Co-design of Control Algorithm and Embedded Platform for Building HVAC Systems

Mehdi Maasoumy*
University of California
Berkeley, CA 94720, USA
mehdi@me.berkeley.edu

Qi Zhu
University of California
Riverside, CA 92521, USA
qzhu@ee.ucr.edu

Cheng Li
Nanyang Tech. University
639798 Singapore
li@arch.ethz.ch

Forrest Meggers
Singapore-ETH Centre
Future Cities Laboratory
118999 Singapore
meggers@arch.ethz.ch

Alberto Sangiovanni-Vincentelli
University of California
Berkeley, CA 94720, USA
alberto@eecs.berkeley.edu

ABSTRACT

The design of heating, ventilation and air conditioning (HVAC) systems is crucial for reducing energy consumption in buildings. As complex cyber-physical systems, HVAC systems involve three closely-related subsystems – the control algorithm, the physical building and environment and the embedded implementation platform. In the traditional top-down approach, the control algorithm and the embedded platform are in general designed separately leading to sub-optimal systems. We propose a co-design approach that analyzes the interaction between the control algorithm and the embedded platform through a set of interface variables (in this paper we address in particular sensing accuracy). We present six control algorithms that take into account the sensing error, and model the relation of control performance and cost versus sensing error. We also capture the relation of embedded platform cost versus sensing error by analysis of the collected data from a testbed. Based on these models, we explore the co-design of the control algorithm and the temperature sensing subsystem of the embedded platform to optimize with respect to energy cost and monetary cost while satisfying the constraints for user comfort level.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems

General Terms

Platform-based design

*Corresponding author.

ICCPS13 April 8 – 11, 2013, Philadelphia, PA, USA.

Keywords

co-design, building energy efficiency

1. INTRODUCTION

Total primary energy consumption in the United States increased from 78.3 quads[1] in 1980 to over 100 quads in 2008, of which the building sector accounts for about 40%. The building sector is also responsible for almost 40% of greenhouse gas emissions and 70% of electricity use. About 50% of the energy consumed in buildings is directly related to space heating, cooling and ventilation[1]. Therefore, reducing building energy consumption by designing smart control systems to operate the heating, ventilation and air conditioning (HVAC) system in a more efficient way is critically important to address energy and environmental concerns worldwide.

Smart buildings today have sophisticated and distributed control systems as part of a Building Automation System (BAS). The task of a BAS is to maintain building climate within a specified range, control the lighting based on the occupancy schedule, and monitor the system performance and failures. To accomplish these tasks, a BAS has to deal with computation and communication non-idealities stemming from the distributed nature of the implementation platform.

The design of HVAC systems involves three main subsystems – the physical building and its environment, the control algorithm that determines the system operations based on sensing inputs from the building and the environment, and the embedded platform that implements the control algorithm. In the traditional top-down approach, the design of the HVAC control algorithm is done without explicit consideration of the embedded platform. The underlying assumption is that the computation and communication capabilities of the embedded platform are sufficiently performing for any type of control mechanism. However, with the advent of more complex HVAC control algorithms for energy efficiency, the use of distributed networked platforms, and the imposition of tighter requirements for user comfort, this assumption on the embedded platform is no longer valid. Various aspects of the platform, including sensor accuracy and

[1]A quad is a unit of energy equal to 1.055×10^{18} joules.

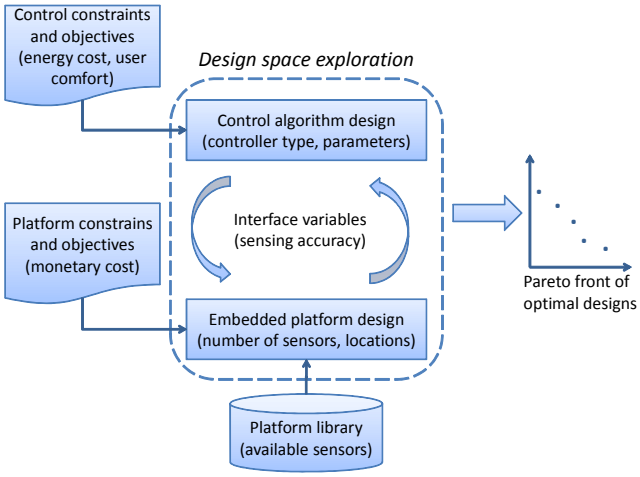

Figure 1: Co-design framework for HVAC systems

availability, communication channel reliability, and computing power of embedded processors, may have a significant impact on the quality and cost of a BAS. Thus, the design of the control algorithm should take into account the configuration of the implementation platform and *vice versa*, i.e., the control algorithm and the embedded platform should be *co-designed*.

In this paper, we analyze the performance and energy cost of six HVAC control algorithms under different assumptions on temperature sensing accuracy. Indeed, we observed in our experiments that the accuracy of sensing data has significant impact on the control algorithms, to the extent that *different algorithms should be chosen depending on the different sensing accuracy of the implementation platform* . Based on this observation, we propose a framework to co-design the HVAC control algorithm with the part of the embedded platform that directly affects sensing accuracy – specifically the choices of sensor locations and the number of sensors. We believe this approach can be extended to a general framework for co-designing the control algorithm and the entire embedded platform. Fig. 1 shows the structure of the co-design framework, with blocks in parentheses representing the focus of this paper.

As shown in Fig. 1, the key aspects of the co-design process are 1) identifying the *interface variables* that are significant to both the control algorithm and the embedded platform, 2) designing the control algorithm with consideration of the interface variables and modeling the relation of control metrics (e.g. control performance) versus the interface variables, 3) capturing the relation of platform metrics (e.g. platform monetary cost) versus the interface variables, and 4) co-exploring the design of the control algorithm and the platform through the interface variables.

In this paper, we focus on temperature sensing accuracy, which on one side significantly affects the control algorithm and on the other directly relates to the design of the sensing system. During the exploration of the design space, for any given monetary budget, we find the optimal choice of sensor locations and the number of sensors to maximize the accuracy of sensing data. Sensing accuracy is then used for the selection of the HVAC control algorithm (from the six candidate controllers) to minimize energy cost while satisfying user comfort requirements. We set the monetary budget to a range of different values, and find the design that consumes the minimal energy for each given budget. The result of this design space exploration is a Pareto front of optimal monetary cost and energy cost.

Various models and control algorithms have been proposed in the literature for HVAC systems [2, 3, 4, 5, 6]. In a recent work, unscented Kalman filtering has been used for online estimation of building thermal parameter estimation [7]. The focus of these papers is on physical modeling and control design without taking into account the limitations of the embedded platform. For instance, in [8], weather and occupancy prediction uncertainties are considered in the control design process, and a robust model predictive control mechanism against prediction uncertainties is derived. However, the uncertainties (errors) from the embedded platform measurements are not addressed. Another set of papers [9, 10] focuses on the design of the embedded software and hardware for a given control algorithm, thus not addressing design space exploration for optimal HVAC system design.

