et al. [4] presented the challenges in monitoring buildings using IoT. Li et al. [5] studied the role of IoT for creating energy awareness among residents in smart buildings. Al-Ali et al. [6] proposed a smart home energy management system using IoT and big-data analytics. Zhang

et al. [7] presented the role of IoT devices in making buildings smarter by deploying additional sensors. Pan et al. [8] provided an IoT framework for smart buildings with discussions on design prototype, and experiments. However, the investigation does not provide detailed insights into controller realization, rather uses heuristics to optimize energy. Consequently, there are no guarantees on optimal performance and energy savings. Similarly, Javed et al. [9] used a random neural network based smart controller for building HVAC system. Albeit, being a decentralized approach it lacks discussions on power consumption in HVAC components such as the air handling unit (AHU) or chiller. Therefore, adopting such results to large commercial buildings is a challenging task. Serra et al. [10] studied implementing HVAC control by deploying IoT devices in residential buildings with a simple implementation. Ruano et al. [11] studied implementing MPC using wireless sensors and IoT platform to implement a intelligent HVAC control. Kastner et al. [12] studied the integration of IoT and IPv6 for building automation systems. Tushar et al. [13] proposed an IoT architecture and methodology for green building energy management. Korkut et al. [14] proposed an approach to estimate thermal model of commercial buildings with packet-losses in the IoT communication channels. Adhikari et al. [15] proposed an algorithm to reduce aggregate energy consumption of HVAC systems by modelling it as a job-shop scheduling problem.

As for the HVAC control, centralized architecture based MPC has been designed in (see, [16,17] and references therein). The centralized approaches solve a complex multi-time step nonlinear optimization model that requires specialized hardware and implementing them with existing BAS is costly [18]. Moreover, the control architecture is quite complex as sensor measurements from each zone has to be transmitted to the central controller and actuators have to be performed from the centralized control. Consequently, they are costly and less scalable. Decentralized approaches optimizing fan power consumption have been proposed in the literature (see, [19,20] and references therein) which overcome the shortcomings of the centralized control. However, the chiller energy consumption has not been considered in these methods [21]. Distributed approaches using dual-decomposition [22], affine quadratic regulator [23], affine disturbance feedback method [24], projected subgradient method [25], and stochastic scenario based [26,27] have been studied in the literature. However, their energy optimization capability is diminished as they consider only AHU fan power and chiller power consumption is not modelled. Furthermore, scalability of the methods to large scale buildings and their implementation aspects have not been studied. These investigation have

not studied the implementation aspects of the MPC and are more conceptual analysis. To overcome the shortcomings of existing control approaches, a hierarchical control architecture that is scalable due to its distributed computations and includes chiller energy consumption in its formulation called the *Token Based Scheduling Algorithm* (TBSA) was proposed by the authors in [1,28]. In TBSA, distributed cooling demands from the different zones in the buildings are placed as service requests to a central scheduler. The cooling requests are used as the baseline by the central scheduler which provisions the cooling energy based on AHU and chiller constraints.

A review of the literature reveals that the TBSA provides a scalable control approach to optimize energy in commercial building HVAC systems. To our best knowledge, a complete implementation of the MPC approach for HVAC system with scalability and smartness with IoT-based implementation has not been studied in the literature. The objective of this paper is to use IoT to implement Smart TBSA (S-TBSA) as an upgrade to existing BAS without having to replace or refurbish existing ones. Consequently, legacy BAS can be upgraded to have energy and operational efficiency. In addition desirable features such as smartness, distributed control, scalability, adaptability, data-aggregation, and data-visibility is provided. To enhance the TBSA, first a recursive model identification approach is used to identify zone thermal models during each control epoch which provides adaptability. Second, a methodology combining sensor measurements and single-layer feed-forward neural network (SLFN) to determine occupant comfort as thermal sensation is presented. Finally, soft-constraints are introduced to guarantee feasibility and provide numerical stability is proposed. Next, the implementation aspects of the proposed Smart-TBSA is illustrated. This includes the description of an IoT prototype and architecture, networking and implementation aspects as well. The IoT prototype provides interoperability, data-visualization, and monitoring capability to the system. We add the following additional contributions to our earlier papers [1,28]:

1. Following methodological contributions are added to our previous work:
 - (a) Recursive identification of zone thermal model
 - (b) Occupant thermal comfort modelled as thermal sensation
 - (c) Soft-constraints to guarantee feasibility of the zone controller.
2. A novel IoT prototype and architecture that implements (hardware, software, and integration) the smart-TBSA and upgrades the existing

BAS are presented.

3. Finally, the deployment results of the IoT prototype and smart-TBSA including the realization aspects of the TBSA, model update, TBSA performance in terms of energy savings and computation times, deployment cost, and discussions on payback times are presented.

The paper is organized as follows. The preliminaries and the token based scheduling strategy are discussed in Section 2. The IoT architecture including the hardware, software and their integration with existing BAS is discussed in Section 3. In addition, the models for HVAC components and the optimization model for the TBSA is also presented. The results of the IoT prototype deployment on a pilot building is provided in Section 4. Conclusions are drawn from our results in Section 5.

2. Preliminaries

The main focus of the paper is to propose a Smart-TBSA which enhances the current capabilities of the TBSA and then show the IoT-based implementation of the decentralized control. This section presents the preliminaries required for understanding the methodology and results presented in the paper. This section provides the nomenclature, brief description of HVAC system, and an introduction to TBSA.

2.1. HVAC system description

The schematic of the variable air volume (VAV) controlled HVAC system supplying cooling energy to different zones in a commercial building is shown in Fig. 1. It consists of an air-handling unit (AHU) which receives air from two sources: the return air from the zones that are ventilated by the AHU and fresh air infused from the outside depending on the CO₂ levels in the zones. The mixed air is passed through the filters and then through the cooling coils, which is a heat exchanger that has cooled water supplied from the chiller. Consequently, the warm air rejects heat to the chilled water in the cooling coils and the AHU fan circulates the cooled air into the zones through the building's duct network. Dampers are fitted in the ducts to adjust the air flow by changing the cross-sectional areas affecting the flow through each zone using the VAV [29]. This air cools the zone and a fraction of this air is vented out, while the rest of the air is fed into mixing chamber.

The HVAC system is controlled using the BAS which implements the

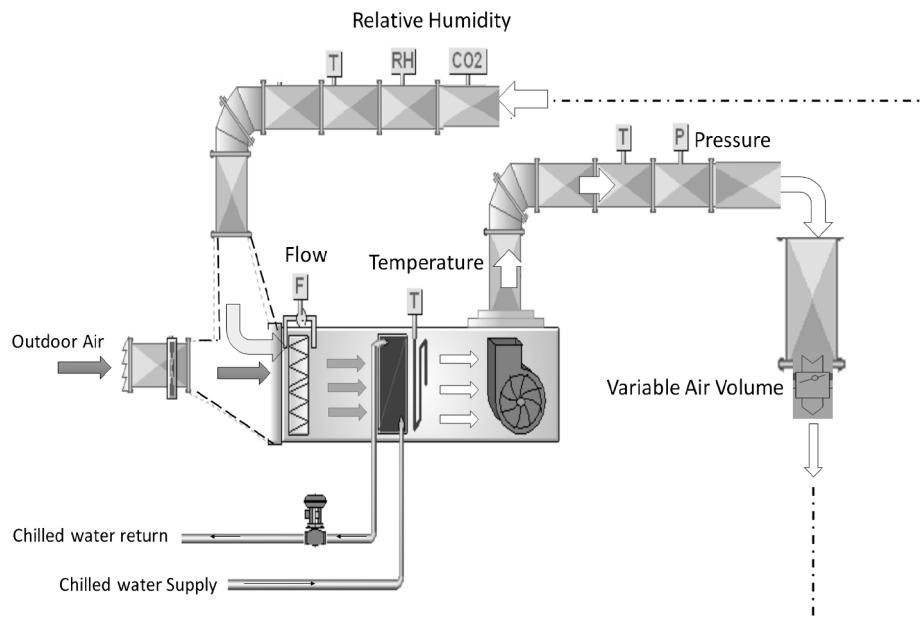


Fig. 1. Schematic of VAV based HVAC control.

AHU and chiller controls. The AHU has three closed-loop control: pressure, temperature, and fresh air injection. In addition each zone has the VAV which is controlled to adjust the temperature. Whenever, the cooling demand increases/decreases in a zone, the VAV is opened/closed to change the mass-flow rate of air thereby maintaining the zone temperature within user-defined comfort margins. The opening/closing of VAV causes pressure differences in the AHU duct which is again brought back to the pre-set value by controlling the AHU fan through a variable frequency drive. The AHU temperature also varies due to changes in thermal load, to maintain a constant flow, the chiller flow is commanded through an automated valve. The fresh air injection is controlled using a closed loop that logs on the CO₂ levels in the zones by opening/closing the fresh air damper. Similarly, the chiller's pump can be commanded to change the cooling in the AHUs. Since, one chiller supplies cooled water to many AHUs the control actions are less frequent.

2.2. Token Based Scheduling Algorithm

The objective of Token Based Scheduling Algorithm (TBSA) is to provide a scalable and decentralized control architecture for reducing the energy consumption in both AHU and chiller without compromising human comfort. In the TBSA, the cooling is considered as a service requested by different users (zones) as *Tokens*. The central controller receives these *Token Requests* from various zones and solves a scheduling problem considering underlying operation and physical constraints of AHU and chiller. Then the cooling energy is allocated to each zone, called the *Token Allocation* which is implemented in a decentralized manner in each zone by using their VAV control. The TBSA provides good scalability due to decentralized computation and, models

both fan and chiller power in its formulation as against existing decentralized approaches. The TBSA was proposed by the authors in their previous works [28]. While the algorithm showed energy savings and scalability, it lacked adaptability and had limited smartness. Moreover, implementation aspects of the control approach has not been proposed. Implementing TBSA with legacy BAS is a challenging task due to interoperability issues among communication protocols, need to deploy sensors (IoT sensors are cheap and compact), requirement for costly control hardware, and traditional control architecture which is centralized does not favour decentralized control required for TBSA.

In what follows first, we present a Smart-TBSA which is an enhancement to our previous contributions with the following elements:

- (i) Recursive identification of zone thermal models to have adaptability and agility towards varying ambient conditions in the zone.
- (ii) Occupant comfort modelled as thermal sensation which is predicted from sensor measurements and single-layer feed-forward neural network which provides smartness to the system.
- (iii) soft-constraints that guarantee convergence and ability for the optimization models to be solved in on-board computers.

Second, we propose an IoT prototype for realizing Smart-TBSA. This includes hardware, software and integration aspects including their interface with traditional BAS. Detailed discussions on the sensor selection, actuation, and communication protocols of the different modules used for control is presented. Communication interfaces which includes IoT gateway and the ability to provide interoperability using IoT is also illustrated. In addition, we also show the modifications to be performed to convert the Smart-TBSA from being an abstract optimization routine to an active control strategy is proposed.

