Occupancy Modeling and Prediction for Building Energy Management

VARICK L. ERICKSON, University of California, Merced
MIGUEL Á. CARREIRA-PERPIÑÁN, University of California, Merced
ALBERTO E. CERPA, University of California, Merced

Heating, cooling and ventilation accounts for 35% energy usage in the United States. Currently, most modern buildings still condition rooms assuming maximum occupancy rather than actual usage. As a result, rooms are often over-conditioned needlessly. Thus, in order to achieve efficient conditioning, we require knowledge of occupancy. This article shows how real time occupancy data from a wireless sensor network can be used to create occupancy models, which in turn can be integrated into building conditioning system for usage-based demand control conditioning strategies. Using strategies based on sensor network occupancy model predictions, we show that it is possible to achieve 42% annual energy savings while still maintaining American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) comfort standards.

Categories and Subject Descriptors: I.6.5 [**Simulation and Modeling**]: Model Development; J.7 [**Computers in Other Systems**]: *Command and control*

General Terms: Algorithms, Measurement

Additional Key Words and Phrases: Occupancy, HVAC, ventilation, energy savings, demand response, machine learning

ACM Reference Format:
Varick L. Erickson, Miguel Á. Carreira-Perpiñán, and Alberto E. Cerpa. 2014. Occupancy modeling and prediction for building energy management. ACM Trans. Sensor Netw. 10, 3, Article 42 (April 2014), 28 pages.
DOI: http://dx.doi.org/10.1145/2594771

1. INTRODUCTION

In 2006, approximately 35% of the energy in the United States was used for heating, ventilation, and air-conditioning (HVAC) systems [EIA 2010]. Studies suggest that 15% to 25% of HVAC energy can be saved by setting ventilation rates based on actual occupancy [Brandemuehl and Braun 1999]. Currently, the majority of HVAC systems condition rooms assuming maximum occupancy during normal working hours and are turned off at night. This leads to inefficiency as rooms are often conditioned to levels that are not appropriate for the number of occupants actually occupying the areas. An HVAC could waste energy supplying ventilation enough for 30 people when only 10 actually occupy a room.

This work is based on two previous manuscripts; “OBSERVE: Occupancy-based system for efficient reduction of HVAC energy,” in *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN 2011)* and “Occupancy based demand response HVAC control strategy,” in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building (BuildSys 2010)*.
This material is based upon work partially supported by the National Science Foundation under grants #0923586 and #1254192, the California Institute for Energy and Environment under grant #MUC-09-03, and the Center for Information Technology Research in the Interest of Society under grant #442130-19900.
Authors’ address: University of California, Merced, 5200 North Lake Road, Merced, CA 95343; email: {verickson; mcarreira-perpinan; acerpa}@umerced.edu.

DOI: http://dx.doi.org/10.1145/2594771

To increase the efficiency of HVAC systems, a system for detecting occupancy is needed to condition rooms appropriate to usage. Though there are several methods that are commonly used for detecting occupancy within modern buildings, these methods present limitations. Passive infrared (PIR) sensors, commonly used for controlling lighting, are a simple way of detecting if a room is occupied or not, but do not give information regarding how many people occupy the room. This information is necessary for CO_2 ventilation. Using CO_2 sensors directly for regulating CO_2 is also unsuitable for conditioning strategies. CO_2 buildup is slow, and by the time sensors detect high levels of CO_2 that trigger ventilation, occupants of the room are likely to already feel uncomfortable [Fisk et al. 2006]. If not properly calibrated, these sensors can also be inaccurate [Fisk et al. 2006]. Electrical loads have also been used for occupancy estimation [Abushakra and Claridge 2001]. However, this method assumes each occupant contributes to the load and requires accurate occupancy data for calibration. While calendar data would be useful for regularly scheduled situations such as classrooms, there are many feasibility issues with deriving occupancy data from private calendars. In order for a calendar system to work, it must include location information and assumes that a person is always where the calendar specifies. There many situations where these assumptions do not hold. Many people do not include locations for the calendar events. People can skip meetings or hold unscheduled meetings.

For demand-response HVAC control, occupancy detection needs to be accurate, reliable, and able to capture occupancy changes in real time. The Smart Camera Occupancy Position Estimation System (SCOPES) [Kamthe et al. 2009] developed at UC Merced is a 16-node sensor network of cameras that captures occupancy changes among areas with approximately 80% accuracy in near real time. With newer, more powerful hardware, it is likely the accuracy will be even greater.

Though the ability to adjust an HVAC system based on real-time occupancy is an important step toward greater efficiency, perhaps just as important is the ability to anticipate room usage based on current room usage. This needs to be addressed since conditioning a room is not instantaneous and requires time for adjustments. For example, if it is known that a large number of people are in a lobby area, we want the HVAC system to know an adjacent conference room will be used with high probability and begin conditioning the room beforehand.

The following are the key contributions of our work:

(1) Our work on occupancy modeling uses inter-room relationships over time to model occupancy. In addition, our models are developed using accurate real-world data rather than occupancy walk-through surveys or other similar estimates of occupancy.
(2) We demonstrate how models developed from sensor network data can be integrated with an HVAC control strategy to achieve significant energy savings.
(3) We show that we can still satisfy ASHRAE conditioning standards while saving significant energy. In certain cases, we show that our predictive strategies meet user demand better than typical baseline strategies.
(4) We examine a PIR-based WSN solution and show that binary measurement of occupancy is not always sufficient for HVAC control. Under some circumstances, PIR-based HVAC control performs worse than current strategies; in order to achieve maximum energy savings within a building, accurate real time levels of occupancy must be incorporated into HVAC strategies.

The purpose of this article is to examine several approaches to modeling occupancy that can be integrated into a system such as SCOPES and how these models can be utilized for a predictive HVAC control strategy. We begin in Section 2 by describing a representative occupancy measurement system that would be suitable for our occupancy models and the data collection process. We discuss several previously developed models in Section 3. Section 4 describes the basic framework required for a Markov Chain (MC) occupancy models. Based on this framework, three different MC-based models are developed. An evaluation of three MC models is done Section 5. In Section 6, we define the criteria for HVAC conditioning and propose a control strategy that utilizes the MC model that performed best from Section 5. We test the strategy using an EnergyPlus building simulation and evaluate the energy savings and performance in Section 8. We describe other related work in Section 9 and finally conclude with Section 10.

2. SENSING AND DATA COLLECTION

In our work, we use the SCOPES system developed by Kamthe et al. [2009]. SCOPES is a system comprising 16 sensor nodes on the ceiling of the corridors of a university building. Nodes were deployed at transition boundaries within the hallways of the building. Each transition boundary is comprised of a group of three nodes. Each node is a Cyclops camera interfaced with a Moteiv Tmote Sky module via an intermediate adapter board. Both pieces of hardware run TinyOs as the operating system. The Cyclops is a low-power camera with an on-board 4MHz ATmega128L microcontroller (MCU) and 512KB of external SRAM. The external SRAM is divided into eight, 64KB memory banks to overcome the limitations of the addressable memory. Each bank is capable of storing 80 64×64 pixel grayscale images. At every transition boundary location, three nodes sense the same area where the nodes take turns capturing the sensed area. Whenever a person crosses any one of these transition points, the cameras capture and process the image data and determine if a transition occurred and the direction.

Since the computational resources of the nodes are limited, only lightweight image-processing algorithms are used. Object detection is achieved by background subtraction, and the background is updated continuously. When an object is detected, the pixels of the image are classified as an object, shadow, or background depending on a predetermined threshold. All pixels classified as objects are grouped using a connected component algorithm creating a blob. The centroid and pixel count of the blob is determined and the direction is inferred by consecutive images. Once a mote has processed all images in its bank, object information is transfered to a base station using multihop communication. We found SCOPES was able to detect 80% of all recorded transitions within a 24-hour period of time and a false positive rate of about 19%.

2.1. Data Collection

Five days (Mon-Fri) of ground truth occupancy data were gathered for ten areas using seven cameras covering 18 transition boundaries. Figure 1 shows the building areas used for data collection. The cameras recorded images at 1.5 fps. We explored many different methods to help assist the gathering of ground truth data. Initially, we used a state-of-the-art technique that can count torsos and legs to identify images that contain people [OpenCV; Viola and Jones 2001; Lienhart and Maydt 2002] but found this was only able to identify 80% of the images with people. Even the most advanced techniques are not 100% accurate and need manual correction. In the end, we used a simple background subtraction technique to identify images that contain moving objects and then processed by hand to verify direction and the boundaries being crossed.

The hall occupancies are generally 0. When the hallway is occupied, the occupancy is usually low and the occupants typically exit within 4 to 8 seconds. There are a few instances that the hall occupancy deviate from this typical pattern. When the data was collected, there was construction work being done near Lab 1. On Monday and

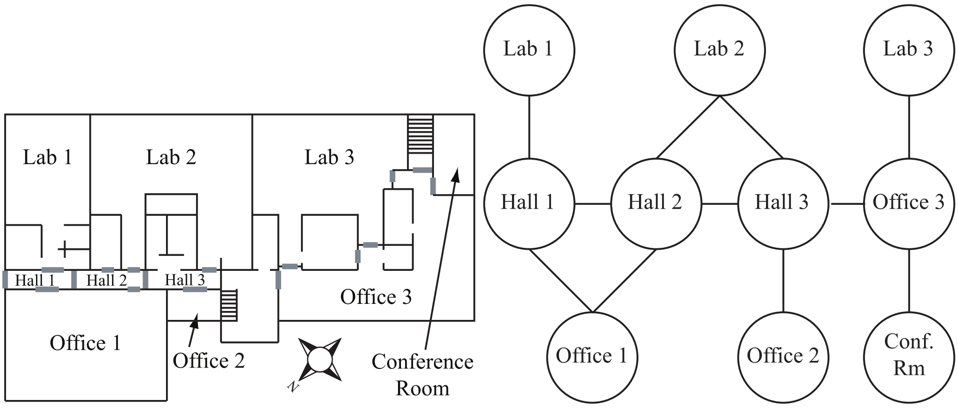


Fig. 1. The left shows the ten areas data was collected for and the 18 boundaries (gray lines) defining the areas. The right shows a graph representation.

Thursday, about seven people conducted building inspections in the hallway. There are also some instances where people held lengthy conversations in the hallway sections.