The main contributions of this paper are the following:

- We propose six different control algorithms that take into account sensing accuracy, including two on-off controllers, two model predictive controllers (MPC) and two robust model predictive controllers (RMPC), each type with either extended Kalman filter (EKF) or unscented Kalman filter (UKF). We capture the relation between the control performance/cost and sensing accuracy in Simulink models for simulation.

- We establish the relation between sensing accuracy and the number and locations of temperature sensors, based on the analysis of measurement data collected from a well-instrumented testbed.

- We propose a framework for co-designing the control algorithm and the embedded platform, and apply it to a specific co-design case of the HVAC control algorithm (by choosing from the six candidate controllers) and the sensing system (by choosing the temperature sensor number and locations) to optimize energy and monetary cost.

The remainder of the paper is organized as follows. Section 2 presents preliminaries. Section 3 addresses the modeling of sensing and prediction accuracy in the system dynamics. Section 4 presents the design of the control algorithms with EKF and UKF to address sensing accuracy, and demonstrate the impact of sensing accuracy on the control performance and cost. Section 5 introduces an approach for determining the relation between sensing accuracy and the number and locations of sensors. Section 6 explores the design space for both control algorithm and sensing system, based on the models we build in Section 4 and 5. Section 7 concludes the paper and discusses future work.

2. PRELIMINARIES

We use the model that was proposed in [6] where a building is considered as a network. There are two types of nodes in the network: walls and rooms. There are in total n nodes, m of which represent rooms and the remaining $n-m$ nodes

represent walls. The temperature of the i-th wall is governed by the following equation:

$$C_{w_i}\frac{dT_{w_i}}{dt} = \sum_{j\in\mathcal{N}_{w_i}}\frac{T_j - T_{w_i}}{R'_{ij}} + r_i\alpha_i A_i q''_{rad_i} \quad (1)$$

where T_{w_i}, C_{w_i}, α_i and A_i are the temperature, heat capacity, absorption coefficient and area of wall i, respectively. R'_{ij} is the total resistance between wall i and node j. q''_{rad_i} is the radiative heat flux density on wall i. $\mathcal{N}_{w_i}$ is the set of all of neighboring nodes to node w_i and, r_i is equal to 0 for internal walls, and to 1 for peripheral walls.

The temperature of the i-th room is governed by the following equation:

$$\begin{aligned} C_{r_i}\frac{dT_{r_i}}{dt} = &\sum_{j\in\mathcal{N}_{r_i}}\frac{T_j - T_{r_i}}{R'_{ij}} + \dot{m}_{r_i}c_a(T_{s_i} - T_{r_i}) \\ &+ w_i\tau_{w_i}A_{w_i}q''_{rad_i} + \dot{q}_{int_i} \end{aligned} \quad (2)$$

where T_{r_i}, C_{r_i} and $\dot{m}_{r_i}$ are the temperature, heat capacity and air mass flow into the room i, respectively. c_a is the specific heat capacity of air, A_{w_i} is the total area of window on walls surrounding room i, τ_{w_i} is the transmissivity of glass of window i, q''_{rad_i} is the radiative heat flux density radiated to room i, and $\dot{q}_{int_i}$ is the internal heat generation in room i. $\mathcal{N}_{r_i}$ is the set of all of the neighboring nodes to room i. w_i is equal to 0 if none of the walls surrounding room i has window, and is equal to 1 if at least one of them has. The details of building thermal modeling and estimation of the un-modeled dynamics is presented in [2, 6, 5]. Note that we estimate the values of $q''_{rad_i}(t)$ and $\dot{q}_{int}(t)$ based on the following equations.

$$q''_{rad_i}(t) = \tau\hat{T}_{out}(t) + \zeta \quad (3)$$

$$\dot{q}_{int}(t) = \mu\hat{\Psi}(t) + \nu \quad (4)$$

where $\hat{T}_{out}$ and $\hat{\Psi}$ are the (actual) outside air temperature and CO_2 concentration in the room. Parameters τ, ζ, μ and ν are obtained by the parameter estimation algorithm detailed in [5].

The heat transfer equations for each wall and room yields tothe following state space form of the system dynamics

$$\begin{aligned} \dot{x}_t &= f(x_t, u_t, \hat{d}_t) \\ y_t &= Cx_t \end{aligned} \quad (5)$$

where $x_t \in \mathbb{R}^n$ is the state vector representing the temperature of the nodes in the thermal network, $u_t \in \mathbb{R}^{lm}$ is the input vector representing the air mass flow rate and discharge air temperature of conditioned air into each thermal zone, and $y_t \in \mathbb{R}^m$ is the output vector of the system which represents the temperature of the thermal zones. l is the number of inputs to each thermal zone (e.g., air mass flow and supply air temperature). C is a matrix of proper dimension and the disturbance vector is $\hat{d}_t = g(q''_{rad_i}(t), \dot{q}_{int}(t), \hat{T}_{out}(t))$, where g is approximated as a linear function. This leads to

$$\hat{d}_t = aq''_{rad_i}(t) + b\dot{q}_{int}(t) + c\hat{T}_{out}(t) + e \quad (6)$$

By substituting (3) and (4) into (6) and massaging the resulting equation we get

$$\begin{aligned} \hat{d}_t &= (a\tau + c)\hat{T}_{out}(t) + b\mu\hat{\Psi}(t) + a\zeta + b\nu + e \\ &= \bar{a}\hat{T}_{out}(t) + \bar{b}\hat{\Psi}(t) + \bar{e} \end{aligned} \quad (7)$$

where $\bar{a} = a\tau + c$, $\bar{b} = b\mu$, and $\bar{e} = a\zeta + b\nu + e$.

In what follows we use linearized system dynamics for control design. However, the original non-linear model is used for state estimation and filtering and as the plant to compute the actual temperature evolution. System dynamics is linearized around the nearest equilibrium point, by starting from an initial point and searching, using a *Sequential Quadratic Programming (SQP) algorithm*, until it finds the nearest equilibrium point to the specified operating point of the system (details in [2]). Discretizing the state space realization using zero-order hold leads to the following discrete time LTI system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + E\hat{d}_k \\ y_k &= Cx_k \end{aligned} \quad (8)$$

From this equation, we introduce the modeling of sensing and prediction accuracy, as shown in Section 3.

3. SENSING AND PREDICTION ACCURACY MODELING

3.1 Sensing accuracy

In this paper, we focus on temperature sensing accuracy to determine the noise characteristics of the indoor temperature (x_k in (8)), which drives the design of the control algorithms (see Section 4).