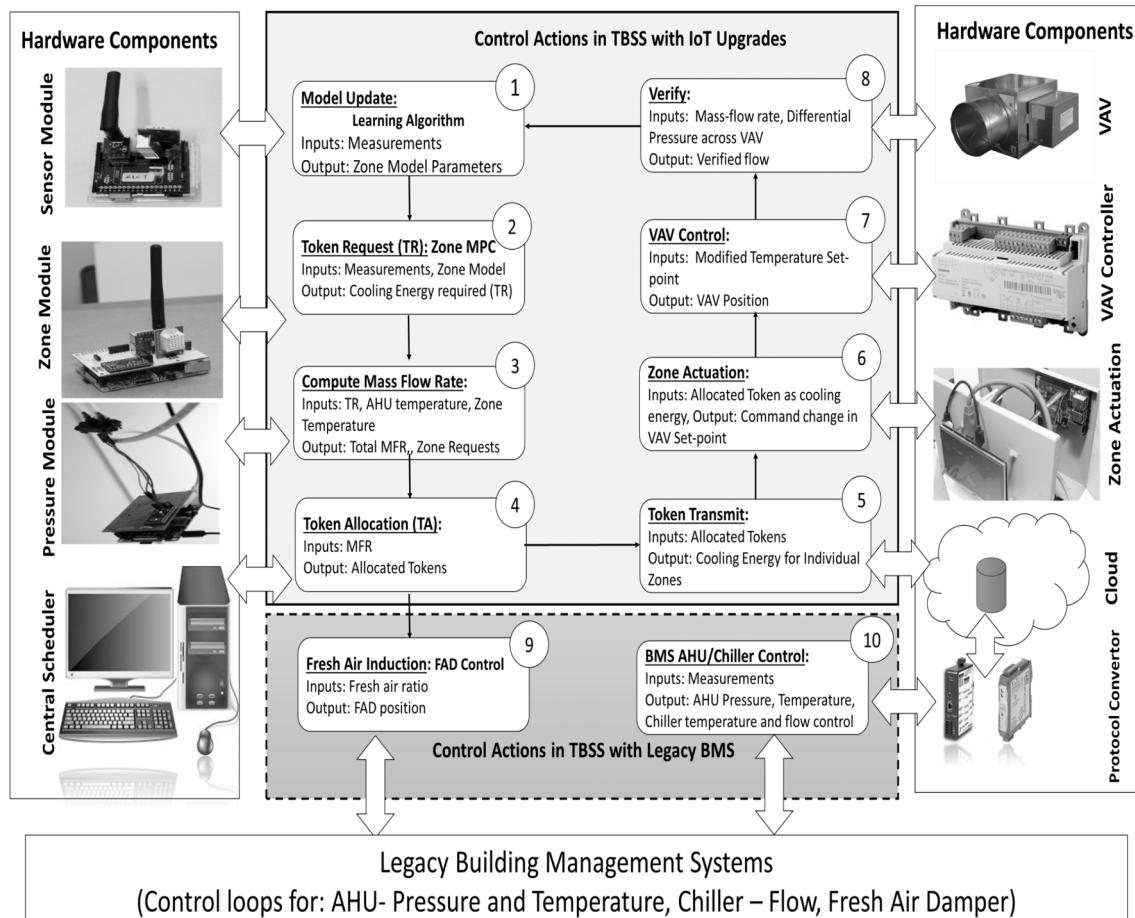


Fig. 2. Token Scheduling Framework for HVAC Scheduling.

Finally, deployment results of the proposed Smart-TBSA on a test-bed and their performance in terms of energy savings, adaptability, and smart detection of occupant's comfort are presented. We use the International Performance Measurement and Verification Protocol (IPMVP) to study the energy savings.

3. Smart-Token Based Scheduling Algorithm

In this section, the mathematical formulation of Smart-TBSA with the HVAC components is first provided as a background to understand the implementation. Then the IoT architecture along with the hardware, software, and integration aspects are discussed after presenting the control approach.

The Smart-TBSA and associated components are shown in Fig. 2. Both IoT hardware and the control approach are described. First, we describe the different steps.

The different steps are:

3.1. Recursive model adaptation

Consider a building with zone thermal dynamics given by [1]:

$$T_i(k+1) = \alpha_i^1(k)T_i(k) + \alpha_i^2(k)g_i(k) + \alpha_i^3(k)T_{oa}(k) + \alpha_i^4(k)Q_i(k) + \varepsilon(k) \quad (1)$$

where T_i is the temperature of zone i , g_i is cooling energy supplied to zone i , \dot{Q} represents the load forecast due to thermal input from internal loads, T_{oa} is ambient temperature, and ε is assumed to be a normally distributed bounded disturbance with $\|\varepsilon(k)\|_2 \leq \eta$ with $\eta \in \mathbb{R}^+$. Here k is the sample index, and i is the zone index with $i = 1, \dots, N_z$.

Remark 1. Even though normally distributed Gaussian has an unbounded support, we assume a bound on the 2-norm of the noise ($\|\varepsilon(k)\|_2 \leq \eta$) to choose a set of high probability. More discussions on this assumption can be found in [30]

From (1), one can see that the zone thermal model is influenced by various factors such as current room temperature, heating load, occupancy, weather conditions, and other factors that are intrinsic and extrinsic to the building. As these factors are nonlinear and time-varying modelling their influence on building thermal dynamics is essential for energy optimization. The Smart-TBSA uses sensor modules that provides measurements of temperature, CO₂ and humidity at each zone. In addition, forecasts on disturbances such as weather are obtained using either web or cloud interfaces. The measurement thus obtained are transmitted to the *Zone Modules* (described later) using wireless communication channels. The model is updated in Zone Modules based on the most recent measurements and forecasts using a recursive model identification approach. Thus, the model parameter estimation problem for each zone i can be written as:

$$T_i(k+1) = \mathbf{H}_i(k)x_i(k) + \varepsilon$$

where

$$\mathbf{x}_i(k) = [\alpha_i^1(k) \ \alpha_i^2(k) \ \alpha_i^3(k) \ \alpha_i^4(k)]^T$$

$$\mathbf{H}_i(k) = [T_i(k) \ g_i(k) \ T_{oa}(k) \ \dot{Q}_i(k)].$$

In the recursive identification problem, the optimization problem \mathcal{P}_1 is solved during each time-instant k to identify the model parameters with a forgetting factor λ_i , i.e.,

$$(\mathcal{P}_1): \min_{x_i(k)} \sum_{k=0}^{N_p} \lambda^{N_p-k} \left[(T_i(k) - \mathbf{H}_i(k)x_i(k))^T \Pi_i(k) (T_i(k) - \mathbf{H}_i(k)x_i(k)) \right]$$

$$\text{s. t.} \quad \Pi_i(k) \in \mathbb{R}^+$$

The different steps of the model update algorithm is presented in **Algorithm 1** and it provides the recursive updates of the parameters for

each zone i at time epoch k [31].

Algorithm 1. Adaptive Model Parameter Estimation

- 1: **Inputs:** $P \in \mathbb{R}^{n \times n}$ Inverse Correlation Matrix and $\lambda \leftarrow$ Forgetting Factor
- 2: **Measurement:** $T_i(k+1)$, $T_i(k)$, $g_i(k)$, $T_{oa}(k)$, and $\dot{Q}_i(k)$
- 3: **Estimation:** $\leftarrow \tilde{T}_i(k+1) = \mathbf{H}_i(k)x_i(k)$
- 4: **Estimation Error:** $e(k) = \tilde{T}_i(k+1) - T_i(k+1)$
- 5: $\varphi = \mathbf{H}_i(k) \times P$
- 6: **Gain vector update:** $\Gamma = \varphi / (\lambda + \varphi \times H^T)$
- 7: **Coefficient Adaptation:** $x_i(k+1) = x_i(k) + \Gamma \times e(k)$
- 8: **Inverse Correlation Matrix Update:** $P = (P - \Gamma \times \varphi) / \lambda$
- 9: **Output:** $x_i(k)$

Remark 2. To speed up the control, the model parameters were updated only when the model error breached some threshold value.

Remark 3. The recursive model identification in our case consists of 4 variables and implementing RLS requires matrix computation which was feasible to be implemented in resource constrained hardware. In addition, we initialized the model parameters with nominal parameter values obtained from off-line least square measurements and used a low forgetting factor during implementation in IoT hardware. The scalability needs to be investigated when the number of model parameters and the forgetting factor increases.

3.2. Computing occupant comfort limits

To account for the thermal comfort of the user, we need to build a thermal comfort model which determines the temperature set-points in each zone. Thermal sensation though abstract provide good insights into occupant comfort than detection as they consider underlying ambient conditions and are quantized as shown in Fig. 3. We model the thermal comfort as

$$S_{TC} = f_{TC}(T, V, H, T_{MRT}, I_C, I_M)$$

where S_{TC} represents the thermal sensation scale. For the indoor environment, the human thermal sensation mainly depends on six parameters: air temperature, air speed, relative humidity, mean radiant temperature, clothing insulation, and metabolic rate [32–34], which are denoted by T , V , H , T_{MRT} , I_C , and I_M , respectively. Some other excellent thermal comfort model can be found in [33,34]. In this work, we selected the widely used six-parameters-based model.

Since thermal sensation is an abstract quantity that can be interpreted from quantities which have complex behaviours, we use the universal approximation ability of Single-hidden Layer Feedforward Neural network (SLFN), we use a SLFN to fit this thermal comfort model. The network structure of a SLFN is shown in Fig. 4. Given an input $\mathbf{x} = [x_1, x_2, \dots, x_6] = [T, V, H, T_{MRT}, I_C, I_M]$, the output of the hidden neuron j , denoted as o_j , can be expressed as

$$o_j = f_a \left(\sum_{i=1}^6 a_{ij} x_i + b_j \right)$$

where $f_a(\cdot)$ is the activation function such as Sigmoid function, a_{ij} is the weight from input neuron i to the hidden neuron j , and b_j is the bias of hidden neuron j . The final output of the SLFN is

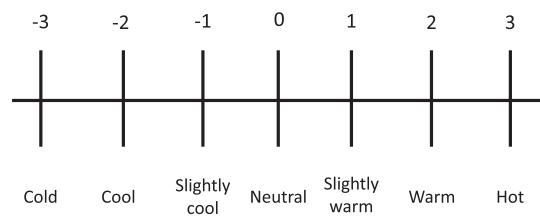


Fig. 3. ASHRAE thermal sensation scale.

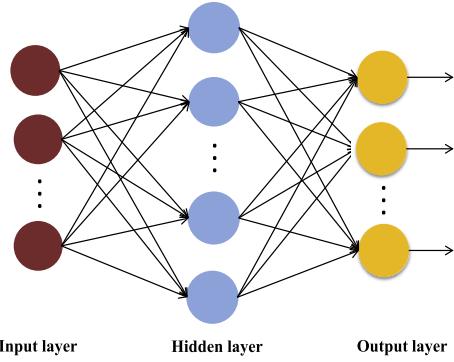


Fig. 4. Single-hidden layer feedforward neural network (SLFN) architecture for thermal sensation.

$$S_{TC} = \arg \max_s f_S \left(\sum_{j=1}^L c_{js} o_j + d_s \right) \quad (2)$$

where c_{js} is the weight from hidden neuron j to output neuron s , d_s is the bias of the output neuron s . L is the number of hidden neurons, $f_S(\cdot)$ is a linear function or a Sigmoid function, and $s \in \{-3, -2, -1, 0, 1, 2, 3\}$ represents the thermal sensation scale. We use the back propagation [35] and Levenberg-Marquardt algorithm [36] techniques to train this SLFN, i.e., to determine the weights and biases of this SLFN. In the training process, we minimize the mean-squared-error of the output of the SLFN. In practice, we can directly solve this SLFN by the MATLAB Neural Network Toolbox (Neural Net Fitting).

With SLFN, we can directly obtain the real-time thermal sensation index of the indoor environmental given the six parameters. Obviously, the air temperature is positively correlated with the thermal sensation scale. Thus, given the other five parameters, there exist a lower bound and an upper bound such that if the air temperature is between the two bounds, the indoor environment is neutral and the users feel comfort. Therefore, user-defined comfort limits are modelled from the parameters using (2) by gradually reducing or increasing the air temperature from the current one until S_{TC} becomes -1 and 1 , respectively.