Figure 2 shows the occupancy data for ten areas. The Office 1, Office 3, Lab 1, and Lab 3 show occupancy pattens more typical of work routines though with some slight variations. Office 1 is used by school administration. Workers come in consistently at 8 am, leave for lunch, come back from lunch, and then leave at 5 pm. Office 3, Lab 1, and Lab 3 are used by professors and graduate students. Though the same general daily pattern can be seen, we can also see some occupants stay later (all night in some cases) and arrive and leave at odd hours. Office 2 and the Conference Room are similar since they both are mainly used for meetings. Office 2 is a small office that is used for student counseling meetings and certain administrative work. Lab 2 is currently not being used by any department and serves as storage space.

3. PREVIOUS MODELS

This article is an extension of Erickson et al. [2011] that also includes the moving window Markov Chain model described in Erickson and Cerpa [2010]. In particular, we include a comparison among the blended Markov Chain, closest distance Markov Chain, and the moving window Markov Chain with respect to modeling occupancy. In our previous work [Erickson et al. 2009], we developed a multivariate Gaussian fit of room occupancies. We start first by defining the occupancy state for a building. The occupancy state x is a vector of m dimensions, where x_i is a nonnegative integer corresponding to the occupancy (number of people) in room i. We define a probability distribution $p(x)$ on such state vectors. Over a specified period of time, we model the room occupancies with a multivariate Gaussian model. Although the vectors x are discrete, the Gaussian model provides a reasonable approximation, and is simple to estimate and use for inferences. For our model, we define hourly pdfs that allow the calculation of the probability for a particular occupancy state occurring within a given hour.

Let $O = (r_1 \cdots r_m)$ denote all occupancies that occur per second during a period of time and r_i is a vector of occupancies for room i where $i = 1 \cdots m$. Let μ_i denote the average occupancy for room r_i. We calculate a vector of means $\mu = (\mu_1, \ldots, \mu_m)$ and covariance matrix Σ from O. Using μ and Σ, we define a probability density function (PDF) f:

$$f(O; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(O - \mu)^T \Sigma^{-1} (O - \mu) \right\}.$$

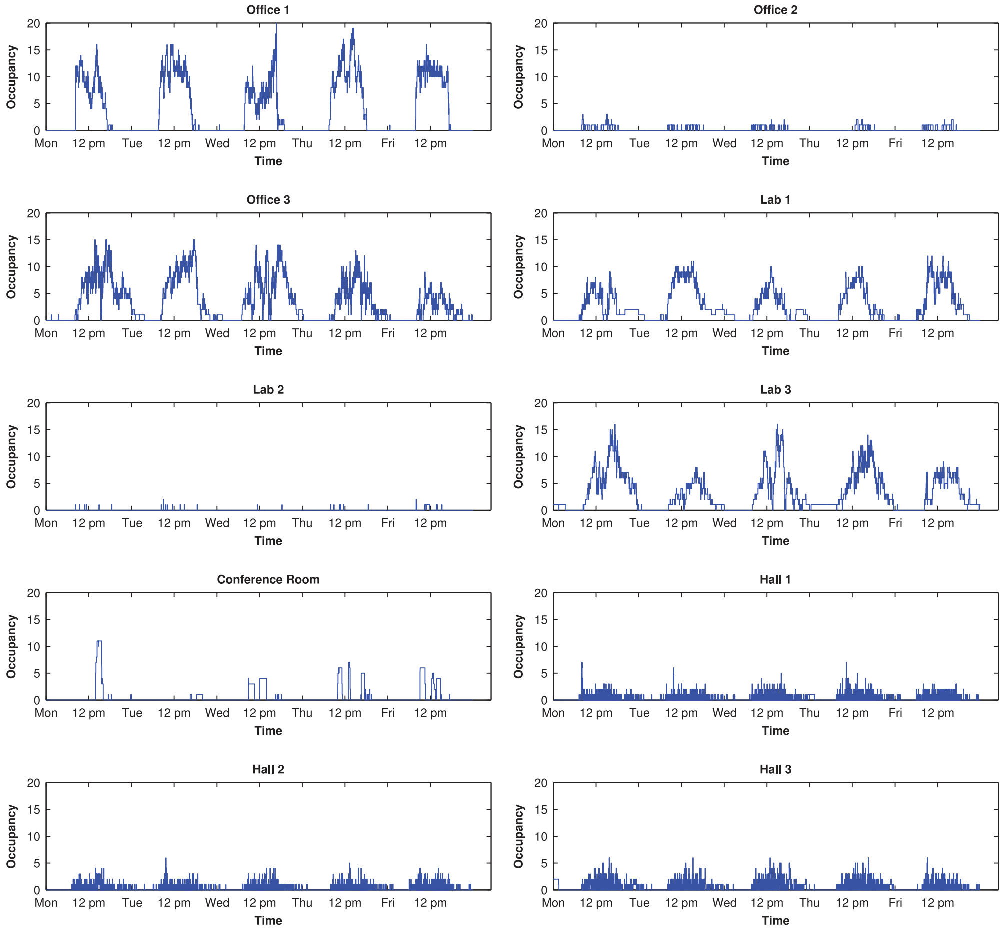


Fig. 2. Five days of ground truth occupancy data for eight areas.

We define hourly Gaussian models with mean μ_h and covariance matrix Σ_h where f can give a probability of an occupancy occurring for a specific dataset O_h for hour h. Using this function, we can randomly draw occupancy vectors from the distribution. Given a starting occupancy, all the possible occupancies that can occur in the next timestep are examined. For each possible occupancy the probability of the occupancy occurring using the current PDF is calculated. Using these probabilities, the occupancy for the next timestep is chosen. The drawback of this method is that it causes a great deal of pacing behavior. For example, if a person leaves the office to enter a hallway at one timestep, there is a high probability that the person will re-enter the office in the next timestep. This occurs since the distribution does not take into account the behavior observed in the previous timestep. This can cause predictions to favor parts of the distribution that have high probabilities. This is particularly pronounced for rooms rarely occupied. In these cases, the model rarely allows room entry and vacates the room too quickly.

In order to create a prediction model suitable for HVAC control, we need a model that captures the temporal nature of the occupancy changes along with the inter-room correlations and occupant usage of the areas. The ABM does not take into account

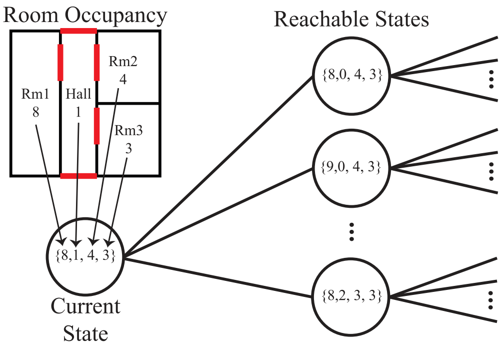


Fig. 3. The occupancy state representation.

inter-room correlations and would require an additional modeling framework to be used for predictions based on time. The Gaussian model considers both the temporal and inter-room correlations. However, the distributions do not take into account previous behavior and do not accurately capture the usage of seldom used rooms.

4. MARKOV CHAIN MODEL

This section discusses the basic structure and concepts of a Markov Chain based occupancy model. Using this structure, we develop three different occupancy models; Closest Distance Markov Chain, Moving Window Markov Chain, and Blended Markov Chain. The best performing and most elegant model is the Blended Markov Chain (Section 4.3), which is used for the simulations and energy analysis sections of the paper (Sections 6–8).

We model the temporal dynamics of the occupancy in a building with a Markov Chain (MC). The state of the chain consists, as with the Gaussian model, of a vector x containing the occupancy at each room, and transitions to a new state occurs with a probability that depends only on the current state and the time. The are several advantages to this approach. The first is that it enables us to predict the occupancy distribution at $t + \Delta t$ given the occupancy distribution at time t by multiplying the Δt with the transition matrix. This allows us to predict when rooms will be likely occupied and begin conditioning beforehand. The other advantage to this approach is that this prediction takes into account the inter-room correlations. For instance, a waiting room could have a high positive corralation with the adjactent offices. A Markov Chain in a state with the occupancy 0 for the waiting room would then indicate with high probability that adjacent offices are also empty. Thus, the model takes advantage of the inter-room corralations in order to make improved predictions.

Note that state transitions must respect the room graph of the building as shown in Figure 1. Internal movement of occupants among adjacent rooms maintains the sum of the components of the occupancy vector. External movement of occupants to/from the outside of the building from rooms connected to the outside will change the said sum. This limits the states to which we can transition to from a given state (we call these *reachable* states). Figure 3 illustrates this.

The MC state at each time is represented by a vector where each component represents occupancy in a specific room (Figure 3). Let R represent the set of n rooms to be modeled. For each room in R, there is a maximum occupancy. We define $X = \{x_0, \ldots, x_m\}$ to be the set of all room occupancy combinations that are possible given the maximum room occupancies of R where m is the total number of states. Thus, X represents all observable occupancy states that can be represented by a given set of rooms R.

One issue that needs to be addressed is the potentially large state space. Assume we have $n = 4$ rooms, each with a maximum occupancy of 20 people. In this case, the MC will contain $m = 20^4 = 160{,}000$ states. As the number of states increases, more data is required to calculate the probabilities of the observable states. The transition matrix is sparse because many transitions are impossible due to physical constraints of the building. However, the fact still remains that as more rooms are added, the number of states to manage increases exponentially. To reduce the component cost, we define the MC only on the states that were actually observed during training. Although we will not be able to generate transitions to all observable states, practically our model will be realistic if using a large enough training set. As the WSN gathers data over time, more of the state space will be represented. Additional strategies of reducing the state space are addressed in Section 4.4.

With the observable states of the MC defined, we next define the transition probability matrix. Let p_{ij} represent the probability of moving from state j to i where x_t represents an occupancy state at time t. For each state in X, we calculate

$$p_{ij} = P(x_{t+1} = i | x_t = j).$$

Since the training data collected is at the resolution of seconds, each time step of the MC represents 1 second. The transition probabilities are estimated from the data normalized counts:

$$p_{ij} = n_{ij} / \sum_{k=1}^{m} n_{ik},$$

where n_{ij} is the number of times a transition from state i to state j in the set, and m is the total number of states.