Measurement inaccuracies of individual sensors can be categorized into the following three types: no information, completely incorrect information, and incorrect but in-range information. The error of the indoor temperature estimation, denoted by ϵ^{rt}, is affected by accuracy of individual sensors, number and locations of sensors in a thermal zone, and physical properties of the building. Statistics of ϵ^{rt} may be extracted from historical data as shown in Section 5.2, and we assume it is additive to the temperature measurement. Accordingly, the temperature measurement in (8) is updated as $z_k = Cx_k + F\epsilon^{rt}_k$. Where F is a matrix of proper dimension and z_k is the temperature reading at time k.

3.2 Prediction accuracy

Disturbance prediction ($\hat{d}_k$ in (8)) in our model depends on the prediction of CO2 concentration level in the room (Ψ) and on the prediction of ambient air temperature (T_{out}) as shown in (7). We use ϵ^c_k to denote the error in CO2 level prediction, and hence the predicted CO2 level is

$$\Psi(k) = \hat{\Psi}(k) + \epsilon^c_k \quad (9)$$

and ϵ^{ot}_k to denote the error in ambient air temperature prediction, and hence the predicted ambient temperature is

$$T_{out}(k) = \hat{T}_{out}(k) + \epsilon^{ot}_k \quad (10)$$

Based on (7), disturbance prediction can be expressed using CO2 and ambient temperature prediction errors

$$\begin{aligned} \hat{d}_k &= \bar{a}\hat{T}_{out}(k) + \bar{b}\hat{\Psi}(k) + \bar{e} \\ &= \bar{a}(T_{out}(k) - \epsilon^{ot}_k) + \bar{b}(\Psi(k) - \epsilon^c_k) + \bar{e} \\ &= d_k - (\bar{a}\epsilon^{ot}_k + \bar{b}\epsilon^c_k) \end{aligned} \quad (11)$$

where d_k denotes the predicted disturbance at time step k. The above equation suggests that the disturbance prediction error, denoted by w_k, is a linear combination of the CO2 prediction error, ϵ^c_k, and the ambient air temperature

prediction error, ϵ_k^{ot}. This leads to the following state update and output measurement equation (modified from (8))

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + E(d_k - w_k) \\ z_k &= Cx_k + Fv_k \end{aligned} \tag{12}$$

where $w_k = -\xi(\bar{a}\epsilon_k^{ot} + \bar{b}\epsilon_k^c)$ and the constant ξ is a function of the discretization method and step of the continuous system dynamics, and $v_k = \epsilon_k^{rt}$ (i.e. the temperature measurement error). The CO2 and ambient air temperature forecast errors are *uncorrelated* random variables with variance σ_c and σ_{ot}, respectively. Hence, variance of w_k is calculated by

$$\begin{aligned} \sigma_w &= \mathbb{E}[(w - \hat{w})(w - \hat{w})^T] \\ &= \mathbb{E}\left\{[-\xi(\bar{a}\tilde{\epsilon}^{ot} + \bar{b}\tilde{\epsilon}^c)][-\xi(\bar{a}\tilde{\epsilon}^{ot} + \bar{b}\tilde{\epsilon}^c)]^T\right\} \\ &= \xi^2(\bar{a}^2\sigma_c + \bar{b}^2\sigma_{ot}) \end{aligned} \tag{13}$$

where $\tilde{\epsilon}^{ot} = \epsilon^{ot} - \hat{\epsilon}^{ot}$ and $\tilde{\epsilon}^c = \epsilon^c - \hat{\epsilon}^c$. In this paper, we assume typical values for σ_{ot}, σ_c and other constants in simulations: $\sigma_{ot} = 2$, $\sigma_c = 50$, $\bar{a} = 0.01$, $\bar{b} = 0.06$ and $\xi = 10$.

If CO2 and temperature sensors are deployed to facilitate the predictions of CO2 level and ambient air temperature, the values of σ_{ot} and σ_c will be largely affected by the choice of CO2 and temperature sensor type, number and locations.

4. CONTROL ALGORITHMS

4.1 Controller design

In typical buildings, the temperature of air in rooms and in AC ducts is measured. These measurements are normally inaccurate and noisy due to the effects enumerated in Section 3.1. Also temperatures of the *slow dynamic* states, which include walls, ceiling and furniture, are not easy to measure. For simpler controllers such as on-off controllers, just filtering the measurement may be sufficient for control purposes. However, in the case of more sophisticated controllers such as model predictive controllers (MPC), estimation of temperature of all the states is required, including the states whose temperature cannot be measured. In this paper we use Kalman filtering technique to estimate and filter the states of system using available measurements. To the best of our knowledge, the effectiveness of Kalman filtering for *state* estimation and control applications in the presence of noisy data in buildings has not been explored. We consider three controllers: on-off[2], MPC and RMPC.

4.1.1 On-off controller

The on-off controller is designed in a way that the valves can have three states: fully opened, minimally opened[3] during occupied hours, or fully closed at night. The duration of each state of the valve cannot be less than 1 hour, in order to be consistent with the actuation time step of other controllers. The controller turns on the heating mode when the room temperature falls below the lower limit and turns it to either minimally open or fully closed (depending on whether it is occupied or unoccupied hours of the day) when the temperature is within the comfort zone, and turns on the cooling mode if the temperature goes above the higher limit.

[2]Other controllers such as PI could also be used here.

[3]Not fully closed due to ventilation and air quality reasons.

4.1.2 MPC formulation

A model predictive control problem is formulated with the objective of minimizing a linear combination of total energy consumption and peak airflow. The fan energy consumption is proportional to the *cube* of the airflow, hence minimizing the peak airflow would dramatically reduce fan energy consumption. We consider a cost function for the MPC which includes a linear combination of the total heating flow into the building in the form of hot air (given by l_1-norm of input) and the peak of airflow (given by l_∞-norm of input)[4]. To guarantee feasibility (constraint satisfaction) at all times we use soft constraints. The predictive controller solves at each time step the following problem

$$\begin{aligned} \min_{U_t, \overline{\Theta}_t, \underline{\Theta}_t} \quad & \{|U_t|_1 + \kappa|U_t|_\infty + \rho(|\overline{\Theta}_t|_1 + |\underline{\Theta}_t|_1)\} = \\ \text{s.t.} \quad & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} + Ed_{t+k|t} \\ & y_{t+k|t} = Cx_{t+k|t} \\ & \underline{\mathcal{U}}_{t+k|t} \le u_{t+k|t} \le \overline{\mathcal{U}}_{t+k|t} \\ & \underline{T}_{t+k|t} - \underline{\theta}_{t+k|t} \le y_{t+k|t} \le \overline{T}_{t+k|t} + \overline{\theta}_{t+k|t} \\ & \underline{\theta}_{t+k|t}, \overline{\theta}_{t+k|t} \ge 0 \end{aligned} \tag{14}$$