3.3. Token requests

The zone modules receive the sensor measurements and compute the cooling energy required to meet the user defined comfort margins computed from the thermal sensation model. We denote the minimum cooling energy as *Token* which is computed by solving a MPC, whose aim is to minimize the cooling energy subjected to user defined comfort margins (thermostat set-points) and operating constraints expressed as cooling energy bounds over a predefined time-horizon. The cooling energy computed by MPC is called *Token Requests* of each zone and these computations are performed in parallel. Unlike our previous contributions where the comfort bound were provided by the users, in Smart-TBSA they are determined from thermal sensation. To raise the *Token Request*, each *Zone Module* solves the following optimization problem,

$$\begin{aligned} (\mathcal{P}_2): J_{i,i}(H_p) &= \min \sum_{k=1}^{H_p} g_i(k) + C_i(k)\gamma_i(k) \\ \text{s. t.} \\ T_i(k+1) &= \alpha_1^i T_i(k) + \alpha_2^i g_i(k) + \alpha_3^i T_{od}(k) + \alpha_4^i(k)Q_i(k) + \varepsilon(k) \\ T_{il}(k) - \gamma_i(k) &\leq T_i(k) \leq T_{ih}(k) + \gamma_i(k) \\ g_{il}(k) &\leq g_i(k) \leq g_{ih}(k) \\ \gamma_i(k) &\geq 0 \\ \forall k: 1 \leq k \leq H_p, \forall i \in N_z \end{aligned}$$

where $\gamma_i(k)$ is a slack variable which serves as a soft-constraint to relax the set-point bounds and to provide feasibility to the optimization

models when solved using single-board computers and $C_i(k)$ is the cost for soft-constraints which is usually a very high-value.

Remark 4. The soft-constraints provide feasibility in cases where the algorithm starts with an initial temperature wherein even applying the maximum cooling to the zone does not bring the temperature down to the feasible region. Further, by allowing small deviation that is usually not sensitive to the user it also enhances energy savings.

3.3.1. Compute mass-flow rates

The cooling requests $g_i(k)$ are converted to mass-flow rates using simple computations that depends on the zone and AHU temperature. The cooling requests as mass flow rates from different zones are aggregated and transmitted to the central scheduler. For a fixed planning horizon H_w , let $g_i^{\text{opt}}(k)$ in kJ s^{-1} be the optimal cooling request from the zone modules, then the zone mass-flow rates can be deduced as

$$\dot{m}_i^{\text{opt}}(k) = \frac{g_i^{\text{opt}}(k)}{c_p(T_i^{\text{opt}}(k) - T_c)}, \quad k = 1, \dots, H_w \quad (3)$$

and the corresponding minimum total mass flow of cold air on the planning horizon required for zone i is computed as:

$$TokB_i(H_w) = \sum_{k=1}^{H_w} \dot{m}_i^{\text{opt}}(k), \quad H_w = 1, \dots, W.$$

The $TokB_i(H_w)$ denotes the *minimum number of tokens* needed by zone i on the planning horizon H_w to meet its local temperature constraints. It carries information about the amount of cooling required.

3.3.2. Token allocation

The *Central Scheduler* solves the HVAC energy optimization problem (AHU plus chiller power) as a token scheduling problem considering the underlying constraints imposed by the fan and chiller capacity, pressure distribution in the duct, and other operating constraints. The token allocation for a time-window H_w is performed solving the optimization model (\mathcal{P}_3) given by,

$$\begin{aligned} (\mathcal{P}_3): J_3(H_w) &:= \min_{\dot{m}_i, p_i} \sum_{k=0}^{H_w} \left(\sum_{i=1}^{n_z} \dot{m}_i \right)^2 \\ \text{s. t.} \\ \sum_{k=0}^r \dot{m}_i(k) &\geq TokB(r) \quad \left(\forall r: 0 \leq r \leq H_w \right) \\ p_{i+1} - p_i + f \left(\sum_{q=i+1}^{n_z} \dot{m}_q \right)^2 &\leq 0, \quad i = 0, 1, 2, \dots, n_z - 1 \\ \dot{m}_i^2 &\geq \frac{A_i^{5/2}}{a} \left[p_i - p_{cap} \right], \quad \dot{m}_i^2 \leq \frac{\overline{A_i}^{5/2}}{a} \left[p_i - p_{cap} \right] \\ p_0(k) &\leq p_{cap}. \end{aligned}$$

where p_{cap} is the fan pressure rating and $f()$ denotes the function which approximates the fan power using mass-flow rates of the zones. The above problem is Quadratic Constrained Quadratic Programming (QCQP) and is solved using commercial solvers in the central scheduler.

3.4. Power consumption models

The AHU fan power consumption usually varies as a quadratic function of the mass-flow rate and is given by [37]:

$$P_f = k_f \left(\sum_{i=1}^{n_z} \dot{m}_i \right)^2 \quad (4)$$

Similarly, the chiller power consumption is modelled using a control-oriented model proposed in [37]:

$$P_c = c_p \eta \sum_{i=1}^{n_z} \dot{m}_i \left(T_m - T_c \right) \quad (5)$$

where the mixed air temperature is $T_m = (1 - d_r)T_{oa} + d_r T_r$ and the return air temperature is $T_r = \sum_{i=1}^{n_z} \dot{m}_i T_i / \sum_{i=1}^{n_z} \dot{m}_i$, d_r is the ratio of amount of return air to total air flows. Therefore, the power consumption of the chiller is given below:

$$P_c = c_p \eta \left(\left(1 - d_r \right) \left(T_{oa} - T_c \right) \sum_{i=1}^{n_z} \dot{m}_i + d_r \sum_{i=1}^{n_z} \dot{m}_i \left(T_i - T_c \right) \right) \quad (6)$$

3.4.1. Transmit tokens

The allocated tokens are in terms of mass-flow rate are converted back to cooling energy using services from the cloud that store the zone and AHU temperature. The cooling energy allocated for the zones is transmitted to the *Zone Modules* from the cloud using Ethernet based communication.

3.4.2. Zone actuation

The *Zone Module* then computes the inputs for the *Zone Actuation* and varies the mass-flow rate by commanding VAV by controlling its set-point which is proportional to the increase/decrease in mass-flow rate requested by the zones. This computation is performed in the *Zone Actuation Module*.

3.4.3. VAV control

The set-point change is used by the VAV controller, a Proportional Integral (PI) controller that commands the VAV actuator (an AC stepper motor) with both direction and actuation time proportional to the damper opening/closing requested by the set-point.

3.4.4. Verification

The change in mass-flow rate is reflected in the VAV box is read by using differential pressure sensor mounted in the *Pressure Sensor Module*. This is done to verify the execution of the token allocation.

Remark 5. The legacy BAS is still responsible for maintaining pressure, temperature, fresh air-damper, chiller flow control in the AHU.

Similarly, the chiller controller is responsible for pumping controls of the chiller to maintain constant temperature in the system.

Remark 6. The TBSA is realized by deploying three hardware devices *Sensor Module*, *Zone Module* and *Zone Actuator* in the zone and by deploying a central scheduler (cloud) to implement the *Token Allocation*, web interface, data-base for the control system and for integrating third party applications.

4. An IoT architecture and prototype

The IoT architecture used to implement Smart-TBSA is shown in Fig. 5. It has three tiers: field level, control, and cloud. The field tier consists of the legacy BAS and IoT layer. There are numerous *Sensor Modules* which are placed in each room of a zone (three rooms constitute a zone) and they aggregate measurements on temperature, humidity, and CO₂. The measurements are transmitted to the other IoT devices and gateway using wireless network implemented using nRF24L01 which is a single chip 2.4 GHz transceiver and has been selected due to its ultra low power capabilities. These measurements are transmitted to the *Zone Module* in each zone which in-turn uses weather forecasts from web server or clouds along with the measurements to implement the MPC which computes the minimum cooling energy required for each zone using the zone thermal model. In implementation, the *Zone Module* solves an optimization problem modelled as linear program. The computation results are the cooling rates from which mass-flow rates can be computed using simple mathematical equations which are transmitted to the IoT gateway via an Ethernet interface. In addition, the IoT layer provides Ethernet based communication such as Message Queue Telemetry Transport (MQTT) and Advanced Messaging and Queuing Protocol (AMQP) at the application level. They account for the IP extensions of the IoT layer. Cloud extensions are services provided by the IoT layer for interfacing IoT components and BAS to the cloud through IoT gateway. The web-services are applications used for interfacing cloud or other components of the system with IoT devices.

The legacy control layer serves three purposes: monitoring AHU and chiller control loops, consumption monitoring, and local controls for AHU and chiller. In addition, protocol converters are used to transmit

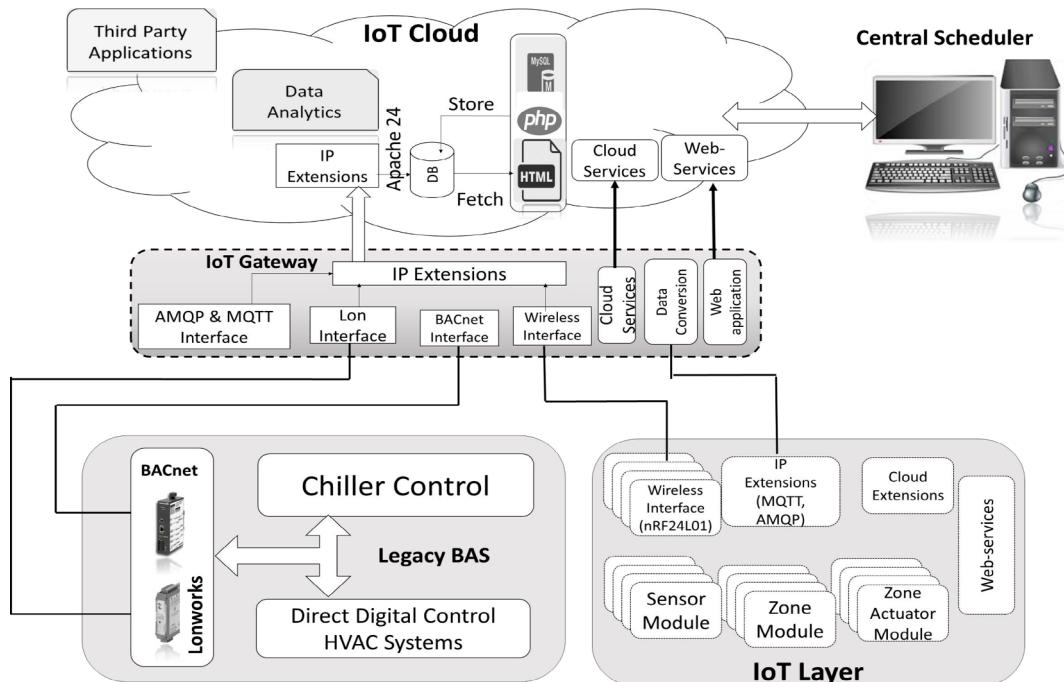


Fig. 5. IoT Architecture for Decentralized Control in Buildings.

data to the IoT gateway from the legacy systems. Communication interface for BACnet¹ and Lonworks² are supported in our implementation [38]. But the interfaces to other protocols can be extended without loss of generality. As such the legacy BAS layer is not flexible in its architecture and therefore no further extension of it is possible due to interoperability issues. The BACnet and Lonworks protocol converters store the data using a web-application from which the data is acquired in the central server using IP connectivity and processed using MySQL.

The IoT gateway forms the control tier of the architecture. It has interfaces to the legacy BAS and IoT layer protocols. These include *LonInterface* and *BACnetInterface* which obtain data from legacy BAS. Similarly, there are Ethernet and Wireless interfaces for the IoT layer. In addition, the IoT gateway has services for harnessing information from the IoT devices, web applications, cloud specific services, and extensions that are used for cloud services. The IoT gateway takes the data from both the BAS and IoT layer, and stores it in the cloud database. Further, it is also used for transmitting information to central scheduler or other applications requiring data.