Since certain occupancy changes occur with greater probability depending on the time of day, the time of day must be incorporated into the model. Consider a person standing in a hallway at 8:00 am. Since it is early in the day, it is likely that the person has arrived for work and will move into either a lab or office. However, if we consider the same scenario at 8:00 pm, it is more likely that the person will exit the hallway in order to leave the building. To incorporate time into the model, we define multiple transition matrices that govern the state changes within different slots of time, thus defining an inhomogeneous MC. That is, we consider separate models for each hour that will be trained using the data for that hour, just as we did in Section 3 with the Gaussian model. It should be noted that this structure allows for simple on-line training; as states are observed, they can immediately be incorporated into the transition tables.

While considering only observed (as opposed to observable) states and multiple transition matrices does allow us to model occupancy dynamics efficiently, a problem arises. Since we only consider states observed in the training data, partitioning the data in temporal sets will create discontinuities at the slot boundaries. Suppose we partition the data to create hourly transition matrices and are predicting occupancy for hours h and $h + 1$. After 3600 steps (one hour), we are in some state x, the hour changes from h to $h + 1$, and the model switches to the hourly transition matrix for $h + 1$. It is possible the transition matrix for hour $h + 1$ has no probability for occupancy state x. This occurs if state x never occurs in the training data for hour $h + 1$. Even though, in reality, the state x can occur in hour $h + 1$, if x does not occur in hour $h + 1$ of the training data, then we cannot calculate the transition probabilities for x. This cannot be solved by introducing a small epsilon probability value of transitioning into another state because the next state chosen may also not be represented in the transition matrix. The MC becomes a random walk until it enters a state that was captured by the training data.

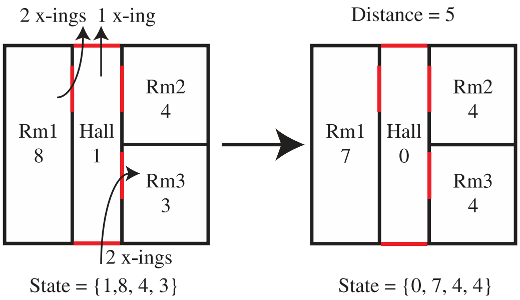


Fig. 4. Example of the closest distance metric with state (*Hall*, *Rm*1, *Rm*2, *Rm*3) and distance 5.

Similarly sink states can also occur if an occupancy state only occurs once in the training set. Suppose we have a set of occupancy states O that we use to train a transition matrix. Let x be the last occupancy state of the training set of O. If x is a unique state in O, then x never transitions to any other state. If the occupancy state transitions to x, it will remain in that state until the model changes to a transition matrix that does contain a transition probability. We propose two different methods of solving the boundary discontinuities and avoiding sink states.

4.1. Closest Distance Markov Chain

The closest distance Markov Chain (CDMC) attempts to solve the discontinuities the time slot boundaries. Suppose we finish simulating hour h and are currently in state x. Before switching to the transition matrix for $h+1$, we first check to see if it contains a probability for x. If the probability does not exist, then we examine all the states of transition matrix $h+1$ that does have a probability and choose the state closest to x.

We define the distance between two states to be the number of transition to or from the outside world in order to account for the difference between the states. Let M_i represent the minimum number of area transitions that is needed for a single person to enter or exit out of a building from room i. Let $A = (a_0, \ldots, a_n)$ and $B = (b_0, \ldots, b_n)$ represent two occupancy states where a_i and b_i are the occupancies of room i. We define distance $d_{A,B}$ between A and B to be $d_{A,B} = \sum_{i=0}^{n} M_i |a_i - b_i|$. Figure 4 shows an example of this distance metric. For this four-room example, the distance between the states is 5. One transition boundary is crossed to leave Hall. Two transitions crossings are needed for the person to leave Rm1. Two transition crossings are needed for the person to enter Rm3. Rm1 and Rm2 require two transitions each since a person must first enter the hallway and then from the hallway exit to the outside world.

Although this method eliminates sink states at the hourly breaks, it does not guarantee that the sequence of states is valid. It is possible for people to “teleport” in or out of areas. Consider the floor plan in Figure 4. Suppose we have the state (0, 0, 0, 0) (all the rooms are empty) and the closest state is (0, 0, 2, 0) (two people are in the office). The transition from (0, 0, 0, 0) to (0, 0, 2, 0) is impossible since the two people must first pass through the hallway to enter the office. Also, it is still possible to enter sink state in the middle of a prediction. Applying the closest distance metric again does little to help since it will likely choose the previous state and then immediately re-enter the sink state.

4.2. Moving Window Markov Chain

Sink states only occur when the next transition matrix does not contain a probability for the current state. To avoid sink states, we wish to ensure that the current state exists in the next transition matrix. One way to avoid sink states is to use a transition matrix that is trained over a moving window of data. As state changes progress, the window of data that the transition matrix is trained changes.

Thus far, we have defined the MC model to have hourly transition matrices that capture the occupancy behavior for a particular hour. However, the choice to train over a specific hour is somewhat arbitrary. We can create a transition matrix for any specified window of time. Rather than define a transition matrix every hour, we instead define a transition matrix every n minutes with each transition matrix still trained with an hour of training data. Thus, each transition matrix overlaps its neighboring transition matrices by $60 - n$ minutes. By having this overlap, we can reduce the likelihood of entering a sink state. Let D represent the training data. Let the function $g(D, t_0, t_1)$ return the transition matrix trained with D defined for the window of time t_0 to t_1. For example, $g(D$, 8:00 am, 9:00 am) represents a transition matrix trained with D for hour 8:00 am to 9:00 am. If $n = 10$ minutes, then we get the following 140 overlapping transition matrices,

$$\begin{aligned} T_1 &= g(D,\ 00{:}00,\ 1{:}00) \\ T_2 &= g(D,\ 00{:}10,\ 1{:}10) \\ &\vdots \\ T_{140} &= g(D,\ 23{:}00,\ 00{:}00) \end{aligned}$$

Once the transition matrices are defined, we perform predictions with Algorithm 1.

ALGORITHM 1: MWMC

```
$CurrState \leftarrow$Starting state
$TransMat \leftarrow T_i$ Starting transition matrix
$Sim \leftarrow \{\}$
$w \leftarrow 0$ Window move counter
for $t = 1$ to Simulation Duration do
  if $w > n * 60$ seconds then
    $NextMat \leftarrow T_{i+1}$ Next transition matrix
    if $CurState \in NextMat$ then
      $TransMat \leftarrow NextMat$
      $w \leftarrow 0$ Reset window counter
      $i \leftarrow i + 1$
    end if
  end if

  if $w < 2 * n * 60$ seconds then
    $CurrState \leftarrow RandomState(TransMat, CurrState)$
  else
    $CurrState \leftarrow$Closest State in $T_{i+2}$
  end if
  $Sim.add(CurrState)$
  $w \leftarrow w + 1$
end for
```

Once entering a transition matrix, after n*60 steps (n minutes) have passed, the algorithm attempts to move to the next overlapping transition matrix. However, it only changes transition matrices if the currently defined state exists in the next transition matrix. Otherwise, it remains in the current transition matrix. Though this lessens the probability of entering a sink state it is still possible to encounter. If this occurs, we use closest state of the next transition matrix.

After exploring the parameters by trial and error, we choose $n = 30$ in order to reduce the number of overlapping matrices while still reducing the possibility of entering a sink state.

4.3. Blended Markov Chain

The blended Markov chain approach solves the problem of encountering a sink state without using the closest distance method. In this method, we blend each matrix into a single state space containing each state of each transition matrix. This is similar to the MWMC, but considers all states observed from the training data.

If the day is partitioned in K parts, then we have K transition matrices $T_1 \dots T_K$ each with m states. We now linearly combine these K transition matrices to obtain K blended transition matrices $\overline{T}_1, \dots, \overline{T}_K$, as follows. The blended transition matrix for slot t is

$$\overline{T}_t = \sum_{s=1}^{K} \beta_{ts} T_s, \tag{1}$$

where the coefficients $\beta_{t1}, \dots, \beta_{tK}$ are positive and sum to one (so that $\overline{T}_t$ is a valid transition matrix). We want these coefficients to be approximately 1 for close slots and quickly decrease to 0 for farther slots. We can achieve this by defining the coefficients as

$$\beta_{ts} = \frac{\alpha(c_t - c_s)}{\sum_{s'=1}^{K} \alpha(c_t - c_{s'})}, \tag{2}$$

where c_t, c_s are the centers of slots t, s, and with a "slot" function

$$\alpha(x) = \sigma\left(\frac{2a}{d}\left(x + \frac{d}{2}\right)\right) - \sigma\left(\frac{2a}{d}\left(x - \frac{d}{2}\right)\right) \qquad x \in \mathbb{R}, \tag{3}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function, $a > 0$ is the slope at the slot boundaries and $d > 0$ is the slot width. Since the sigmoid is monotonically increasing and satisfies $\sigma(-x) = 1 - \sigma(x)$, it follows that $\alpha(x)$ is positive and symmetric around its center. Figure 5 shows the slot functions for several slots. This way, each entry in the blended transition matrix $\overline{T}_s$ incorporates information from all slots, but heavily weighting slot s.

For our model, we choose $K = 48$, partitioning the data into half hour transition matrices with slot widths of $d = 3$. We set a slope of $a = 10$ at the boundaries. We choose the c_k to be center of the current hour. Figure 5 shows the blending coefficient for these particular parameters. By defining the parameters in such a manner, we create overlapping slot boundaries. This increases the number of preferred states available to transition into, and decreases the chance of choosing states completely outside the slot boundaries. The coefficient heavily favors transitions to states within a given hour time slot and somewhat considers states in the adjacent half-hour slots. States outside this time frame are still considered, but with greatly reduced probability. These parameters were chosen through trial and error. The criteria of the parameter selection was to maximize slot size while minimizing the transitions into states completely outside the preferred slot boundaries. While larger values of K could be used, smaller values of K are preferred since a large K reduces the slot size and the number of observed states for each transition matrix. We also prefer to only draw from states that are close with respect to time.

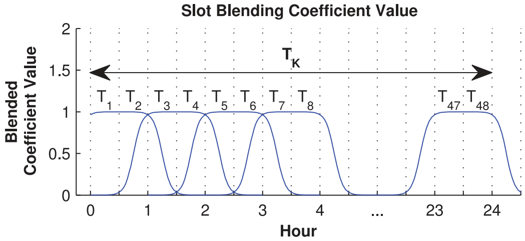


Fig. 5. Relationship of blending coefficient, hourly time slot, and transition matrices.