where $U_t = [u_{t|t}, u_{t+1|t}, \cdots, u_{t+N-1|t}]$ stores the control inputs, $\underline{\Theta}_t = [\underline{\theta}_{t+1|t}, \cdots, \underline{\theta}_{t+N|t}]$ and $\overline{\Theta}_t = [\overline{\theta}_{t+1|t}, \cdots, \overline{\theta}_{t+N|t}]$ are the temperature violations from the lower and upper bounds, respectively. $y_{t+k|t}$ is the thermal zone temperature, and $d_{t+k|t}$ is the disturbance load prediction. $\underline{T}_{t+k|t}$ and $\overline{T}_{t+k|t}$ are the lower and upper bounds on the zone temperature, respectively. $\underline{\mathcal{U}}_{t+k|t}$ and $\overline{\mathcal{U}}_{t+k|t}$ are the lower and upper limit on airflow input by the VAV damper, respectively. Note that based on American Society of Heating Refrigeration and Air conditioning Engineers (ASHRAE) requirements for *air change per hour* (ACH) of rooms, there has to be a minimum *non-zero* airflow during occupied hours for ventilation purposes. ρ is the penalty on the comfort constraint violations, and κ is the penalty on peak power consumption. We use $\rho = 1000$ and $\kappa = 15$, and prediction horizon of N=24 for simulations.

4.1.3 RMPC formulation

A typical robust strategy involves solving a min-max problem to optimize worst-case performance, while enforcing input and state constraints for all possible disturbances. In particular, we formulate the RMPC to minimize energy consumption of the HVAC system and satisfy the temperature and input constraints against additive bounded input uncertainties. Define the worst-case cost function as

$$\begin{aligned} & J_t(x(t), U_t) \triangleq \\ & \max_{w_{[.]}} \{||Pz_{t+N|t}||_p + \sum_{k=0}^{N-1} ||Qz_{t+k|t}||_p + ||Ru_{t+k|t}||_p\} \\ \text{s.t.} \quad & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} + E(d_{t+k|t} - w_{t+k|t}) \\ & z_{t+k|t} = Cx_{t+k|t} + Fv_{t+k|t} \\ & w_{t+k|t} \in \mathcal{W} \quad \& \quad v_{t+k|t} \in \mathcal{V} \\ & \forall \quad k = 0, 1, \cdots, N-1 \end{aligned} \tag{15}$$

[4]Note that using l_1 and l_∞ norms in the cost function, leads to an LP which can be solved very efficiently for even very large dimensions.

where $||.||_p$ can be any polytopic norm. The robust optimal control problem is formulated as follows

$$\begin{aligned} J_t^*(x(t)) \triangleq \quad & \min_{U_t} \quad J_t(x(t), U_t) \\ \text{s.t.} \quad & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} + Ed_{t+k|t} \\ & z_{t+k|t} = Cx_{t+k|t} + Fv_{t+k|t} \\ & u_{t+k|t} \in \mathcal{U} \quad \& \quad x_{t+k|t} \in \mathcal{X} \\ & \forall \ w_{t+k|t} \in \mathcal{W} \quad \& \quad \forall \ v_{t+k|t} \in \mathcal{V} \\ & \forall \quad k = 0, 1, \cdots, N-1 \end{aligned} \tag{16}$$

At each time step t, the first entry of U_t is implemented on the plant. At the next time step the prediction horizon N is shifted leading to a new optimization problem. This process is repeated until the time span of interest is covered.

We utilize Closed-Loop (CL) formulation, which is less conservative than the Open-Loop (OL) [8]. Performance improvement (less conservativeness) is gained due to the fact that in CL, future measurements are taken into account for calculating the control policy. We use the feedback predictions to approximate the solution to the CL problem. We utilize the TLDS parameterization introduced in [8].

4.1.4 Extended Kalman Filter

Due to the nonlinearity of the state update equations for the HVAC system, we use the Extended Kalman Filter (EKF)[5]. Note that EKF can be used due to the differentiability of the state update model.

The KF design requires state-space model of the system and two design parameters: the covariances of the *process noise* and the *measurement noise*, which in our problem correspond to the covariances of the prediction error w_k and the temperature measurement error v_k.

EKF formulation.

In EKF, the state transition and observation models need not be linear functions, but may instead be differentiable functions. the stochastic state-space model is given by

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, d_{k-1}, w_{k-1}) \\ z_k &= h(x_k) + v_k \end{aligned} \tag{17}$$

where w_k and v_k are the process and measurement noise which are assumed to be zero mean, independent and identically distributed (IID) multivariate Gaussian with covariance W_k and V_k respectively (i.e. $w_k \sim \mathcal{N}(0, W_k)$ and $v_k \sim \mathcal{N}(0, V_k)$).

Function f is used to compute the predicted state from the previous estimate and function h is used to compute predicted measurement from predicted state. However, nonlinear functions f and h cannot be applied to the covariance directly. Instead the Jacobian of these functions which is a matrix of partial derivatives is computed.

EKF algorithm.

The KF consists of two steps – *prediction* followed by *update*. In the prediction step, the filter propagates the estimate from a previous time step $k-1$ to the current time step k. The EKF algorithm is as follows. At each time step, first the *a-priori* state estimate is predicted

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}, d_{k-1}, 0) \tag{18}$$

and then state transition and observation matrices are computed using the *a-priori* state estimation

$$F_{k-1} = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1|k-1}, u_{k-1}} \qquad H_k = \frac{\partial h}{\partial x}|_{\hat{x}_{k|k-1}} \tag{19}$$

then *a-priori* state estimation error covariance is predicted

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + W_{k-1} \tag{20}$$

and also the *a-priori* output estimation error (i.e. innovation or measurement residual) is computed

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \tag{21}$$

innovation or residual covariance is computed using the *a-priori* state estimation error covariance given by (20)

$$S_k = H_k P_{k|k-1} H_k^T + V_k \tag{22}$$

the near-optimal Kalman gain is obtained by using the value of S_k

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{23}$$

and finally *a-posteriori* state estimate and the *a-posteriori* state estimation error covariance are updated using

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{24}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{25}$$

Note that matrix inversion can be computationally expensive for large matrices. Therefore, in order to make the computations more efficient, from both an execution time and numerical accuracy standpoint, we utilize the *back substitution* technique with the help of the *Cholesky factorization* and perform the following operations (eqs. (26) to (29)) substituting the steps explained in eqs. (23) and (24). This method solves the problem using Gaussian elimination, without forming the inverse of the matrix.

$$R_k = chol(S_k) \tag{26}$$

$$U_k = P_{k|k-1} H_k^T / V_k \tag{27}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + U_k (V_k^T \backslash \tilde{y}_k) \tag{28}$$

$$P_{k|k} = P_{k|k-1} - U_k U_k^T \tag{29}$$

where the $chol()$ operator performs the Cholesky factorization of its argument resulting in an upper triangular matrix. / and \ perform right and left matrix division, respectively.