Finally, the cloud tier provides a software-as-a-service (SaaS) layer to the implementation. The central controller is implemented as an application in the cloud. It receives the data from the IoT devices through the IoT gateway. The central scheduler then solves the token allocation problem which takes the token requests from different zones and writes to the cloud data-base the allocated tokens which is transmitted to the zone modules using the IoT gateway using AMQP or MQTT. In addition to the token allocation, the cloud applications also use information aggregated from the BAS and IoT layer to perform data-analytics. This is very useful in forecasting occupancy, weather, heating load and other information required for computing token requests. There are also services for web interface and communication services to enable interfacing with the IoT gateway. Allocated tokens are transmitted to the different zone modules by using cloud's services intended for the purpose. Similarly, it also stores energy consumption and other data pushed from the BAS to its data-base for future analysis. The cloud tier enables decentralized control by providing applications through the SaaS delivery model and using its communication services to the field tier.

In our analysis, the weather forecasts are obtained from a local station using web-services. For forecasting occupancy and heating loads, we considered the error between the model and current measurements as the heating load which is slowly changing during the period. The recursive system identification reduces the error in the model, and the MPC takes corrective steps for the forecast errors due to its feedback action. Consequently, the heating due to occupancy and other loads are lumped into a single term and estimated as cooling loads recursively.

4.1. The IoT components

4.1.1. Sensor module

The function of the sensor module is to sense the temperature, CO₂, and relative humidity, and then transmit the measurements using wireless communication to the zone modules from various rooms in a zone. The sensor module is shown in Fig. 6. The main processor is an Arduino Uno to which the sensors are interfaced. The Uno is selected due to the low power consumption required for the task and is relatively stable due to its well-designed power management circuit. The temperature and humidity are measured using a DHT22 sensor. Similarly, the occupancy is detected using CO₂ sensor MH-Z19.

The measurements are transmitted to the central controller using wireless transceiver nRF24L01+, a module that can transmit up to 50 meters. The Nordic Semiconductor's nRF24L01+ wireless transceiver is designed for operation in the ISM frequency band at 2.400–2.4835 GHz.

It is easy to use and very few external passive components are needed to design a radio system with the nRF24L01+. Within the vicinity of a building, it is robust for 100 meters, which makes it suitable to be used in sensor units in buildings. The nRF24L01+'s baseband protocol engine is based on packet communication and has user configurable parameters like frequency channel, output power, and air data rate. The best advantage of using the nRF24L01+ in our project is that we can configure each zone to be on a different channel, a different payload, and a different data rate, thereby ensuring efficient allocation of resources and keeping out interferences.

4.1.2. Zone controller

The zone controller module consists of the latest Raspberry Pi 3 Model B, an Arduino Pro Mini, a wireless transceiver nRF24L01+, a DHT22 (temperature and humidity sensor), and a CO₂ sensor MH-Z19. The Raspberry Pi 3 is used due to its ability to perform complex computations using Python or other programming languages and the Linux Operation System. Most importantly, it has an Ethernet connectivity that could be used to connect to the cloud via the IoT gateway and log the data to our central server. The zone controller module is shown in Fig. 6(b). The zone controller first integrates the measurements from different zones and then implements the MPC which computes the zone token requests. While it uses measurements from sensor modules to compute the token requests, it interfaces with the cloud to receive the forecast information on heating load, weather, and occupancy. Then it solves the MPC to determine the cooling request and transmits it to the central scheduler via the cloud. In addition, the zone controller uses an Ethernet based communication such as MQTT or AMQP. In addition, the zone module receives the allocated tokens from the central scheduler and computes the mass-flow rate proportional to the cooling energy to the zone actuator.

4.1.3. Zone actuation

The zone actuation is implemented within the *Zone Module* and it receives the mass-flow rate allocated to each zone from the central scheduler and then commands a motor to change the temperature set-point of the VAV. This causes the VAV to either open or close depending on the temperature setting provided by the thermostat and this in turn changes the mass-flow rate for meeting the cooling demand. A Proportional Integral (PI) controller was designed to in the RPi module implementing zone control to reach the desired mass-flow rate. Our zone actuator actually commands a direct current servo motor to change the rotation proportional to the mass-flow rate. The angle of rotation is determined using the differential pressure across the VAV cross-bar which is proportional to the mass-flow rate. The thermostat in our case is the Siemens QAX 30.5 which can be programmed to provide variable temperature settings and has an in-built temperature sensor module that acts as a feedback device to the VAV controller, i.e., Siemens RX 32.5. The flow-rate across the VAV can be determined from the differential pressure sensor across the VAV using the relation [39,40]:

$$F = \kappa \times \sqrt{\Delta P} \quad (7)$$

where κ and $\sqrt{\Delta P}$ denote the VAV factor and differential pressure across the VAV, respectively. Then the flow can be scaled by the density to compute the mass-flow rate. This provides an elegant way to implement decentralizing control without disturbing existing control system in buildings. The zone actuator and its components are shown in Fig. 7 with its hardware realization.

4.1.4. Differential pressure sensor module

The differential air pressure sensor module is required to find the mass-flow rate across the VAV. Our implementation consists of the Raspberry Pi 3, an Arduino Pro Mini, and an SDP 610 differential pressure sensor. The sensor is connected by a pair of tubes across the VAV differential pressure sensor inputs and the computed value is

¹ <http://www.bacnet.org/>

² <http://www.echelon.com/>

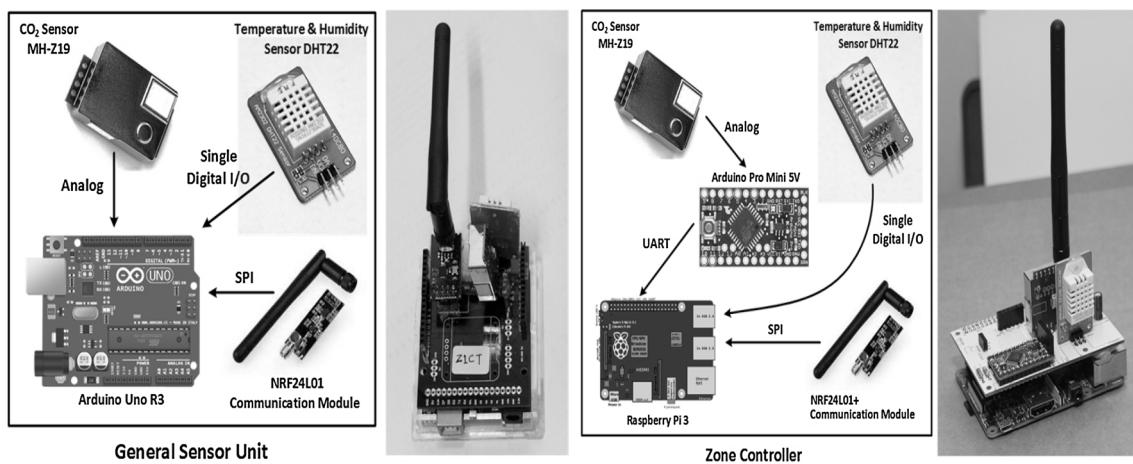


Fig. 6. (a) Sensor Module and its components, (b) Zone controller and its components.

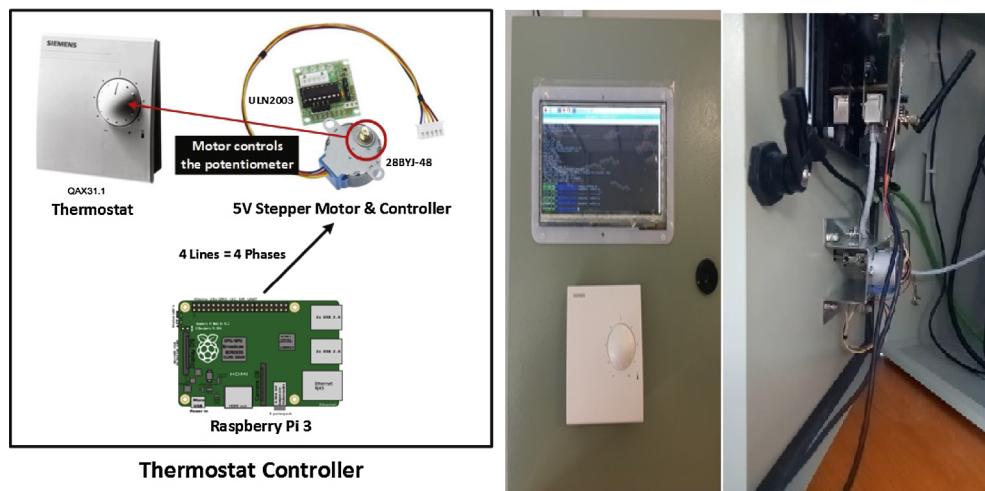


Fig. 7. (a) Zone Actuator and its component, (b) Completed Zone Module Panel with Display.

converted to mass-flow rate using a look-up-table that provides the models the VAV factor at different flow rates and temperature. The differential pressure sensor module is shown in Fig. 8.

4.1.5. AHU instrumentation

The BAS has the AHU instrumentation required for transmitting the power consumption and other AHU relevant data. We use an Agilent data-logger to aggregate the information and transmit to the IoT gateway through a network switch using the Ethernet connection. AHU instrumentation system is shown in Fig. 9.

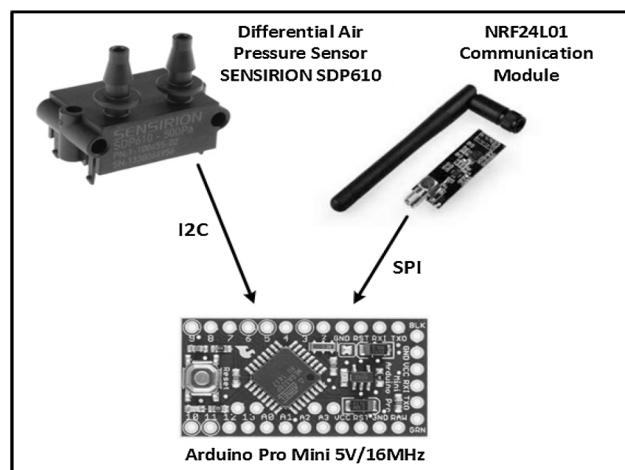
4.1.6. Web interface

A web interface was created to monitor each zone from a web application as shown in Fig. 10. It can be viewed from any hand-held device such as the mobile phone or a computer. Using this interface each zone parameters, control inputs, and AHU measurements can be monitored.

4.1.7. Software architecture

The IoT has different pieces of software working in different layers. In the field layer, the zone module is implemented using Python on Raspberry Pi 3. The MPC and the model update algorithm are easier to implement with the optimization packages such as GNU linear programming kit (GLPK)³ and the Scipy's⁴ optimization routines were used

to update the model. The IoT services are implemented on web based application (e.g., Node JS) for implementing MQTT client. The details of which are avoided here for clarity. The BAS application that transfers data from BACnet and Lonworks are readily available from the vendors and no conversion is made. The IoT gateway has interfaces to the different protocols using python and open source software. The sensor



Air Pressure Sensor Module

Fig. 8. Differential Pressure Sensor Module and its components.