4.4. Scalability to Large Buildings

The number of observable states increases exponentially with the number of rooms, but as described earlier we limit the complexity of our model by using only states observed in the data sample. If this has length s over a given time period, then the number of observed states N satisfies $N \leq s$, and practically $N \ll s$ because of repeated states. Besides, the $N \times N$ transition matrix T is typically sparse, because physical constrains make many transitions impossible and because not every possible transition is observed in the sample anyway. Indeed, the number of nonzeros in T must be smaller than s, and again is practically much smaller than s because of repeated transitions. For our 5-day training set, we observed $s = 432$K samples resulting in $N = 3809$ and 19578 nonzero transitions ($= 0.14\%$ of all N^2 transitions).

It is conceivable that, with a very large number of rooms and a very long sample, N or the number of nonzeros of T are prohibitively large. Several simple strategies can be used to achieve a manageable model. First, s should not be larger than necessary to achieve a sufficiently accurate model. Another strategy for managing the number of states is to utilize multiple MC models. Suppose we have a building with 40 rooms that is divided into two wings each containing 20 rooms. Rather than have a state representation that includes all 40 rooms, we can train one MC per wing. The tradeoff is that more MC models would be required to describe all possible relationships of all the rooms. Lastly, it is often possible to aggregate multiple rooms together thus reducing the number of states. This is because HVAC systems typically condition groups of rooms called zones. We can aggregate room occupancies within a zone, and then create an MC model using zones rather than individual rooms.

5. MARKOV CHAIN PERFORMANCE

In this section, we will examine how well BMC, CDMC, and MWMC capture room occupancy. For each model, 100 simulations of 24 hours were created. Five days (Mon-Fri) of ground truth occupancy data was used to train the models.

While we use hand-corrected ground truth for training our models, it should be noted that it is assumed that a camera system similar to SCOPES would be used to collect data for training in an actual HVAC system. We choose to use ground truth data in this case for model development purposes.

An additional three days (Mon, Wed, Fri) of ground truth data is used for testing. Figure 6 shows two days of training data along with model simulated room occupancy schedules. Even without formal tests, the models seem be capturing the dynamics of the ground truth room occupancies. As expected, the CDMC enters sink states and remains in the state until the end of the hour. This can be seen as multiple flat "plataus" in occupancy, which can be clearly observed in Lab3 at 8 am and in evening at 10 pm. Qualitatively, BMC and MWMC seem to model occupancy variability better than CDMC.

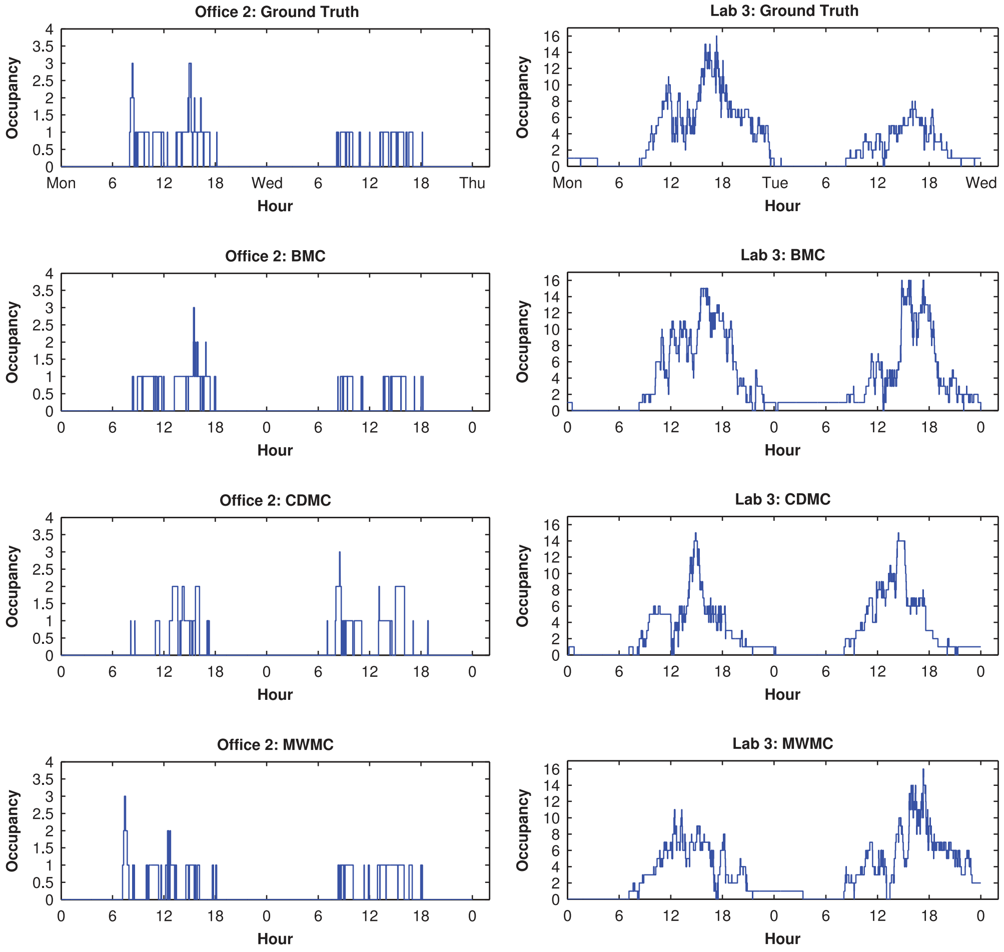


Fig. 6. Comparison of training data with the models for several areas.

5.1. Comparison Metrics

There are several aspects we wish to examine when determining the quality of performance of the models. Given that there is a large amount of variation is expected from day to day, we wish to examine how well the model performs in terms of actual occupancy and how the dynamics of how the occupancy change over time.

We use three metrics to evaluate the quantitative performance of the models. We first examine how long room occupancies remain static to measure occupancy variability. This metric shows how well the models capture certain temporal aspects of the data. In particular, it shows the duration between occupancy changes for a given the current occupancy level. We expect our occupancy models to remain at the different levels of occupancy for similar durations as compared with the ground truth data. The next metric utilizes Jensen-Shannon divergence (JSD). This is a method that applies Kullback-Leibler divergence (KLD) to compare the similarity of two distributions. The advantage of JSD over KLD is that JSD will always return a finite value and is symmetrical. Let $KL(P\|Q)$ represent Kullback-Leibler divergence of distributions P and

Q. JSD between P and Q is defined as

$$JSD(P\|Q) = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M).$$

where

$$M = \frac{1}{2}(P+Q), \quad KL(P\|Q) = \sum_{i} \ln\left(\frac{P(i)}{Q(i)}\right) P(i).$$

We compare the room occupancy distributions of the models and testing data for different windows of time during the day. The last metric considered is the rate that people enter and exit a room. By examining the durations between entrances and exits of a room, we can measure the flow of occupants in and out of a room. Specifically, for a window of time w, we calculate $\lambda_{in,w}$ and $\lambda_{out,w}$ for each room where the variables represent the rate of flow in and the rate of flow out of a room, respectively. Time is taken into account since the rate of flow changes depends on the time of day. For example, we can expect the flow of occupants into Office 1 to be greater in the morning when people are arriving than in the evening when people are mostly leaving.

5.2. Evaluation

Figure 7 compares the average static durations of the various occupancy levels with those seen in the testing set. We use normalized root mean square error (NRMSE) for comparison. For high traffic areas and rooms with low occupancy, BMC and CDMC show similar differences. CDMC typically shows larger differences of durations since the CDMC still tends to get stuck in states near the ends of the hourly partitions. This causes the occupancy level to remain static for periods of time longer than normal. Areas with larger occupancies are prone to containing sink states as more data is required to cover the possible state space. This can be seen in Figure 7 for Office 1 where CDMC remains at several occupancy levels for long durations. Similar results were found for the other areas with higher levels of occupancy such as Office 1, Office 3, Lab 1, and Lab 3. BMC and MWMC show very similar differences. This is expected as both of these methods have a similar method of favoring states within a frame of time. However, as mentioned previously, the MWMC still has the possibility of entering a sink state causing static occupancy levels. However, this occurs much less frequently when compared to the CDMC.

We next examine the JSD for the different models. We partition the day into 3-hour slots. For each of these slots, we generate a distribution of the occupancies and compare this distribution to the ground truth distribution for the corresponding window of time using JSD. Examining the ground truth data in Figure 2, we can expect a fair amount of variation among different days. To establish a baseline of how much divergence is typical from day to day, we compare the occupancy distributions of each testing set day to the remaining testing set days for each window of time and establish the maximum divergence for each time slot. Rooms that are mostly empty, such as Lab2 and the conference room, have JSD of nearly 0 for each window of time. Figure 8 shows the JSD for the rooms that are consistently occupied. When we compare the testing set for each model, we see that the JSD for each window is consistent. We find that all three models have JSDs that are below the maximum observed JSD. We also see that MWMC and BMC have very similar results, which is expected as both strategies have a similar mechanism for choosing the next state.