Once the state is estimated and filtered, the state estimate $\hat{x}_{k|k}$ is used in the control design, for all the cases of On-off, MPC and RMPC. Then the control action u_k is computed based on this estimate. At the next time step, $\hat{x}_{k|k}$ and u_k are fed to the EKF algorithm to estimate the state at the next time step, and this process is repeated over.

Simulation results.

Fig. 2 illustrates the simulated room temperature and air flow inputs of the MPC with EKF over one day period. The prediction error and the temperature measurement error are assumed to be $||w||_\infty = 0.9$ and $||v||_\infty = 0.8$, respectively.

[5]The extended Kalman filter (EKF) is probably the most widely used estimation algorithm for nonlinear systems [11].

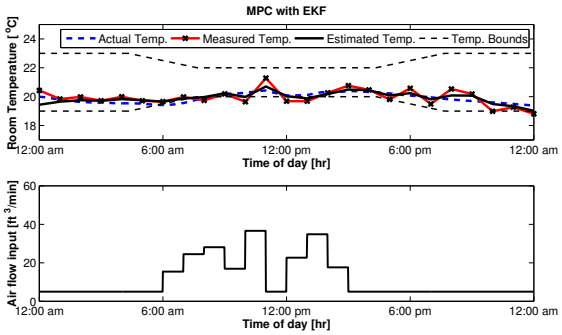

Figure 2: MPC with EKF for estimation and filtering. Performance of EKF-MPC over a range of temperature measurement errors are presented in Fig. 4 and Fig. 5.

4.1.5 *Unscented Kalman Filter*

A nonlinear KF that shows promise as an improvement over the EKF is the unscented Kalman filter (UKF). The basic premise behind the UKF is that it is easier to approximate a Gaussian distribution than to approximate an arbitrary nonlinear function. The UKF addresses the approximation issues of the EKF. Instead of using Jacobian matrix, UKF uses a deterministic sampling approach to capture the mean and covariance estimates with a minimal set of sample points [12]. As with the EKF, we present an algorithmic description of the UKF, omitting some theoretical considerations. More details can be found in [13, 14].

UKF formulation.

The state distribution is represented by a Gaussian random variable (GRV), but is now specified using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the GRV, and when propagating through the true nonlinear system, capture the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any nonlinearity. To elaborate on this, we start by first explaining the Unscented Transformation(UT). The UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation[13].

UKF algorithm.

First, we conduct the following initialization

$$\hat{x}_0 = \mathbb{E}[x_0] \tag{30}$$

$$P_0 = \mathbb{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \tag{31}$$

Given the state vector at step $k-1$, we then compute a collection of sigma points, stored in the columns of the $L \times (2L+1)$ sigma point matrix $\mathcal{X}_{k-1}$, where L is the dimension of the state vector.

$$\mathcal{X}_{k-1} = [\hat{x}_{k-1} \quad \hat{x}_{k-1} + \gamma\sqrt{P_{k-1}} \quad \hat{x}_{k-1} - \gamma\sqrt{P_{k-1}}] \tag{32}$$

where $\gamma = \sqrt{(L+\lambda)}$, and $\lambda = \alpha^2(L+\delta) - L$ is the composite scaling parameter. α is a scaling parameter that determines the spread of the sigma points around $\hat{x}$, and is usually set to a small positive value (e.g. $1e-4 \leq \alpha \leq 1$). δ is a secondary scaling parameter which is usually set to 0 or $3-L$ [14]. Then we propagate each column of $\mathcal{X}_{k-1}$ through time by Δt using the system dynamics, i.e. $(\mathcal{X}_k)_i = f((\mathcal{X}_{k-1})_i)$ for $i = 0, 1, ..., 2L$, where f is given by (5). Having calculated $\mathcal{X}_i$, the *a-priori* state estimate is given by

$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} (\mathcal{X}_k)_i \tag{33}$$

where $W_i^{(m)}$ are weights defined by

$$W_i^{(m)} = \begin{cases} \frac{\lambda}{(L+\lambda)}, & \text{if } i = 0 \\ \frac{1}{2(L+\lambda)}, & \text{if } i = 1, 2, ..., 2L \end{cases}$$

and the *a-priori* error covariance is calculated by

$$P_k^- = \sum_{i=0}^{2L} W_i^c [(\mathcal{X}_k)_i - \hat{x}_k^-][(\mathcal{X}_k)_i - \hat{x}_k^-]^T + Q_k \tag{34}$$

where Q_k is the process error covariance matrix. The $W_i^{(c)}$ weights are defined by

$$W_i^{(c)} = \begin{cases} \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta), & \text{if } i = 0 \\ \frac{1}{2(L+\lambda)}, & \text{if } i = 1, 2, ..., 2L \end{cases}$$

where β is a parameter used to incorporate the prior knowledge of the distribution of x. We use $\beta = 2$ which is optimal for Gaussian distributions [15].

In the *update* step, we fist transform the columns of $\mathcal{X}_k$ through the measurement function. Hence

$$(\mathcal{Z}_k)_i = h((\mathcal{X}_k)_i) \qquad i = 0, .., 2L \tag{35}$$

$$\hat{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} (\mathcal{Z}_k)_i \tag{36}$$

then we compute the *a-posteriori* state estimate using $\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{z}_k^-)$ where K_k is the Kalman gain defined by

$$K_k = P_{\hat{x}_k \hat{z}_k} P_{\hat{z}_k \hat{z}_k}^{-1} \tag{37}$$

$$P_{\hat{x}_k \hat{z}_k} = W_i^c [(\mathcal{X}_k)_i - \hat{x}_k^-][(\mathcal{Z}_k)_i - \hat{z}_k^-]^T \tag{38}$$

$$P_{\hat{z}_k \hat{z}_k} = \sum_{i=0}^{2L} W_i^c [(\mathcal{Z}_k)_i - \hat{z}_k^-][(\mathcal{Z}_k)_i - \hat{z}_k^-]^T + R_k \tag{39}$$

where R_k is the measurement noise covariance matrix. The *a-posteriori* estimate of the error covariance is given by

$$P_k = P_k^- - K_k P_{\hat{z}_k \hat{z}_k} K_k^T \tag{40}$$

Simulation results.

Fig. 3 illustrates the simulated room temperature and air flow inputs of the MPC with UKF over one day period. The prediction error and the temperature measurement error are assumed to be $||w||_\infty = 0.9$ and $||v||_\infty = 0.8$, respectively.

4.2 Control cost and performance vs. sensing accuracy

For each controller designed in Section 4.1, we simulate the energy cost and performance (measured by a discomfort index)[6] with respect to different levels of sensing accuracy.

Fig. 4 and Fig. 5 show the energy cost and discomfort index of the six controllers with either EKF or UKF under

[6]For detailed discussion on *energy cost* and *discomfort index*, refer to [5].