³ <https://www.gnu.org/software/glpk/>

⁴ <https://www.scipy.org/>

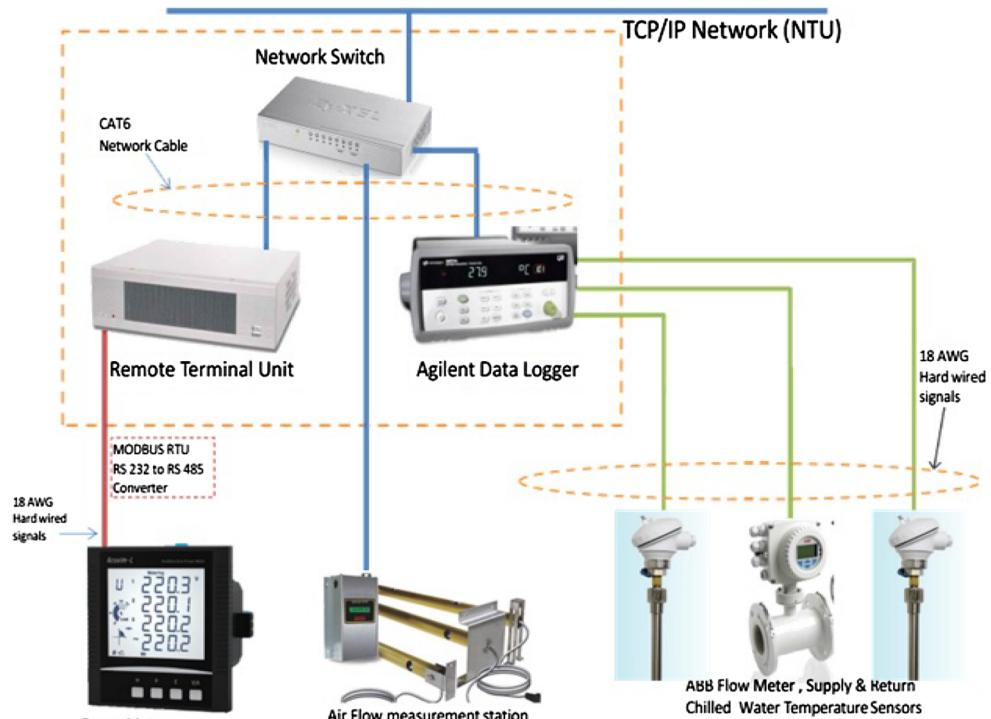


Fig. 9. AHU Instrumentation System.

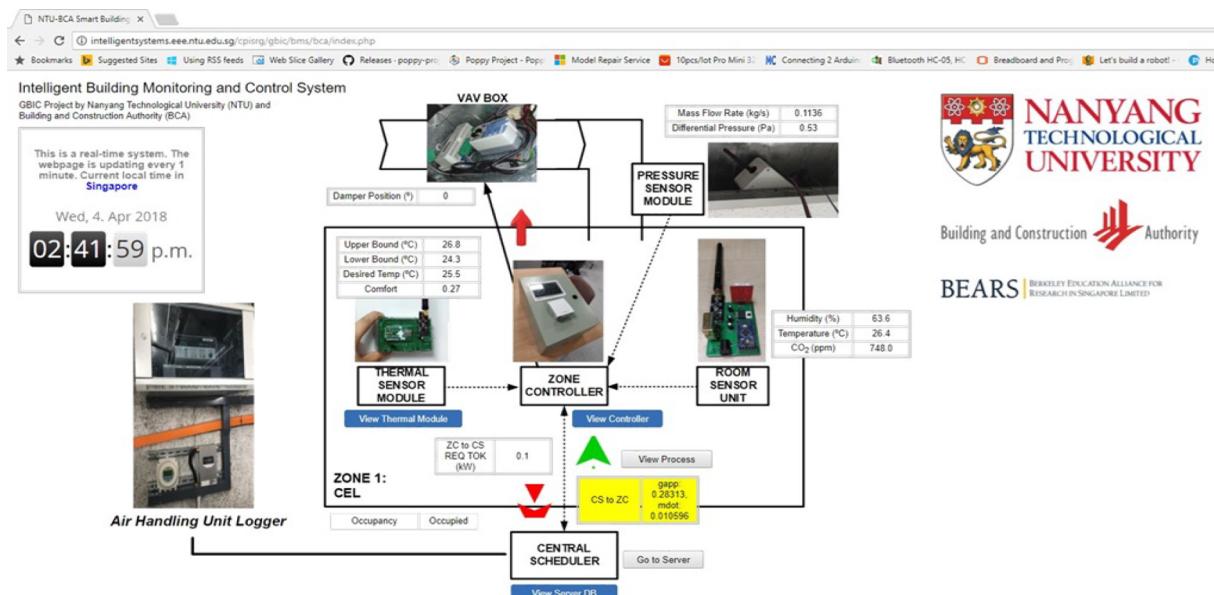


Fig. 10. Web Service for Monitoring.

node measurements implemented on an Arduino are transmitted using NRF24L01 to the RPi which aggregates it using Python libraries (RPi.GPIO, lib_nrf24 and Adafruit_DHT). Then the RPi uses these measurements to compute the optimal cooling energy and transmits it to the central scheduler via Ethernet which is stored in a data-base using MySQL. The software architecture is shown in Fig. 11.

The cloud uses a simple software architecture by using Apache 24 (a http server), PHP 7 + HTML 5 (web), and MySQL (database). The choice is dictated by their open source feature, free to use the software without licensing constraints. The cloud uses Python backend to communicate with the IoT gateway and zone modules. They have routines for communication and data-logging. The central scheduler is

implemented on a Dell server in MATLAB environment with YALMIP⁵. The web interface for the application is shown in Fig. 10. The web application can be used to read real-time data and monitor the control status and energy consumption in the system.

5. Results

This section presents the results of our IoT upgrade deployment and decentralized control on a building test bed having legacy BAS that

⁵ <https://yalmip.github.io/>

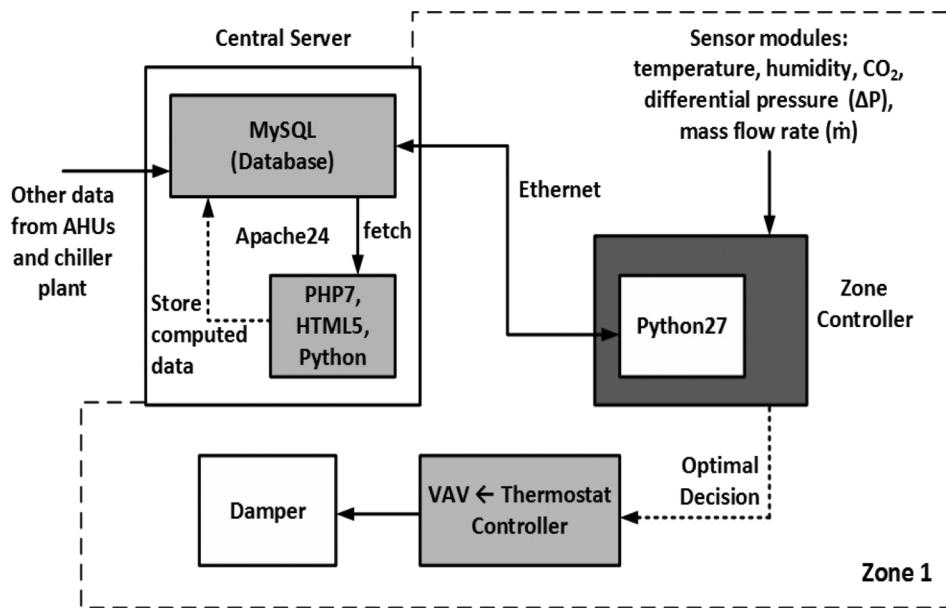


Fig. 11. Software architecture for one zone.

Table 1
Thermal parameters.

Parameter	Value	Unit
c_p	1	kJ K^{-1}
k_f	1.675	$\text{kW s}^{-2} \text{kg}^{-2}$
T_c	14	°C
δ	15	min
H_p	4	h
n_j	7	dimensionless
W	2	dimensionless

integrates AHU control from Siemens Technologies and chiller control from Qunatum Automation. The IoT is implemented as a modular hardware update to the system. The results on energy savings, scalability, implementation aspects, performance of IoT systems, and on various performance metrics are presented.

5.1. Test bed description

The test bed building in our investigation is the building S1 B1 in the School of Electrical Engineering, Nanyang Technological University, Singapore. In particular, the control is deployed in floors B1 and B2. The test bed consists of 85 zones with 6 AHUs being supplied by a single chiller. The chiller also supplies other buildings and floors. Therefore, only fraction of chiller energy is used in the building. The AHU is controlled using a Siemens BAS whereas the chiller controls are provided by Qunatum Automation. Measurements on AHU off-coil temperature and return temperature, relative humidity, pressure, FAD position, CO₂ levels in the AHU, air flow switch and the smoke detector. In addition, the VAV positions in the zone can be monitored from the Siemens controller but cannot be directly commanded as they are controlled by the zone thermostats. The thermal parameters of the HVAC system are shown in Table 1.

The location of zone modules, sensor units, and the AHU instrumentation for each zone within a single AHU is shown in Fig. 12. The VAV floor plan for a single AHU is shown in Fig. 13.

5.2. Experiments on VAV

To implement the TBSA, the mass-flow rate of air in each zone is required and this data cannot be obtained from BAS directly. To compute the mass-flow rate, a differential pressure sensor is connected across the VAV crossbar which measures the pressure difference between the inlet and outlet of the VAV. The zone module computes the mass-flow rate using the Eq. (7). However, still the VAV factor, i.e., κ needs to be computed from experiments. To obtain this parameter, experiments are conducted on VAV to record the differential pressure and the flow rate in L s^{-1} . The relation between the mass-flow rate and differential pressure across the VAV cross bar is shown in Fig. 14(a) and the plot between percentage damper opening versus mass-flow rate is shown in Fig. 14(b).

To open or close the VAV, the zone module controls a motor for adjusting the set-point which should correspond to the desired mass-flow rate provided by the allocated tokens. Experiments were conducted maintaining AHU pressure and temperature constants, the plot between differential pressure and variations in the set-point are shown in Fig. 15. It can be seen that as the temperature set-point increases the flow decreases, and vice versa (see Fig. 16).

5.3. Occupant thermal comfort

The occupant thermal comfort is modelled as thermal sensation which give the lower and upper bounds $T_{ll}(k)$ and $T_{lh}(k)$ of the optimization problem ($\mathcal{P}2$). Data collected from the sensors are: indoor air temperature, humidity, air-velocity, mean radiant temperature, clothing insulation and metabolic rate. Among these variables, the clothing factor is fixed at 0.71 clo based on the ASHRAE standard [32] considering that the occupants wear trousers and short-sleeve shirts working on executive chair. Since, the users employ the zones for reading and writing, we fix the metabolic rate as 1.0 met. We then use the measurements on relative humidity and temperature from sensor nodes shown in Fig. 6. The air-flow is measured using a sensor node and is usually kept at pre-defined values for modelling (e.g., 0.1 m s). These parameters were used to model the thermal sensation scale which provided the user-defined temperature bounds.

5.3.1. Model adaptation

An important improvement to the MPC is the model update step and

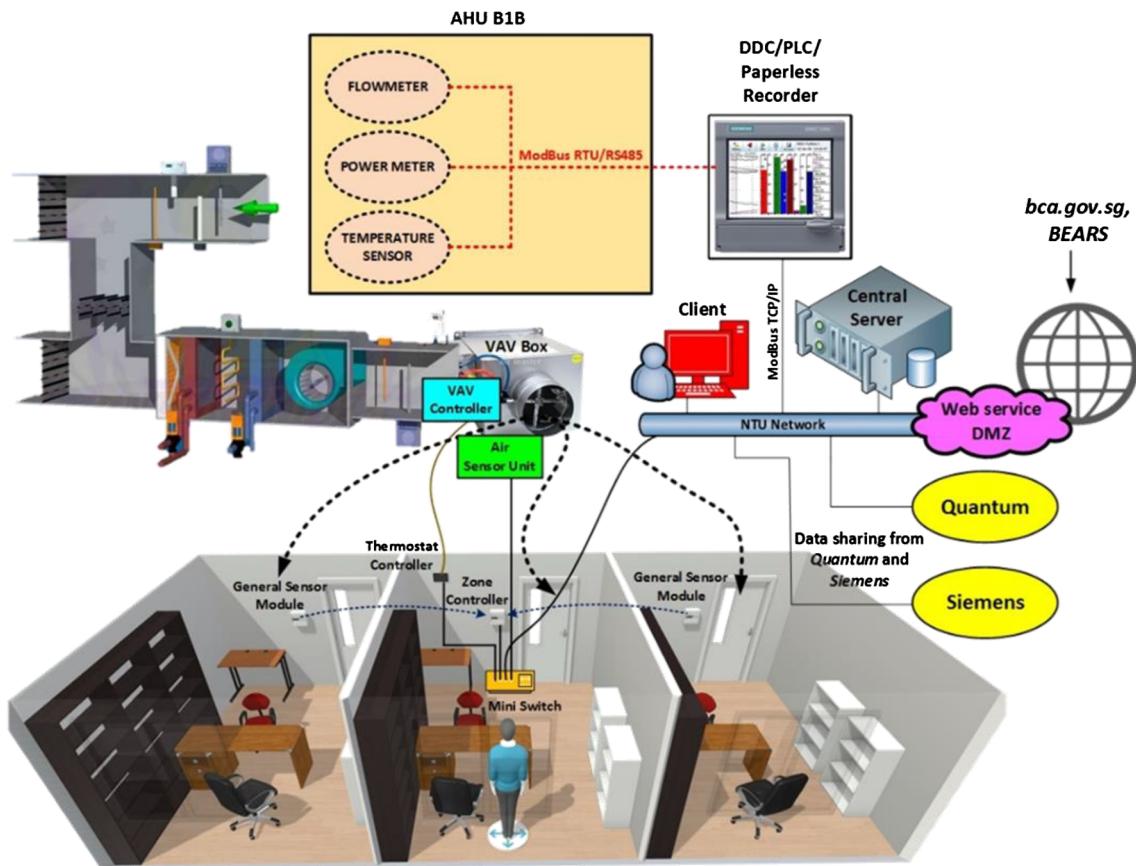


Fig. 12. Location of Zone Modules and Sensors in Each Zone.