Lastly, we examine the flow of occupants for each rooms. We consider the hours 7 am-10 am to be morning, 11 pm–2 pm to be afternoon, and 3 pm–6 pm to be evening. Based on these windows of time, we calculate the flow into and out of rooms. Figure 9 shows the flows into Hall 1 and Office 1 for the different simulations. As with JSD, we will use

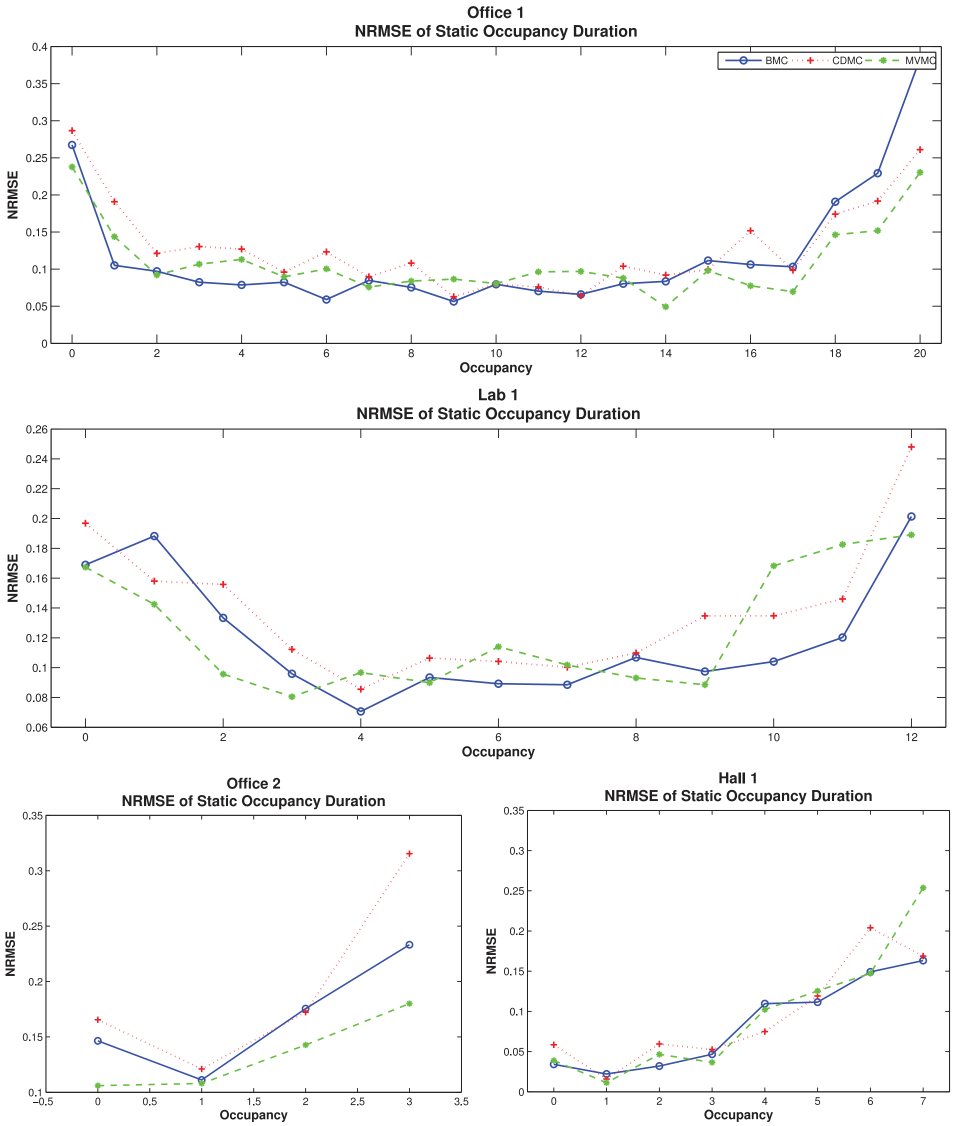


Fig. 7. The NRMSE comparing the average static occupancy duration from 7 am–10 pm.

the ground truth test data to establish upper and lower limits for the λ values. Again, all three methods fall mostly within the limits. However, CDMC and MWMC tends to have more outliers (±2.7σ from the median), which again is caused by sink states. However, since sink states occur with less frequency with the MWMC, the MVMC has fewer outliers than CDMC.

Overall, all three models perform well, but in general, BMC performs better since it does not encounter discontinuities at the boundaries, which can occur for both CDMC and MWMC. For this particular training set, the MWMC did not exhibit discontinuities at the slot boundaries; however, it is possible that with other datasets this could occur. The sequence predicted by BMC is also always valid, which may not be the case for the CDMC and MWMC, where the closest distance is used to escape sink states.

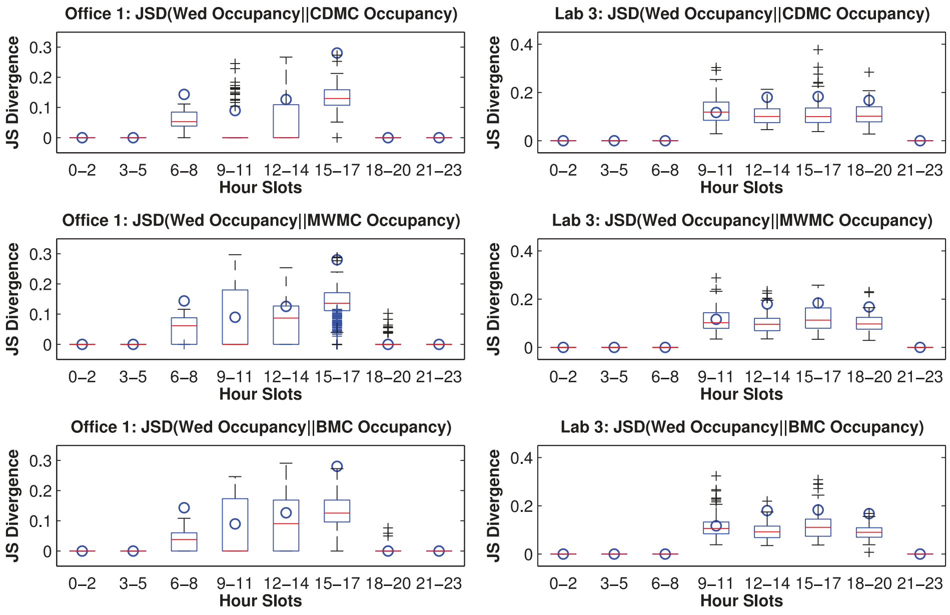


Fig. 8. Each boxplot is for the JSDs observed within the time slot where the + is an outlier. An outlier is defined to be ±2.7σ from the median.

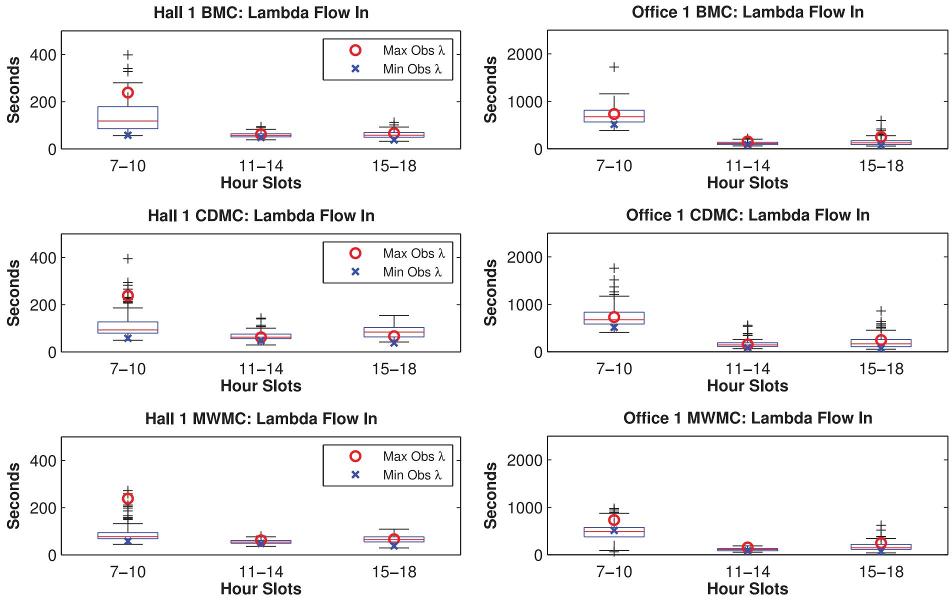


Fig. 9. Flows for Hall 1 and Office 1, and the min and max flows observed in the testing set.

6. HVAC CONDITIONING STRATEGIES

Using the BMC, we define a predictive control algorithm for temperature. We start by defining the thermal and ventilation criteria that our HVAC strategies must meet to have an ASHRAE-compliant building. We then define the occupancy-based temperature control strategy OBSERVE.

Table I. PMV Comfort Classes

Hot		PMV	> 2.5
Warm	$1.5 \leq$	PMV	< 2.5
Slightly Warm	$0.5 \leq$	PMV	< 1.5
Neutral	$-0.5 \leq$	PMV	< 0.5
Slightly Cool	$-1.5 \leq$	PMV	< -0.5
Cool	$-2.5 \leq$	PMV	< -1.5
Cold		PMV	< -2.5

6.1. Conditioning Criteria

6.1.1. Temperature. Thermal comfort is a complex measurement that depends on many aspects such as temperature, humidity, air velocity, occupants' clothing and activity [Ploennigs et al. 2010; Pfafferott et al. 2007]. The most common comfort measurement is Fanger's Predicted Mean Vote PMV as standardized in ISO 7730 [Olesen and Parsons 2002]. Fanger's model is not an undisputed thermal comfort measurement [Olesen and Parsons 2002; Pfafferott et al. 2007; Schumann et al. 2010], but provides good generalized results in many cases. Fanger computes the thermal comfort PMV value between −3.5 and 3.5. Rounding the PMV results in the comfort classes (Table I). Fanger's PMV depends on air temperature [Olesen and Parsons 2002], radiant temperature, humidity, air velocity, occupants clothing and activity. The model is nonlinear and based on large-scale studies in climate chambers.

Measuring the PMV based on this model is very complex due to the many influences. The activity and clothing level of occupants is simplified by assuming defaults such as office work and clothing level is correlated to the outside temperature. However, this still requires an air-temperature, radiant-temperature, humidity, and air-velocity sensor in each room. Air-temperature and simple humidity sensors could be added to our wireless sensor node, but radiant-temperature and air-velocity sensors are complex and expensive, such that large-scale installations are not affordable. Given these complexities, our work focuses on optimally controlling temperature rather than attempting to control PMV. From a control perspective, our goal is to meet a target temperature defined by a comfort metric, which may not necessarily be Fanger PMV. Different comfort metrics will establish different temperature set-points. It is thus more important to meet specific target temperatures than to meet a specific comfort metric. As metrics are developed and better environment sensing is available, we want our system to meet the temperature goals dictated by the new comfort criteria.

6.1.2. Ventilation. ASHRAE Standard 62.1 [ASHRAE 2007b] uses the following to calculate outdoor air ventilation rates:

$$V_{bz} = R_p P_z + R_a A_z, \tag{4}$$

where z denotes the zone, V_{bz} is the ventilation rate, R_p is the minimum CFM/person, P_z is the number of people, R_a is the minimum CFM/ft^2, and A_z is the floor area. R_a and R_p change depending on the target use of the room as specified by the ASHRAE 62.1 standard. This standard is for areas without a demand ventilation system. For our rooms, the required CFM per person is 5.