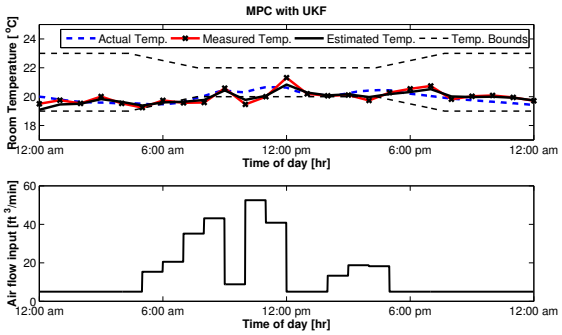

Figure 3: MPC with UKF for estimation and filtering. Performance of UKF-MPC over a range of temperature measurement errors are presented in Fig. 4 and Fig. 5.

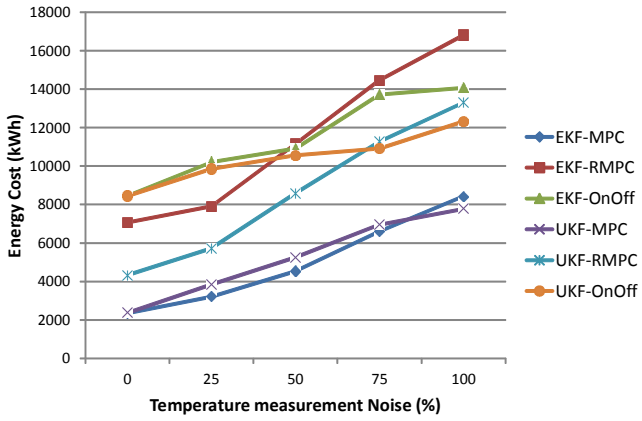

Figure 4: Energy cost vs. measurement noise

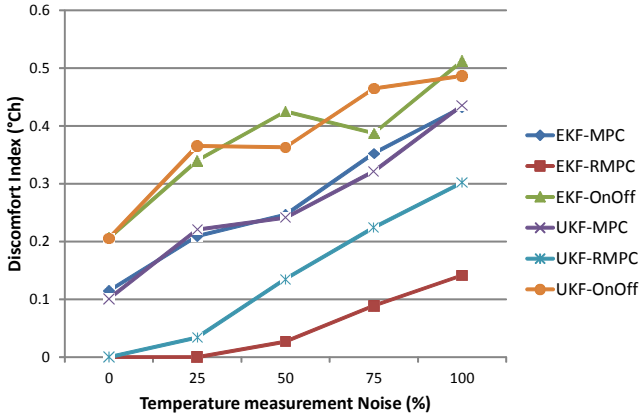

Figure 5: Discomfort index vs. measurement noise

different levels of sensing accuracy from 0% to 100%. Note that we use a normalized uncertainty based on the maximum value of the disturbance. For instance, measurement error of 50% corresponds to $||w||_\infty = 0.5 * ||\hat{d}||_\infty$. More details can be found in [8].

From these results, we can see that some controllers are always superior to some other controllers in terms of both energy cost and discomfort index. For instance, the two MPC

Figure 6: BubbleZERO test-bed

controllers are better than the two OnOff controllers. However, in other cases, particularly among MPC and RMPC controllers, choosing which controller to use depends on *the design requirements and the measurement noise and error (which is affected by the embedded platform)*. For instance, let us assume there is a requirement that the maximum discomfort level should be under 0.3. If we choose a number of accurate sensors and we estimate that noise will be under 25%, then we select EKF-MPC since it consumes the least amount of energy while satisfying the requirement. If we choose less accurate (or fewer) sensors to reduce platform cost and we estimate that noise will be around 75% , then we select UKF-RMPC since it consumes the least energy among the schemes that satisfy the comfort level requirement (the other being EKF-RMPC).

5. SENSING SYSTEM DESIGN AND SENSING ACCURACY

The relation between sensing system design and temperature sensing accuracy, i.e. the temperature measurement error v_k in (12) is quite important in the overall performance of the HVAC system. In this paper, we collect measurements from multiple temperature sensors in a test-bed and analyze their statistics to estimate sensing accuracy under different number of sensors and locations. This approach may be applied in practice if the designer has access to the target building (or buildings/testbeds with similar characteristics) and can deploy sensors for testing, otherwise simulation approaches such as computational fluid dynamics (CFD) analysis may be applied.

5.1 BubbleZERO test-bed setup

Our test-bed BubbleZERO (shown in Fig. 6) is an experimental building and laboratory, and is conceived as part of the Low Exergy Module development for Future Cities Laboratory (FCL) [16]. The BubbleZERO was constructed by ETH Zurich with concrete floor and ceiling as well as LowEx systems installed along with an experimental chiller. It is currently installed on the NUS campus in Singapore. It provides extensive opportunities to test and evaluate the performance of sensing and control systems as well as low energy systems.

Deployed instruments with a network of wireless sensors

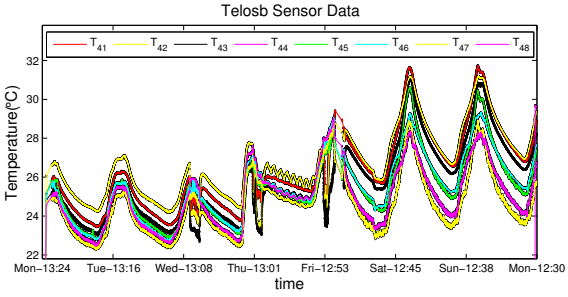

Figure 7: Temperature readings from 8 sensors

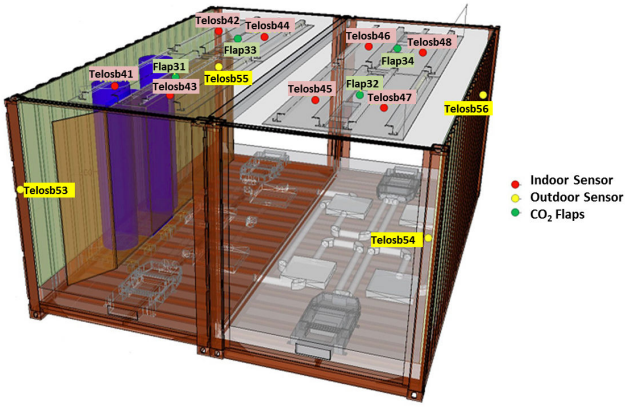

Figure 8: Sensor deployment in the test-bed

provides instant and interactive information on multiple dimensions of the building environment. The use of wireless sensor network systems is well suited for future building operation with its low-cost, low energy, easy-to-deploy, flexibility, and closed-loop sensing and control features. Fig. 8 shows the system structure. The system is built based on the open source wireless sensor network platform Telosb [17], and the software is developed based on the TinyOS [18] operating system. Each sensor node is powered with an embedded CPU, equipped with a wireless communication module for intercommunication, and various interfaces and sensors.