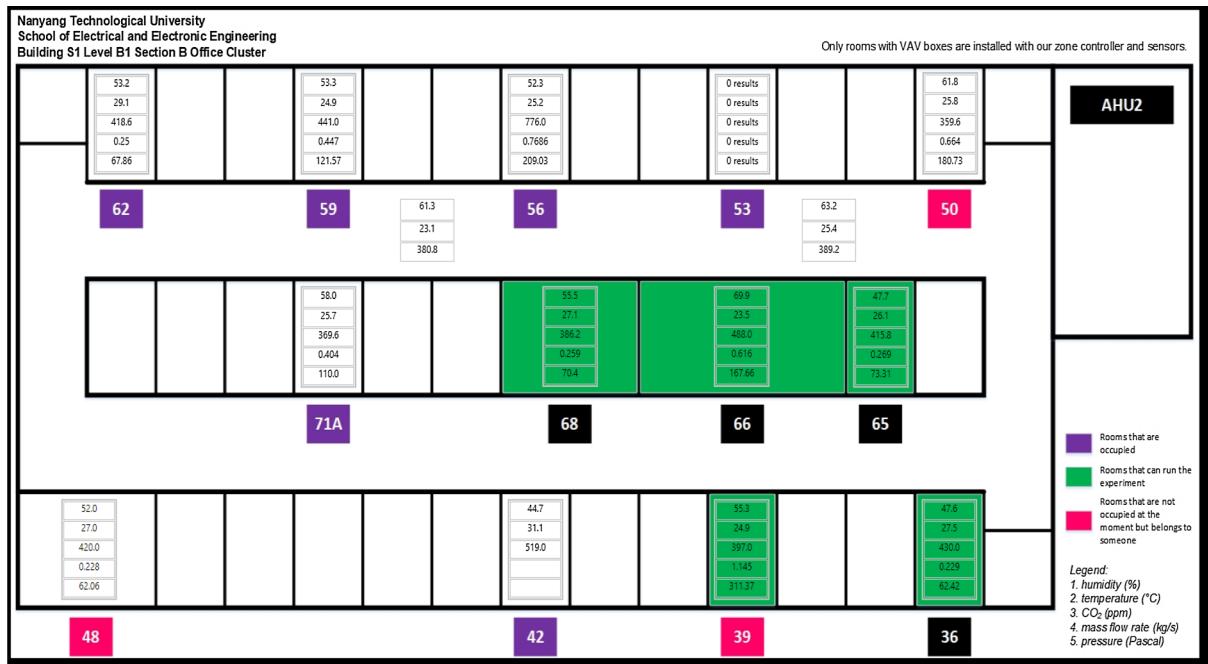


Fig. 13. VAV Floor Plan for a Single AHU.

it provides significant benefits as the zone module learns the model using measurements and forecast. Exploiting the monitoring capability of IoT, the zone thermal model can be adapted or learned by solving the optimization problem (\mathcal{P}_1) during each time-epoch to improve the smartness of the system. The recursive estimation was implemented in

Raspberry Pi with a Python code and our experiments showed that the recursive identification computes the model with less than 2% error with a sampling time of 900 s as can be shown in Fig. 17. This results shows that the recursive identification helps improving the agility, adaptability, and smartness of the control approach. The data was

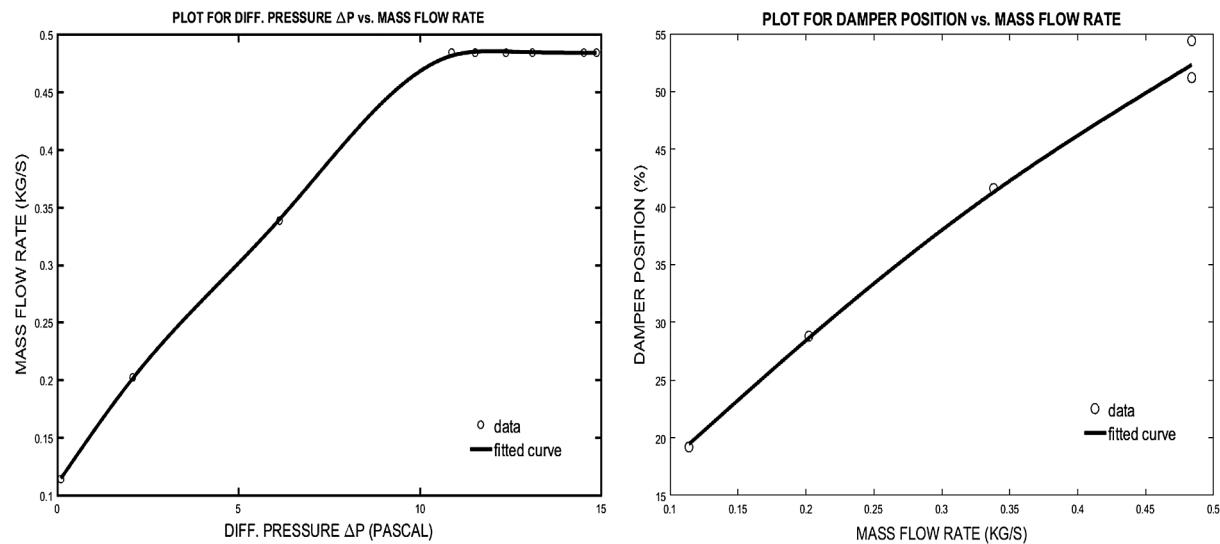


Fig. 14. (a) Plot of mass-flow rate in kg s^{-1} versus differential pressure, (b) VAV mass-flow rate in kg s^{-1} versus per percentage damper position.

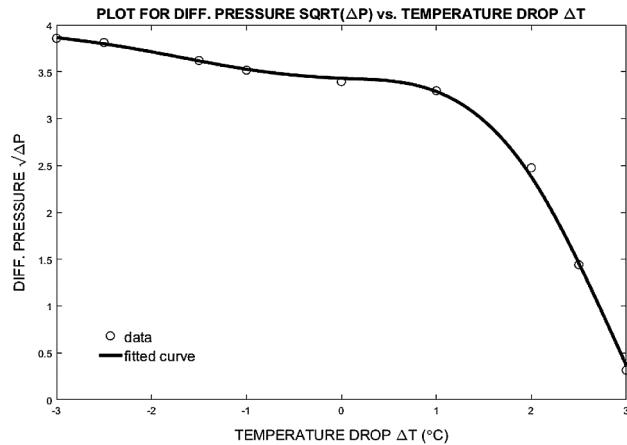


Fig. 15. Plot for variations in set-point temperature to VAV versus the differential pressure.

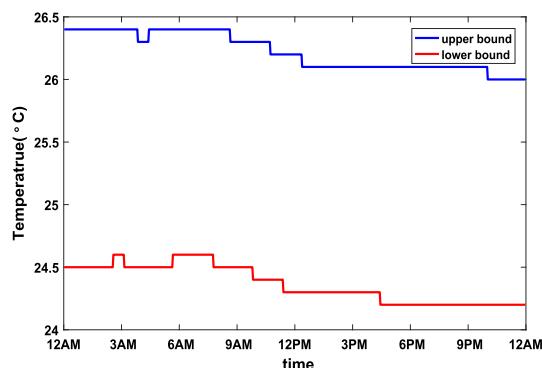


Fig. 16. The temperature bounds in which the indoor thermal condition is neutral and the users feel comfort.

collected by deploying IoT sensors to measure room temperature, ambient temperature obtained from weather, and the cooling energy applied from central Scheduler. The experiments were performed on 27/2/2018 and 28/2/2018 and the recursive method was used to identify the zone thermal model. Our studies showed that the estimates tracked the measurements with proper initialization. However, when initial estimates were not closer to actual, it takes a longer time to converge to optimal. Consequently, proper initialization is paramount to reduce the

computation time.

Remark 7. During Model Adaptation it was observed that there is a slight skew in the actual model and identified one. This is mainly due to the fact that the adaptation starts after obtaining the measurements.

5.4. Performance of Smart-TBSA

5.4.1. International performance measurement and verification protocol

To study the energy savings performance of the IoT upgrade, we use the IPMVP protocol option B, a widely used standard to study retrofits. Selection of the standard was motivated by recommendations from industry and the current standards enforced by building authorities in Singapore. The savings are computed as

$$E_S = E_B - E_R \pm A \quad (8)$$

where E_S , E_B and E_R are the energy savings, energy consumed in the base-line period, and energy consumption during reporting period, respectively. In addition, the A is the adjustment term that is used to compensate for weather conditions, sensor accuracies or other conditions. The method for computing energy savings is shown in Fig. 18.

During the base-line period, thermostat controller for VAV was used and no centralized optimization algorithm is available in the BAS. Therefore, the investigation compares the experimental performance with thermostat controller for having consistency in the results and meeting the standard requirements.

5.4.2. Smart-TBSA versus legacy controller

The energy savings performance of the Smart-TBSA for a duration of 16 hour in the reporting period with the control applied at a sampling time of 900 s is presented. This is compared with the performance of the legacy controller. The energy savings of the central controller for each zone is shown in Fig. 19. The bar chart shows the cumulative heating load and the cooling energy supplied to each of them for the duration. A close inspection reveals that even though the average savings in some zones are low due to thermal disturbances, the average energy savings is close to 18–20%, a significant savings that can lead to quick payback times.

Fig. 20 shows the energy savings with the Smart-TBSA and legacy controller recorded on similar days. One important observation was that the difference in energy savings at the zone level for a period of 7–8 working hours is around 10–12% and it reduces during off-peak load period. Furthermore, one can see that in the later part of the day, the energy savings increase due to the smartness embedded using sensors.