6.2. OBSERVE Temperature Control Strategy

This section defines the Occupancy-Based System for Efficient Reduction of hVac Energy (OBSERVE) predictive control algorithm, which uses a BMC to predictively condition room temperature. The function $BMC(OccState, predLen)$ return, a predicted schedule of occupancy $occPred$, where $occPred_t$ is a probability a space being occupied at time $t = 1 \cdots predLen$ and $OccState$ is the occupancy state at time $t = 0$. This

ALGORITHM 2: OBSERVE temperature control algorithm.

```
$CondTemp_{i,j} \leftarrow$ Condition temperature from time $i$ to $j$
$CurrHour \leftarrow$ Current hour
$T_{TG} \leftarrow$ Temperature such that $PMV = 0$
$T_{ASH} \leftarrow$ Temperature such that $-0.5 < PMV < 0.5$
$pThresh \leftarrow$ Probability threshold of occupancy

for Every $n$ minutes do
  $CurrOcc \leftarrow$ Current occupancy state
  $occPred \leftarrow BMC(CurrOcc, predLen)$

  for Each room $r$ and point of time $t$ in $occPred$ do

    $occupied \leftarrow$ All periods $occPred_{t \to t+60} > pThresh$

    if $occupied > 5$ minutes of next 15 minutes then
      $CondTemp_{t,t+15} = T_{TG}$
    else if $occupied > 20$ minutes of next 60 minutes then
      $CondTemp_{t,t+60} = T_{TG}$
    else if $5 \leq CurrHour \leq 24$ then
      $CondTemp_{i,i} = T_{ASH}$
    end if

  end for
end for
```

conditioning strategy is also applicable for binary occupancy data. The BMC can be easily adapted to binary data by representing the MC state at each time as a vector where each component is a binary indication of occupancy (instead of actual occupancy). Algorithm 2 defines the OBSERVE temperature control strategy.

It should be noted that T_{ASH} is assumed to be a fixed temperature calculated by assuming fixed values for PMV parameters such as metabolism and clothing. Every n minutes, we examine the current occupancy state and then make a predicted schedule *occPred* for occupancy *predLen* into the future from the current time. We then check for two different types of occupancy. It first checks short windows of time (5 minutes of the next 15 minutes) for occupancy to ensure that rooms, such as copy rooms, that are not frequented often but have occupied durations of several minutes are conditioned. The second longer window (20 minutes of the next 60 minutes) ensures that rooms that are frequented constantly but may have short stay durations are still conditioned. Hallways are an example of such an area. Since no standards currently define what constitutes a significant duration of occupancy, we chose values that seem reasonable to us.

7. EVALUATED STRATEGIES

Four different strategies are considered, which are designed to compare PIR occupancy systems and camera occupancy systems with and without predictive conditioning against a standard baseline approach. The first strategy, ASHRAE Baseline, is a baseline that employs a typical HVAC control strategy assuming maximum occupancy for ventilation and conditions all rooms from 7:00–22:00. The second strategy, Reactive Binary, is a binary-based reactive control of temperature and ventilation. This strategy is geared toward a PIR-based system that does not condition predictively. Rooms are conditioned to a target temperature when occupied and are not predictively preconditioned. Rooms that are unoccupied are still conditioned to the max/min temperatures allowed by ASHRAE. Since precise room occupancies are unknown, rooms are ventilated according to the estimated maximum occupancy when occupied. The maximum occupancy is determined using ASHRAE Standard 62.1 based on area and room purpose. The third strategy, OBSERVE Occupancy, uses the OBSERVE algorithm described in Section 6.2. This strategy predictively conditions rooms and adjusts ventilation based on current actual occupancy, which is determined using Eq. (4) during warm months (Apr - Oct). During the colder months, ventilation rates are increased. This increased ventilation is an optimization for terminal reheat HVAC system and is discussed in Section 8.2.1. The last strategy, OBSERVE Binary, uses the OBSERVE algorithm, but uses binary data to train the BMC model. This is to examine a system that utilizes data from a PIR system with predictive conditioning. This strategy makes binary predictions of occupancy and ventilates with the same maximum occupancy assumption of the reactive control strategy.

8. PERFORMANCE RESULTS

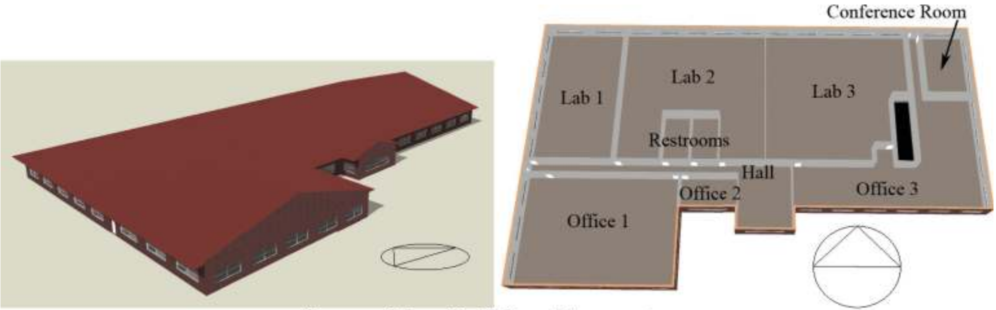
8.1. Building Energy Simulator

EnergyPlus is one of the premiere tools for modeling the energy of buildings. It takes into account factors such as weather, HVAC design, and construction materials. In this section we define the main parameters used for an EnergyPlus model to test our HVAC strategies.

8.1.1. Building Parameters. EnergyPlus simulations are run for three different locations; Fresno CA, Miami FL, and Chicago IL. The EnergyPlus model replicates the geometry shown in Figure 1 and is constructed to ASHRAE standards. Figure 10 summarizes the main model parameters. The HVAC is a terminal reheat system that uses a single air handler unit (AHU) with variable air volume vents (VAV). The sizing of the AHU is done according to ASHRAE Standard 62.1 [ASHRAE 2007b]. This type of HVAC system cools air at the AHU level and heats primarily at the VAV level. This is common for many office buildings [Burke and Keeler 2009]. It is often used since it is able to heat and cool areas while still sharing the same AHU. We save the evaluation of other types of HVAC systems for future work. The set-points correspond the temperatures that predict $-0.5 \leq \text{PMV} \leq 0.5$ assuming 40% humidity, 1 m/s airflow, and clothing coefficients of 1.0 and 0.5 for winter and summer, respectively (see Figure 10). Fresno, Miami, and Chicago have 40%, 52%, and 32% average humidities, respectively. The HVAC system controls humidity to be between 35% and 45% using a cool-reheat heating coil.

8.1.2. Simulated Occupancy Schedules. EnergyPlus models typically use static occupancy schedules specified in sources such as ASHRAE 90.1 [ASHRAE 2007c] and DOE-2 [DOE 2012] and are often based on survey data. For our model, however, we will use the BMC to generate occupancy schedules. A BMC trained with 5 days of ground truth data is used to generate 23 days of simulated "ground truth" occupancy data for the building model. These simulations will serve as our occupancy schedules for the weekdays of the month. The building is assumed to be empty on weekends. Static occupancy schedules are used for the restrooms since we have no data for privacy reasons.

Our control strategy assumes a system similar to SCOPES for occupancy monitoring. For 20% of the time, the system inverts the direction or detects a false positive or negative transition. We simulate SCOPES system error by artificially introducing errors into the simulated ground truth data. Because directional errors occur with roughly the same frequency, even with 80% accuracy, the observed system data is close to ground truth. Since errors may produce states that do not exist in the BMC, the closest distance metric defined earlier is used to find the closest state for the OBSERVE



EnergyPlus Building Parameters

HVAC	Single Duct AHU, VAV with terminal reheat Gas heating and cooling 9 zones
Humidity	40%
Airflow	1 m/s
Clothes	0.5–1.0
Temperature Setbacks	Heating: T_{tg} = 75°F, T_{ASH} = 70°F Cooling: T_{tg} = 78°F, T_{ASH} = 82°F
Areas	Total: 30,130 ft^2
Materials	Concrete exterior walls Metal interior walls 30% double glazed windows

Fig. 10. Main building parameters used.

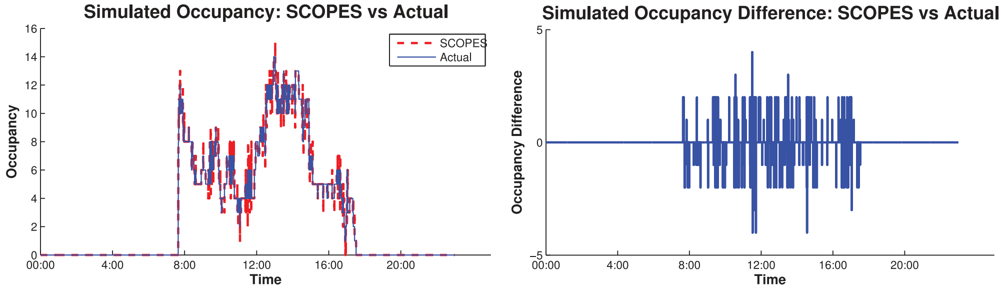


Fig. 11. Simulated SCOPES data.

algorithm. Figure 11 compares the simulated SCOPES with the simulated ground truth.

8.2. Energy Savings

We evaluate the energy savings possible using the defined HVAC strategies for each location. These results take into account the fan, pump, heating (gas), and cooling (gas) energy consumption of the building. Figure 12 shows the monthly breakdown of the energy consumption for each location. Significant energy savings are achievable over a standard commonly used baseline; we see up to 112% improvement (Figure 13) in some cases. We see that heating and cooling account for most of the energy usage and that heating requires more energy than cooling. OBSERVE Occupancy out-performs binary strategies. The strategies have the most substantial savings during the colder months. This is primarily because heating consumption is greater than cooling consumption.