The system contains 4 sub-systems as shown in Fig. 8. The Environment Sensing System senses the indoor and outdoor ambient parameters. Temperature and humidity are monitored for both indoor and outdoor, while CO2 concentration is collected for indoor. The environment sense system includes 8 indoor sensors, 4 CO2 concentration sensors and 4 outdoor sensors. The 8 indoor sensors (Telosb41-Telosb48) are hanged at the ceiling panels. They measure temperature and humidity at different locations in the room. The 4 CO2 flaps (CO2flap31-CO2flap34) are installed at the ceiling to measure the indoor CO2 concentration and also to control the air exhaust for the room.

5.2 Analysis of historical data

We collected the data from the test-bed for one week period (July 16-23, 2012). Fig. 7 shows sensor readings from the 8 indoor temperature sensors located in the test-bed.

As shown in Fig. 8, the 8 sensors are spread out to cover the entire area of BubbleZERO. The spatial sensor location variation leads to different levels of measurement accuracy from each sensor with respect to the average temperature in the space. To obtain the relation between sensing accuracy and choice of number and locations of sensors, we analyze the data collected from the test-bed. Note that sensor Telosb44 has only stored a few data points due to faulty behavior; consequently, we neglect that sensor in our analysis.

We consider the average of all the remaining 7 sensors as the actual temperature of the room. We then select a different number of sensors and calculate the *difference* between the average measure of the selected k sensors ($k = 1, 2, ..., 6$) and the average measure of all 7 sensors (i.e. the actual temperature of the room). This difference provides an estimation of the temperature sensing accuracy under certain selection of the sensors, and can be regarded as the measurement error of this set of sensors. The root mean square (rms) of this difference is denoted by Δ_{rms}.

To further study the effect of sensor locations on sensing accuracy, for each k value, we enumerate all possible sets of sensors (with different locations) and pick the set that provides the minimal rms value of its measurement error, which is denoted by δ_m^b. We also calculate the measurement error of a randomly selected set (assuming equal probability for each sensor) and denote it by δ_m^r. The results of this analysis is shown in Fig. 9. For instance, when $k = 1$, the best sensor (i.e. the one that provides the minimal rms value with respect to the average of all 7 sensors) is sensor T_{45}, which is located in the south eastern part of the bubble and provides $\Delta_{rms} = 0.29$. When $k = 2$, the best two sensors are T_{41} and T_{48}, which are the two sensors located in the two opposite corners, southwest and northeast of the bubble and leads to $\Delta_{rms} = 0.18$. In Fig. 9, we also use the normal distribution to approximate the measurement error and calculate the corresponding mean and variance. The variance will then be used in the selection of control algorithms.

As mentioned in Section 3.2, in addition to the measurement error, the prediction error may also depend on the design of the sensing system, if CO2 sensors and outdoor temperature sensors are deployed to facilitate the prediction. The BubbleZERO test-bed provides CO2 and outdoor sensors that may enable this study, although currently we do not have enough functional CO2 and outdoor sensors (e.g. only two CO2 sensors have readings available). We plan to conduct such study in the future.

6. EXPLORATION OF CONTROL ALGORITHM AND SENSING SYSTEM

Based on the results presented above, we explore the design space for both control algorithm and sensing system to build a Pareto front of optimal energy and monetary cost under the constraint on user comfort level.

Specifically, we first choose the number of sensors based on a given monetary budget. We assume the best sensor locations are known to the designers (possibly through testing different location combinations) and decide the sensing data accuracy according to the statistics in Fig. 9, i.e. the measurement error δ_m^b. Then by setting the measurement error v_k in (12) to σ_m^b, we conduct simulations for all six controllers and choose the one that minimizes the energy cost and satisfy the constraint on user comfort level as measured by the discomfort index. Results are shown in Fig. 10.

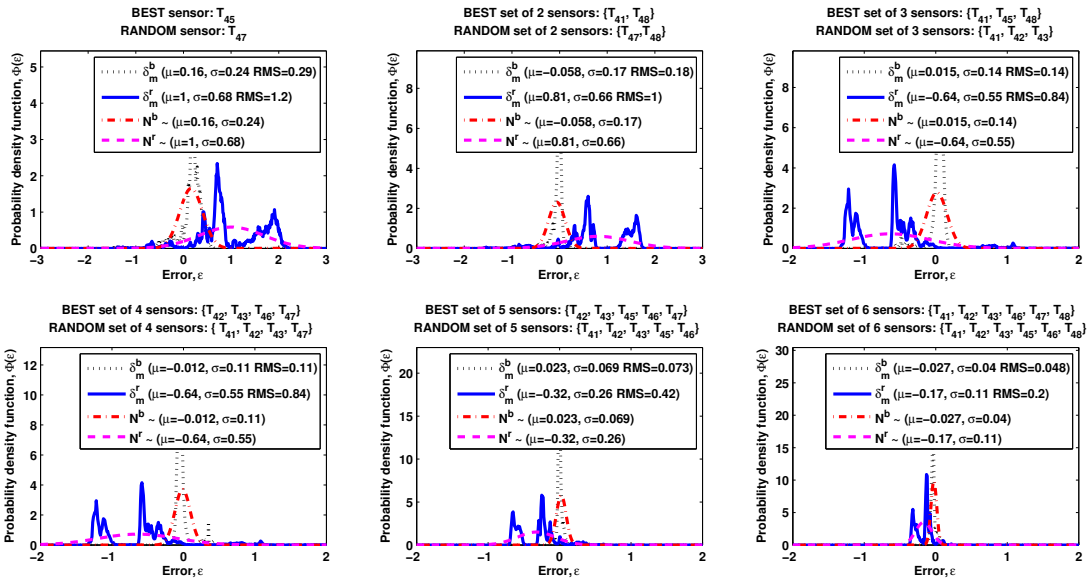

Figure 9: Average error of k sensors for the minimal error set of sensors and a random choose of sensors. Each figure lists in its title the best and random set of k sensors for k=1,2,...,6, and plots the pdf of its measurement error (i.e. the difference between the average of k sensor readings with the average of all 7 sensor readings), denoted by δ_m^b and δ_m^r, respectively. The best and random set of sensors are selected based on their resulting Δ_{rms}. N^b and N^r represent the normal distribution.