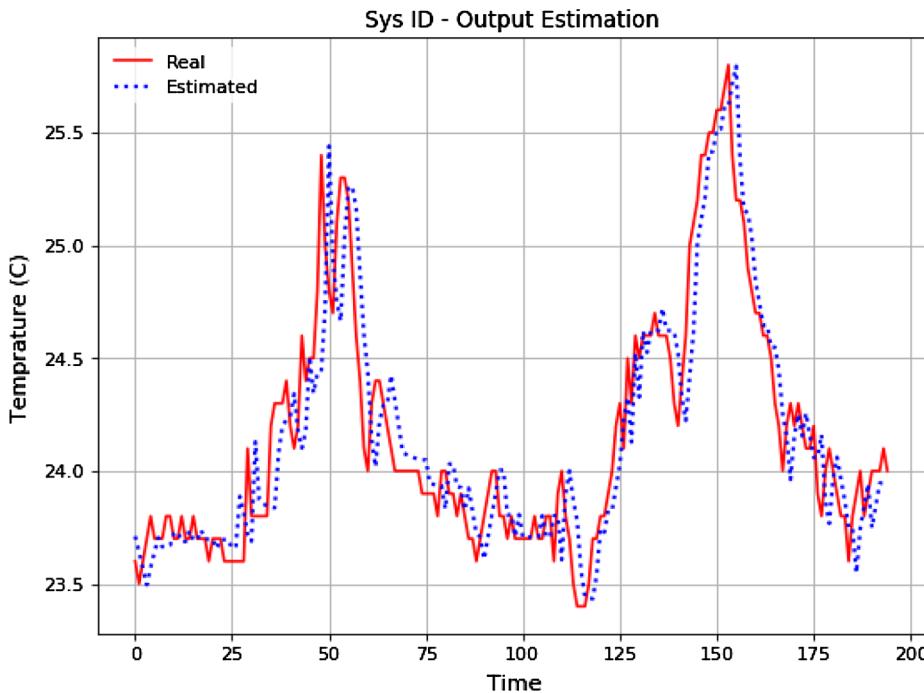


Fig. 17. Results on Model Adaptation for 200 min.

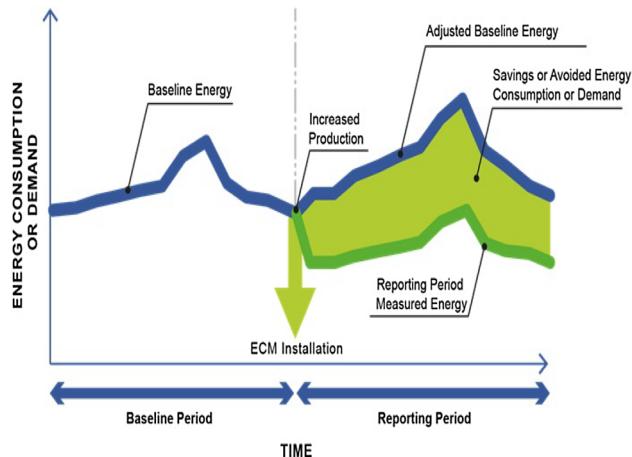


Fig. 18. Illustration of the IPMVP.

In addition, we performed the energy saving comparison over one AHU comprising of 13 zones and supplied from a chiller during October–December. The AHU fan consumption in (kW) and chiller energy consumption (kW) equivalent of cooling load supplied on Friday's having similar cooling load profiles is shown in Figs. 21 and 22, respectively. Experiments on different weeks during the period revealed that the energy savings varied between 15 and 31% depending on the cooling load, ambient temperature, and disturbances. In average the energy savings were between 18 and 23% in winter with the proposed Smart-TBSA.

The computation time and the energy savings performance of the Smart-TBSA is studied using models fitted from the real-time data and emulating the zones is shown in Table 2. The performance suggests that the computation time is not significant compared to the sampling time of the system. A detailed study on the scalability of the algorithm up to 300 zones can be found in [28]. Moreover, the current implementation of the Smart-TBSA central scheduler uses the YALMIP package in MATLAB which slows down the implementation. However, the YALMIP's symbolic capability provides significant advantages and flexibility to

dynamically change the optimization model of the central controller. In spite of this, the computation time is still reasonable for our sampling time and hence its adapted. Details of computation time integrating CPLEX and MATLAB can be found in [28].

5.4.3. Zone module performance

In addition to monitoring, the zone module implements the decentralized control as MPC computations with a Python code that solves the optimization model in (\mathcal{P}_2) using the GLPK solver. The zone module is mounted on the room having the VAV, and as a single zone in the test bed consists of three rooms, it takes the average of the measurements from the other two rooms. On receiving the measurements, it executes the MPC to compute the optimal cooling energy. One execution of the zone module takes around 5–6 ms and the algorithm sampling time is 900 s. Further, the communication time between the zone and sensor module is 800–1000 μ s with a sensing range of 50 m. The average delay in the sensing module in the zone is around 200 μ s. The packet-losses from the sensor module to the zone module is around 2–3% and from the zone module to the central controller is around 1%. However, re-transmissions of the variable is inherently handled due to the Ethernet based communication. The zone modules push the measurements, control variables and other information into the cloud's database for future use. They receive forecasts on thermal load and weather information from the cloud.

5.5. Cost-benefit analysis

Our analysis shows that the payback time for the IoT upgrade is around 1–1.5 years. This is computed taking per unit electricity charges in Singapore as the baseline and assuming an average 20% energy savings over a day. Additional savings in the form of reduced carbon emissions and cost savings during peak-periods have not been considered. The cost was obtained for 85 zones and can vary as the number of zones differ. Additionally, since sensor modules installed in the rooms adjacent to the one containing the VAV, increases the data aggregated by 200% and the web interface enhances the data-visualization by the same amount. These modifications are obtained with little hardware upgrades and at minimum cost. The key KPIs achieved with

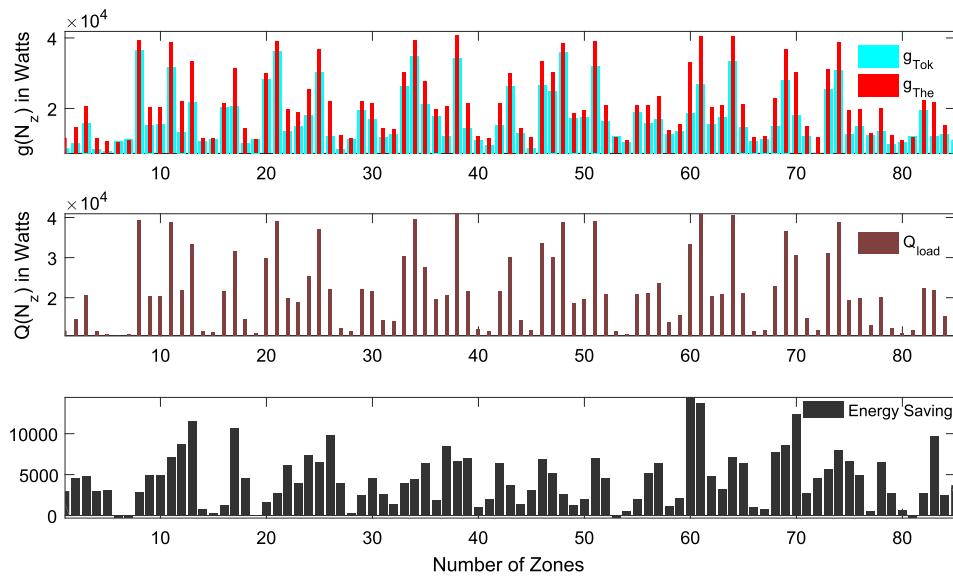


Fig. 19. Results for 16 Hours with Token Based Scheduling versus Thermal Control on 85 Zones.

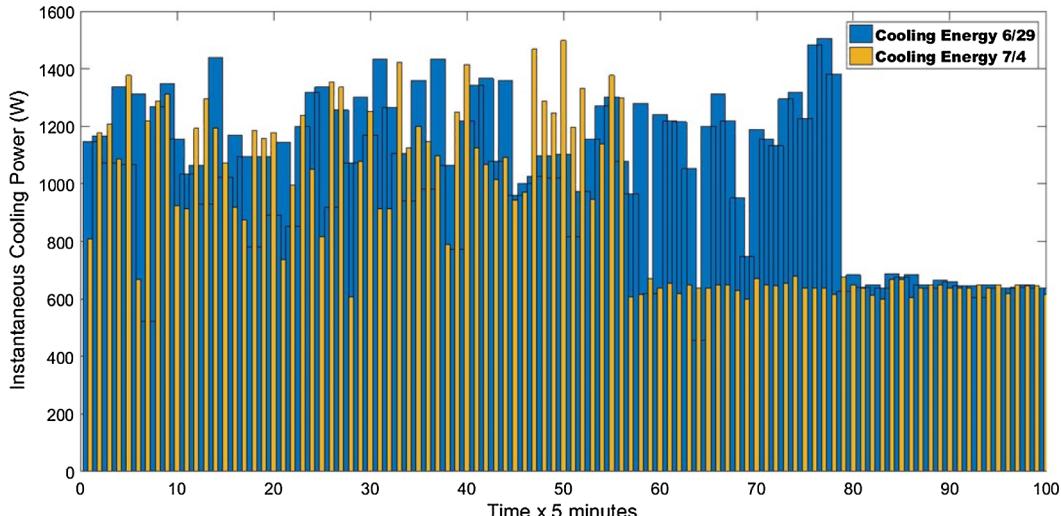


Fig. 20. Zone Controller Comparison.

the Smart-TBSA is shown in Table 3.

6. Conclusions

This investigation presented a scalable and economic IoT upgrade for legacy BAS in buildings for decentralized HVAC control. The method combined the IoT with a hierarchical and decentralized control strategy called the *Smart-Token Based Scheduling Algorithm*. From the methodological viewpoint the Smart-TBSA enhanced the previous works of the author with three elements: (i) recursive model adaptation, (ii) occupant thermal comfort modelled using SLFN, and (iii) soft-constraints to temperature bounds. These modifications provided desirable features such as adaptability, agility, smartness, reliability, and energy efficiency. The investigation also presented an IoT architecture and prototype describing the different components. The architecture, components and their interactions of the IoT prototype were explained in detail. This included discussions on the hardware selection and design with detailed realization, software components, and the integration. Further, communication among different IoT components as well with the legacy BAS was provided. The hardware components included

the *Sensor Module*, *Zone Module*, *Zone Actuator* and *Central Scheduler*. In addition, the description of the IoT gateway for communicating with the BAS and cloud was also provided. Discussions on realizing the controller as an upgrade to existing BAS was also provided which included experiments to model HVAC components such as the variable air volume (VAV), fresh air damper and others were presented.

The IoT prototype and the Smart-TBSA algorithm were demonstrated in a test bed building consisting of 85 zones at the Nanyang Technological University, Singapore. The results demonstrated that the TBSA and IoT integration saved 20% energy and provided good scalability. Further, the TBSA can be implemented as an economic and scalable upgrade to legacy BAS system in the building. The upgrades were modular, simple and cost effective. The proposed approach provides a sub-optimal solution and more suited for commercial buildings which are the limitations of the proposed approach. Further, robustness to IoT related constraints such as packet-losses, noise, and disturbance effects on sensor measurements needs to be studied. Dealing with occupant preferences and IoT constraints are future course of this investigation.

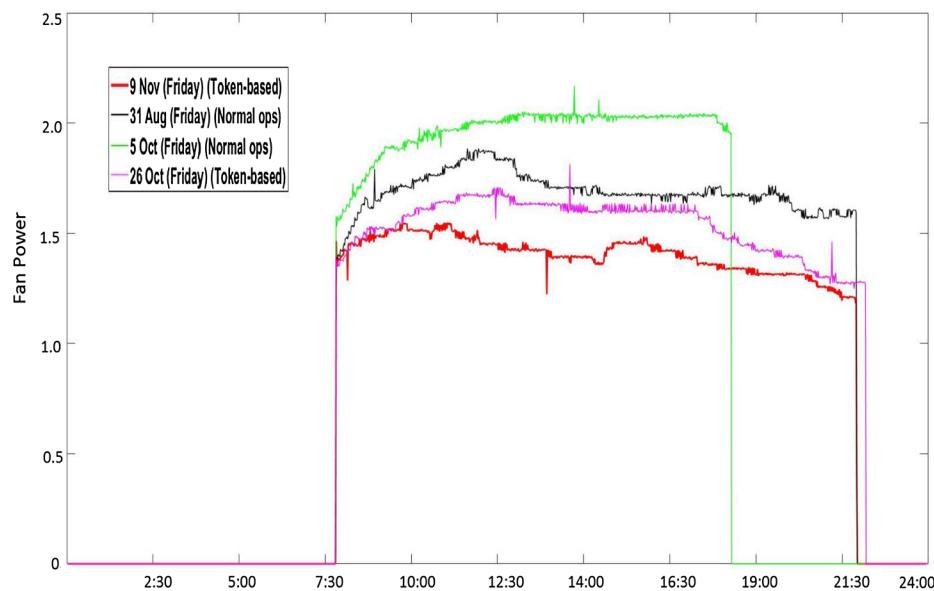


Fig. 21. Fan Power Consumption in Winter.