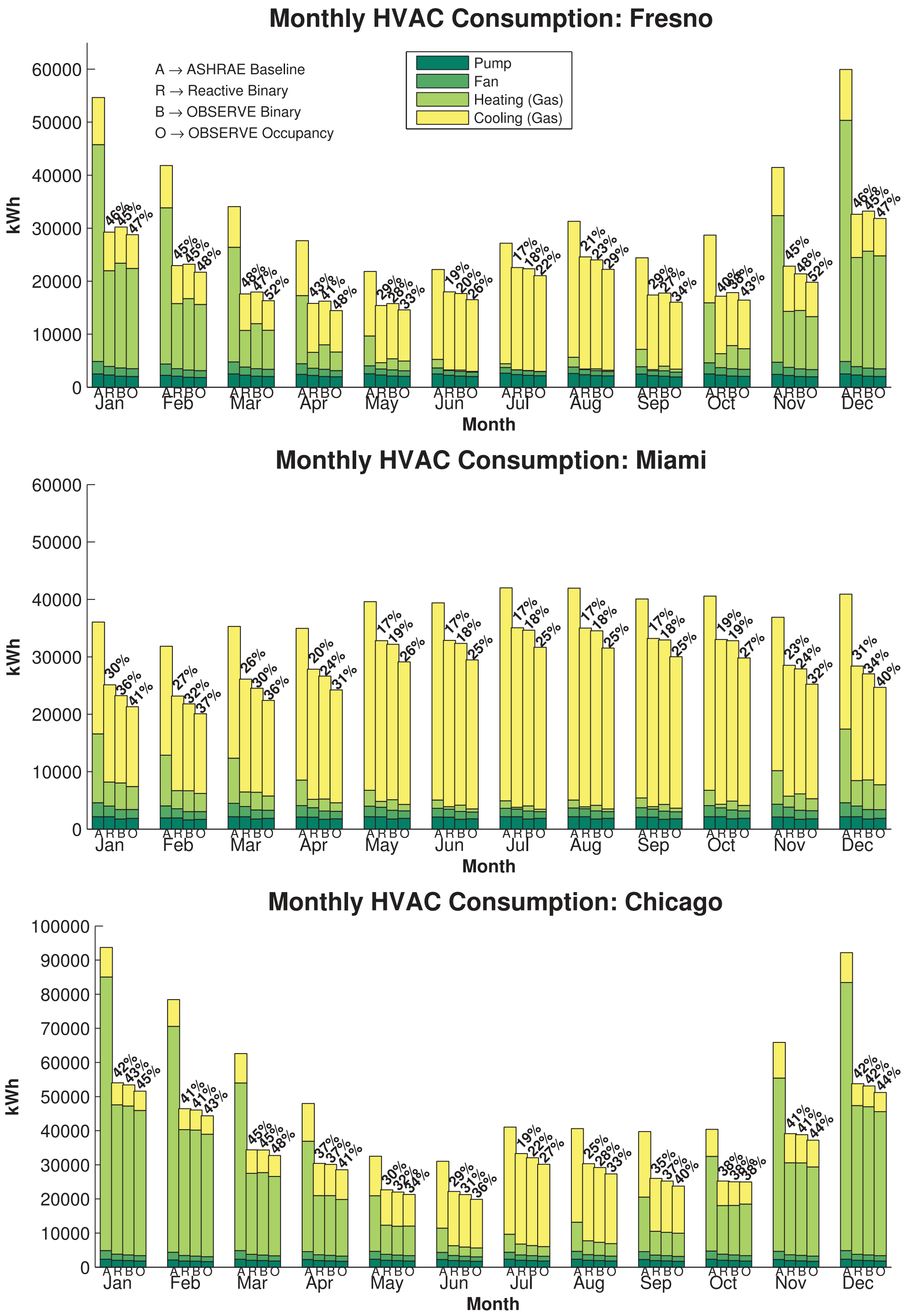
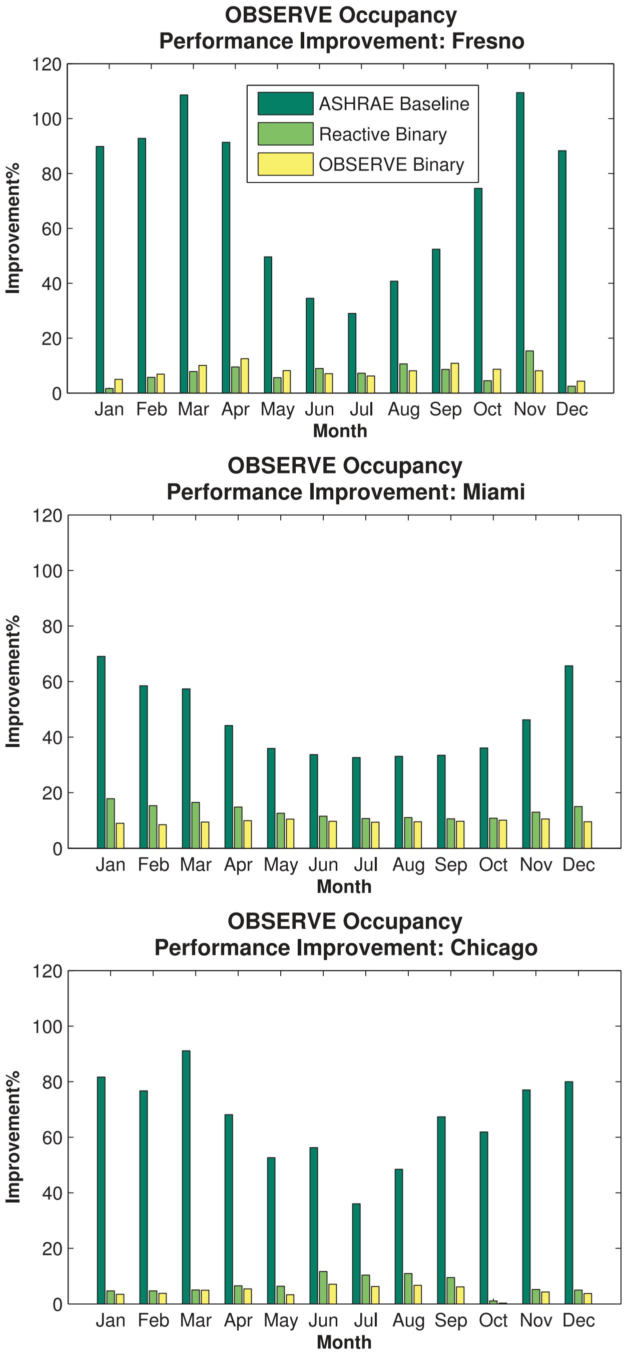


Fig. 12. The breakdown of the energy consumption for each month and strategy for three different locations.



OBSERVE Occupancy
Performance Improvement: Fresno

OBSERVE Occupancy
Performance Improvement: Miami

OBSERVE Occupancy
Performance Improvement: Chicago

Fig. 13. Improvement of energy consumption over baseline.

During the coldest months in Fresno, the strategies are 45%–47% more efficient than baseline; in particular, OBSERVE Occupancy shows 88% to 112% improvement over baseline. For Fresno, OBSERVE Occupancy, OBSERVE Binary, and Reactive Binary show annual savings of 42.3%, 38.3%, and 37.9%, respectively. Similar results are found for the Chicago location. The difference among strategies increases during the warmer summer months. For summer months in Fresno, OBSERVE Occupancy shows 9%–11% improvement over Reactive Binary and 7%–10% improvement over OBSERVE binary. For summer months in Chicago, OBSERVE Occupancy shows 10%–12% improvement over Reactive Binary, and 6%–7% improvement over OBSERVE Binary. This suggests OBSERVE Occupancy is more efficient for warm weather areas. When we examine the results from the warmer Miami location, OBSERVE Occupancy shows 11%–18% improvement over both binary strategies throughout the year. OBSERVE Occupancy again shows significant improvement over baseline (33%–69%). We see that OBSERVE Occupancy is 26%–41% more efficient than baseline whereas Reactive Binary is 17%–30% more efficient. OBSERVE Occupancy, OBSERVE Binary, and Reactive Binary have annual savings of 30.4%, 23.7%, and 21.4%, respectively, for Miami.

8.2.1. Discussion. We see that all three strategies are significantly more efficient than baseline and that there are noticeable differences among the strategies. The main source of these differences is related to ventilation and the terminal reheat system used for the building. OBSERVE Occupancy is able to adjust the ventilation rate according to estimated occupancy whereas OBSERVE Binary and Binary Reactive use the maximum ventilation rate during periods of occupancy. Typically, increasing the ventilation rate introduces additional outside air into the air handler loop and increases the energy consumption since the outside air needs to be conditioned; recirculating air is more efficient since it takes less energy to maintain preconditioned air. However, there are other ventilation efficiency factors that need to be considered with a terminal reheat system. A terminal reheat system partially preheats outside air before entering the AHU then reheats air at the room VAV. Under certain conditions, it is more efficient to increase ventilation. Increasing ventilation means losing some energy since more outside air needs to be conditioned. However, increased ventilation distributes the heating load across all the VAV's and increases the amount of VAV pre-heated air into the loop. Based on EnergyPlus simulations, we found increased ventilation during cold months increases efficiency. The same is not true for cooling. Since cooling is done completely at the AHU level, the VAV's can no longer be recruited to help distribute the load and any additional ventilation decreases efficiency. The significant improvement of OBSERVE Occupancy over the binary strategies in Miami confirms this. Thus, an HVAC system that does all the conditioning in the AHU will benefit the most from optimal venting. This could be achieved in many terminal reheat systems by increasing the AHU preheating.

The optimal ventilation strategies works best in areas with higher cooling loads. In these situations, the binary models do not save as much energy. This is can be seen in the results for Miami. The OBSERVE Occupancy consistently consumes 6%–10% less energy than a binary-based strategy. A large building with an average bill of $54K a month would save an additional $3K to $5.4K a month. This additional savings, entirely attributed to more optimal ventilation, is nontrivial. Over the lifetime of a building, it is cost effective to pursue even small gains in efficiency [Erickson et al. 2013].

8.3. Conditioning Effectiveness

While substantial energy savings are possible, we must verify the building is ASHRAE compliant. For this analysis, we focus on the Fresno EnergyPlus simulations.