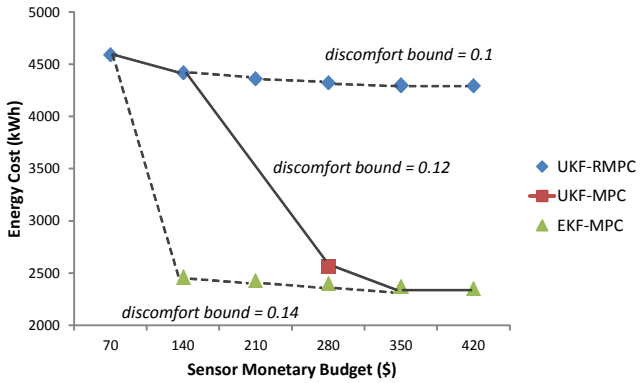

Figure 10: Pareto front under comfort constraints with best sensor locations

If it is required that the discomfort index be less than 0.12, the *Pareto front consists of different control algorithm choices depending on the sensor monetary budget*. In our platform, each Telosb unit costs about $70. When the budget is set to be under $140, we may choose 1 or 2 sensors, in which case the control algorithm that can satisfy the comfort level constraint and provide the minimal energy cost is UKF-RMPC (with energy cost around 4500). When the budget is set be more than 350, we may choose 5 or 6 sensors, in which case the control algorithm that has the minimal energy cost and satisfy the comfort constraint is EKF-MPC (with energy cost under 2500). When the budget is around 280, we may choose 4 sensors, in which case the best control algorithm is UKF-MPC.

Intuitively, when we have a small budget, the sensing data accuracy is lower and we need a more robust algorithm to satisfy the comfort level constraint; hence the RMPC controller (the MPC controllers do not satisfy the constraint in this case). When we have a large budget, the sensing accuracy is higher and we may choose the more energy-efficient algorithms, hence the MPC controllers. If the discomfort index is required to be less than 0.1, we will choose the UKF-RMPC controller under any budget, since the other controllers either do not satisfy the comfort constraint (i.e. EKF-MPC, UKF-MPC, OnOff controllers), or cost more energy (i.e. EKF-RMPC). If the discomfort index is set to be 0.14, we will choose EKF-MPC if there is budget for more than 1 sensor.

If the sensor locations are selected randomly (the theoretical best locations might not be known or accessible in practice), the energy cost and performance of each controller under certain budget are different from the best location case. Fig. 11 shows the exploration results with random sensor locations. We can see that similar to the best location case, *the selection of control algorithms depends on the number of sensors*. For instance, the solid line represents the case where the discomfort index is required to be less than 0.2. When the budget is under 280, we may select as most 4 sensors and the best control algorithm is UKF-RMPC (other controllers do not satisfy the comfort constraints except for EKF-RMPC but it has higher energy cost). When the budget is more than 350, the best control algorithm is EKF-MPC. If the discomfort index is required to be less than 0.1, the best control algorithm is always UKF-RMPC. Note that the energy cost under any number of sensors is also more than the energy cost in the best location case with the same number of sensors, which is to be expected.

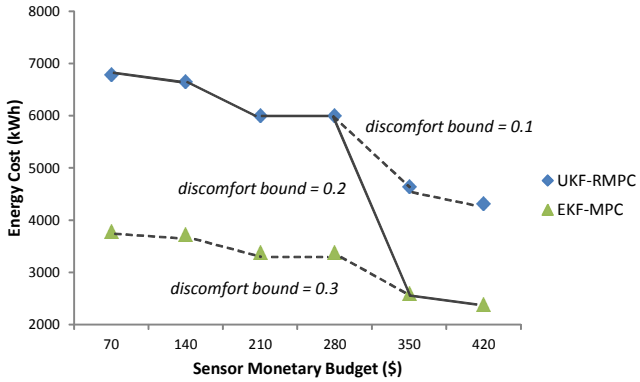

Figure 11: Pareto front under comfort constraints with random sensor locations

7. CONCLUSION

We proposed a co-design framework for HVAC systems to explore control algorithm design and sensing platform selection concurrently by analyzing their inter-dependencies. We designed six control algorithms (OnOff, MPC and RMPC controllers with either EKF or UKF filtering), to effectively address sensing and prediction errors. We also analyzed the relation between sensing accuracy and the number and locations of sensors, using the collected data from a well-instrumented test-bed. Based on these models and analysis, we explored the design space of both control algorithm and sensing platform and generated Pareto fronts with optimal energy and monetary cost.

In the future, we plan to study the inter-dependencies between the HVAC control algorithm and the embedded platform. In particular, we will analyze the relation between the prediction error and the design of the embedded platform (e.g. the choice of the CO2 and outdoor sensors), and leverage the findings in co-design. We will broaden our consideration of the embedded platform design from the sensing system to the computation and communication components, such as the impact of communication reliability on the control algorithm.

8. ACKNOWLEDGEMENT

This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

9. REFERENCES

[1] (2012, Feb.) Building energy data book of doe, website: http://buildingsdatabook.eren.doe.gov.

[2] M. Maasoumy, "Modeling and optimal control algorithm design for hvac systems in energy efficient buildings," Master's thesis, University of California, Berkeley, 2011. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-12.html

[3] A. D. Y. Ma, A. Kelman and F. Borrelli, "Model predictive control of thermal energy storage in building cooling systems," *IEEE Control System Magzine*, pp. 1–65, 2011.

[4] F. Oldewurtel, A. Parisio, C. 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 *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 5100–5105.

[5] M. Maasoumy and A. Sangiovanni-Vincentelli, "Total and peak energy consumption minimization of building hvac systems using model predictive control," *IEEE Design and Test of Computers*, Jul-Aug 2012.

[6] M. Maasoumy, A. Pinto, and A. Sangiovanni-Vincentelli, "Model-based hierarchical optimal control design for HVAC systems," in *Dynamic System Control Conference (DSCC)*, 2011.

[7] P. Radecki and B. Hencey, "Online building thermal parameter estimation via unscented kalman filtering," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3056–3062.

[8] M. Maasoumy and A. Sangiovanni-Vincentelli, "Optimal control of HVAC systems in the presence of imperfect predictions," in *Dynamic System Control Conference (DSCC)*. ASME, 2012.

[9] Y. Yang, A. Pinto, A. Sangiovanni-Vincentelli, and Q. Zhu, "A design flow for building automation and control systems," in *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*. IEEE, 2010, pp. 105–116.

[10] Y. Yang, Q. Zhu, M. Maasoumy, and A. Sangiovanni-Vincentelli, "Development of building automation and control systems," *Design Test of Computers, IEEE*, vol. 29, no. 4, pp. 45 –55, aug. 2012.

[11] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[12] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, 2000, pp. 153–158.

[13] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, vol. 3. Spie Bellingham, WA, 1997, p. 26.

[14] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, 1995. Proceedings of the*, vol. 3. IEEE, 1995, pp. 1628–1632.

[15] E. Wan and R. Van Der Merwe, "The unscented kalman filter," *Kalman filtering and neural networks*, pp. 221–280, 2001.

[16] Future Cities Lab. http://www.futurecities.ethz.ch/.

[17] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 2005, pp. 364–369.

[18] P. Levis, S. Madden, J. Polastre, R. Szewczyk, E. Brewer *et al.*, "Tinyos: An operating system for sensor networks," *Ambient intelligence*, vol. 35, 2005.