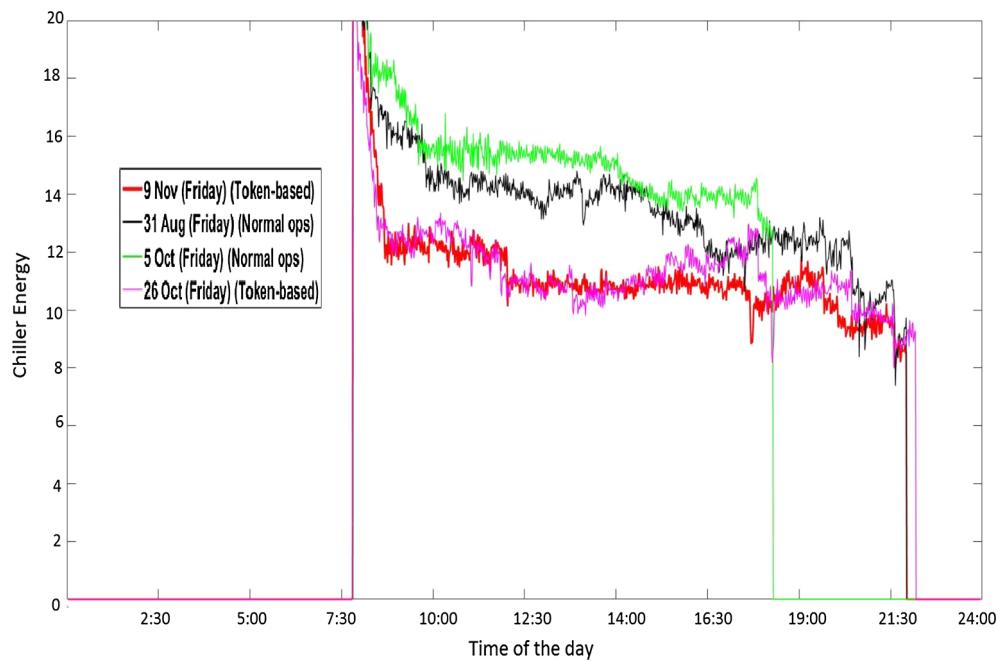


Fig. 22. Chiller Power Consumption in Winter.

Table 2

Results for 16 Hours Simulations Token Based Scheduling versus Classical Control.

#zones	Computation time in [s]	Saving in % 16 h
5	11.76	26.95
10	17.78	23.07
25	31.95	16.52
50	55.9	24.68
85	90.48	20.53
100	96.79	17.78

Table 3

Key Performance Indicators.

S.No.	KPI	Comment
1	Scalability	Up to 300 zones
2	Decentralization	Partial AHU and Chiller centralized
3	Upgrade cost	Minimum
4	Energy Savings	Average 20%
5	Model Update	Recursive
6	System Agility	Increased
7	Data visibility	Up by 40%
8	Flexibility	Increased by modular design
9	Payback time	1–1.5 years
10	Cloud Integration	Yes

References

- [1] Radhakrishnan N, Su Y, Su R, Poolla K. Token based scheduling for energy management in building HVAC systems. *Appl Energy* 2016;173:67–79.
- [2] Cao X, Dai X, Liu J. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy Buildings* 2016;128:198–213.
- [3] Minoli D, Sohraby K, Occhiogrosso B. IoT considerations, requirements, and architectures for smart buildings – energy optimization and next generation building management systems. *IEEE Internet Things J* 2017;4(1):1. [https://doi.org/10.1109/JIOT.2017.2647881 <http://ieeexplore.ieee.org/document/7805265/>](https://doi.org/10.1109/JIOT.2017.2647881).
- [4] Coates A, Hammoudeh M, Holmes KG. Internet of things for buildings monitoring: experiences and challenges. Proceedings of the international conference on future networks and distributed systems. ACM; 2017. p. 38.
- [5] Li W, Logenthiran T, Phan V-T, Woo WL. Implemented IoT-Based Self-Learning Home Management System (SHMS) for Singapore. *IEEE Internet Things J* 2018;5(3):2212–9.
- [6] Al-Ali A, Zualkernan IA, Rashid M, Gupta R, Alikarar M. A smart home energy management system using IoT and big data analytics approach. *IEEE Trans Consum Electron* 2017;63(4):426–34.
- [7] Zhang X, Adhikari R, Pipattanasomporn M, Kuzlu M, Rahman S. Deploying IoT devices to make buildings smart: performance evaluation and deployment experience. *Internet of things (WF-IoT)*, 2016 IEEE 3rd world forum on. IEEE; 2016. p. 530–5.
- [8] Pan J, Jain R, Paul S, Vu T, Saifullah A, Sha M. An internet of things framework for smart energy in buildings: designs, prototype, and experiments. *IEEE Internet Things J* 2015;2(6):527–37. [https://doi.org/10.1109/JIOT.2015.2413397. arXiv:1603.08025](https://doi.org/10.1109/JIOT.2015.2413397).
- [9] Javed A, Larijani H, Ahmadiin A, Emmanuel R, Mannion M, Gibson D. Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for HVAC. *IEEE Internet Things J* 2017;4(2):393–403. <https://doi.org/10.1109/JIOT.2016.2627403>.
- [10] Serra J, Pubill D, Antonopoulos A, Verikoukis C. Smart HVAC control in IoT: energy consumption minimization with user comfort constraints. *Sci World J* 2014.
- [11] Ruano A, Silva S, Duarte H, Ferreira PM. Wireless sensors and iot platform for intelligent HVAC control. *Appl Sci* 2018;8(3):370.
- [12] Kastner W, Kofler M, Jung M, Gridling G, Weidinger J. Building automation systems integration into the internet of things the IoT6 approach, its realization and validation. Emerging technology and factory automation (ETFA), 2014 IEEE. IEEE; 2014. p. 1–9.
- [13] Tushar W, Wijerathne N, Li W-T, Yuen C, Poor HV, Saha TK, et al. Internet of things for green building management: disruptive innovations through low-cost sensor technology and artificial intelligence. *IEEE Sig Process Mag* 2018;35(5):100–10.
- [14] Bekiroglu K, Srinivasan S, Png E, Su R, Poolla K, Lagoa C. An internet of things compliant model identification methodology for smart buildings. 2017 IEEE 56th Annual conference on decision and control (CDC). ieeexplore.ieee.org; 2017. p. 4440–5.
- [15] Adhikari R, Pipattanasomporn M, Rahman S. An algorithm for optimal management of aggregated HVAC power demand using smart thermostats. *Appl Energy* 2018;217:166–77.
- [16] Ma Y, Matusko J, Borrelli F. Stochastic model predictive control for building hvac systems: complexity and conservatism. *IEEE Trans Control Syst Technol* 2015;23(1):101–16. <https://doi.org/10.1109/TCST.2014.2313736 <http://ieeexplore.ieee.org/document/6802411/>>.
- [17] Bengea SC, Kelman AD, Borrelli F, Taylor R, Narayanan S. Implementation of model predictive control for an HVAC system in a mid-size commercial building. *HVAC R Res* 2014;20(1):121–35. <https://doi.org/10.1080/10789669.2013.834781>.
- [18] Wu Z, Jia Q-S, Guan X. Optimal control of multiroom HVAC system: an event-based approach. *IEEE Trans Control Syst Technol* 2016;24(2):662–9.
- [19] Long Y, Liu S, Xie L, Johansson KH. A hierarchical distributed mpc for HVAC systems. American control conference (ACC), 2016. IEEE; 2016. p. 2385–90.
- [20] Koehler S, Borrelli F. Building temperature distributed control via explicit MPC and “trim and respond” methods. Control conference (ECC), 2013 European. IEEE; 2013. p. 4334–9.
- [21] Koehler S, Danielson C, Borrelli F. A primal-dual active-set method for distributed model predictive control. *Optimal Control Appl Methods* 2017;38(3):399–419.
- [22] Ma Y, Anderson G, Borrelli F. A distributed predictive control approach to building temperature regulation. *Am Control Conf* 2011:2089–94. <https://doi.org/10.1109/ACC.2011.5991549 <http://ieeexplore.ieee.org/document/5991549/>>.
- [23] Putta V, Zhu G, Kim D, Hu J, Braun J. Comparative evaluation of model predictive control strategies for a building HVAC system. American control conference (ACC), 2013. IEEE; 2013. p. 3455–60.
- [24] Oldewurtel F, Jones CN, Parisio A, Morari M. Stochastic model predictive control for building climate control. *IEEE Trans Control Syst Technol* 2014;22(3):1198–205.
- [25] Pang Y, Hu G, Spanos CJ. Distributed model predictive control of a hvac system via a projected subgradient method. 2018 IEEE 14th international conference on control and automation (ICCA). IEEE; 2018. p. 878–83.
- [26] Zhang X, Schildbach G, Sturzenegger D, Morari M. Scenario-based MPC for energy-efficient building climate control under weather and occupancy uncertainty. 2013 European control conference, ECC 2013. 2013. p. 1029–34.
- [27] Yu L, Xie D, Jiang T, Zou Y, Wang K. Distributed real-time hvac control for cost-efficient commercial buildings under smart grid environment. *IEEE Internet Things J* 2018;5(1):44–55.
- [28] Radhakrishnan N, Srinivasan S, Su R, Poolla K. Learning based hierarchical distributed HVAC scheduling with operational constraints. *IEEE Trans Control Syst Technol Accepted* 2017:1–8.
- [29] Kaam S, Raftery P, Cheng H, Paliaga G. Time-averaged ventilation for optimized control of variable-air-volume systems. *Energy Buildings* 2017;139:465–75.
- [30] Paganini F. A set-based approach for white noise modeling. *IEEE Trans Autom Control* 1996;41(10):1453–65.
- [31] Haykin SS. Adaptive filter theory. Pearson Education India; 2008.
- [32] A. Standard, Standard 55-2004: Thermal environmental conditions for human occupancy; 2004.
- [33] Rupp RF, Vásquez NG, Lamberts R. A review of human thermal comfort in the built environment. *Energy Buildings* 2015;105:178–205.
- [34] Kwong QJ, Adam NM, Sahari B. Thermal comfort assessment and potential for energy efficiency enhancement in modern tropical buildings: a review. *Energy Buildings* 2014;68:547–57.
- [35] Chauvin Y, Rumelhart DE. Backpropagation: theory, architectures, and applications. Psychology Press; 2013.
- [36] Moré JJ. The levenberg-marquardt algorithm: implementation and theory. Numerical analysis. Springer; 1978. p. 105–16.
- [37] Kelman A, Borrelli F. Bilinear model predictive control of a HVAC system using sequential quadratic programming. IFAC World Congress. 2011.
- [38] Merz H, Hansemann T, Hübner C. Building automation: communication systems with EIB/KNX, LON and BACnet. Springer Science & Business Media; 2009.
- [39] Okochi GS, Yao Y. A review of recent developments and technological advancements of variable-air-volume (vav) air-conditioning systems. *Renew Sustain Energy Rev* 2016;59:784–817.
- [40] Pang X, Piette MA, Zhou N. Characterizing variations in variable air volume system controls. *Energy Buildings* 2017;135:166–75.