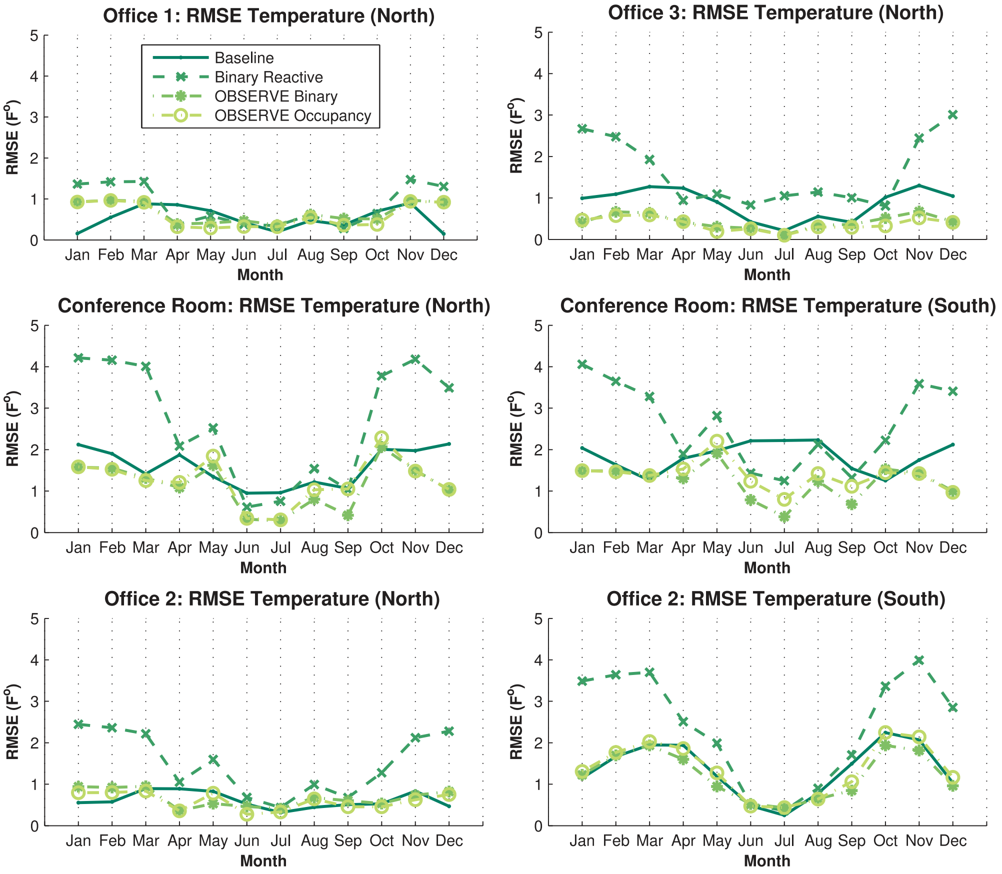


Fig. 14. Monthly RMSE temperature for each strategy for Fresno.

8.3.1. Temperature Effectiveness. We examine the temperature of the rooms during all times of occupancy. In particular, we are interested in areas such as the Conference Room, which are not occupied the majority of the day. In order to be ASHRAE [ASHRAE 2007a] compliant, we must maintain the set-point temperatures to ensure $-0.5 \leq PMV \leq 0.5$. As mentioned before, we examine temperature directly rather than PMV; our goal is to meet the temperature set by any metric of thermal comfort. For this analysis, we examine the root mean square error (RMSE) of the room temperatures (Figure 14). We also examine building orientation to observe the influence of solar gain.

All four strategies perform similarly for Office 1. Reactive Binary has the highest RMSE (1.3°) for Nov-Mar. Both OBSERVE strategies have similar results throughout the year. Baseline performed best during the coldest months.

OBSERVE Occupancy performs the best out of the four strategies for Office 3 with a RMSE of 0.1–0.3 F°. OBSERVE Binary has a slightly higher RMSE of 0.1–0.5 F°. The Reactive Binary strategy is highest with a RMSE of 0.9–2.8 F°. For Dec-Jan, Reactive Binary has a RMSE of 2.5 F° as compared with the RMSE of 0.1 F° and 1.0 F° for OBSERVE Occupancy and baseline, respectively.

Two different building orientations are considered for Conference Room and Office 2 to show the effect of solar gain. North is defined in Figure 1. South refers to the building turned 180°. Reactive Binary has higher RMSE for the south orientation

(0.4–3.6 F°) than the north orientation (0.2–2.4 F°). Both OBSERVE strategies have lower RMSE than Reactive Binary for both orientations. Generally, both OBSERVE strategies have slightly lower RMSE than baseline for the north orientation. Both OBSERVE strategies show higher RMSE for the south orientation than the north orientation; this is caused by solar gain.

For the Conference Room, Reactive Binary has higher RMSE during the colder months (4.0–4.2 F°) for both orientations. During the summer, Reactive Binary has higher RMSE for the south orientation. For the north orientation, the baseline does better than OBSERVE strategies for winter and spring. During the summer, however, both OBSERVE strategies have lower RMSE than baseline. Comparing OBSERVE strategies, OBSERVE Binary has lower RMSE than OBSERVE Occupancy for the summer south orientation (0.3–1.1 F° vs 0.8–1.2 F°).

8.3.2. Discussion. Reactive Binary tends to have higher RMSE than the other strategies because of the slow ramp-up to target temperature. Once an occupant enters a room, it takes time to reach the target temperature. For areas occupied most of the time such as Office 1, Reactive Binary performance improves but does not match quality of service of predictive strategies since there is still a temperature reaction delay at the beginning of the day when the first occupant enters. This delay becomes more critical for areas sporadically occupied, such as Office 2 and the Conference Room. A room occupied once or twice a day, such as the Conference Room, has insufficient time to reach the target temperature and is only at the correct temperature if the outside conditions happens to match the target temperature (spring, fall). RMSE increases if the room receives afternoon sun, as is shown for the southern orientation of the Conference Room and Office 2. Rooms such as Office 2 that are constantly visited have slightly lower RMSE since the thermal momentum creates an additive effect that helps maintain temperature. These results show that reactive strategies based on a PIR WSN do not work in practice and cannot reach the target temperature required by the users for all the different scenarios tested.

In some cases, OBSERVE strategies perform better than the baseline since the baseline uses a static HVAC schedule. This is seen for Office 3, which is occupied by professors and graduate students working at odd hours. The static baseline schedule stops conditioning during normal working hours whereas OBSERVE anticipates after hours usage. Even Reactive Binary out-performs the baseline during mild months (spring and fall) when rooms are occupied off-hours and the outside and target temperatures are close.

Both OBSERVE strategies perform similarly. However, there are instances in warm weather where OBSERVE Binary performs slightly better. This is related to over ventilation. Assuming maximum occupancy forces the VAV vent to open fully when conditioning a room and less time is required to cool the room. The same is not true for heating since air from the loop is only partially preheated.

8.3.3. Ventilation Effectiveness. The minimum ventilation rate is established by Eq. (4). Since the occupancy detection system can undercount occupants, we add an additional 10% to occupancy estimates. Figure 15 shows the ventilation rate for a summer day. The baseline strategy assumes a maximum ventilation rate. The binary strategies use a maximum ventilation rate only when an area is occupied. OBSERVE Occupancy uses a modified schedule. It uses maximum ventilation during cold months (Oct-Apr for Fresno/Chicago) and optimal ventilation during the warm months (year-round for Miami). From the summer day ventilation, we can see that the baseline greatly overventilates the building. While binary-based strategies are an improvement, we can see that binary ventilation rate far exceeds the required ventilation rate. It has a RMSE

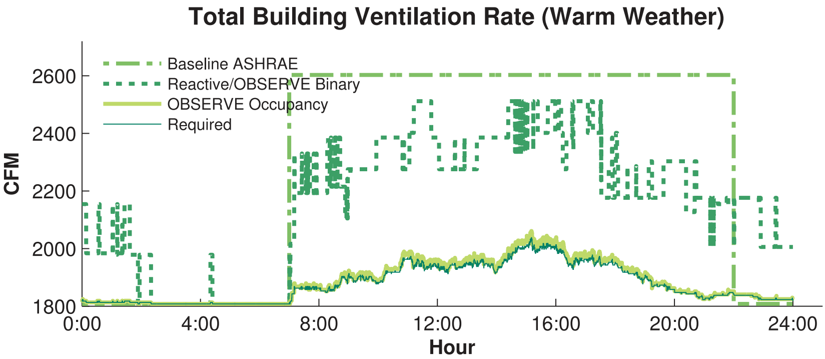


Fig. 15. Total building ventilation rates for each strategy for one particular summer day.

of 335.0 CFM (Normalized RMSE of 146%). OBSERVE however, remains close to the required ventilation rate and has a RMSE of 9.5 CFM (NRMSE of 4.1%). OBSERVE is usually above the required ventilation because of the 10% safety margin for undercounting. OBSERVE fell below the required ventilation rate only 0.0004% of the time. From Oct-Apr, both binary and OBSERVE share the same ventilation rates and have a NRMSE of 146%.

9. RELATED WORK

Few works on occupancy modeling have been published that are relevant to demand control systems. Most acknowledge that a major obstacle for developing occupancy prediction models is lack of data. It is time-consuming to gather ground truth occupancy data even for a small set of rooms.

The simplest and most commonly used occupancy models are sets of predefined static coefficients that are multiplied with a maximum room occupancy. To estimate the occupancy of a room at 8 am given that the room has a maximum occupancy of 30, the coefficient for 8 am is simply multiplied with 30. Different sets of coefficients are used, depending if it is a weekday, weekend, or holiday and the purpose of the building. ASHRAE Standard 90.1 [ASHRAE 2007c], BLAST [Herron et al. 1983], and DOE-2 [DOE 2012] define several occupancy profiles for office buildings daytypes. These models are commonly used for energy simulation tools such as eQuest [eQuest Building Energy Analysis Tool] or EnergyPlus during the design phase of buildings and are also used to determine static conditioning strategies for buildings.

Several methods of modeling occupancy do so using multiple sources of sensory input. Abushakra and Claridge [2001] show how occupancy can be modeled using linear regression models. Data was collected for lighting and equipment loads and occupancy is estimated using a walkthrough survey of the building. Using the electrical loads as the indicator variables and occupancy estimations as the response variable, linear regression models for weekdays, weekends, and holidays were created. The main limitation of this model is the reliance on energy usage to determine if someone is present. In general, it will tend to underestimate occupancy levels. For example, occupants attending a large group meeting in a conference room may not be adding any additional loads causing the model to report the room empty. Occupancy prediction for conditioning is also difficult as it would first require the ability to predict electrical loads.

Agarwal et al. [2011] utilize a deployment of PIR and door sensors to obtain a binary indication of occupancy. They use a reactive strategy that adjusts the temperature based on current occupancy and estimate-potential savings using EnergyPlus. This ignores the ramp-up time required for a room to be brought to temperature. The paper also does not account for the impact of ventilation on energy savings.

Lu et al. [2010] also utilize door and PIR sensors for binary detection of occupancy for residential buildings and examine reactive and predictive controls strategies. The predictive strategy is achieved using a Hidden Markov Model. The model estimates the probability of home being in one of three states: unoccupied, occupied with an occupant awake, and occupied with all occupants asleep. The authors provide results for real-world deployment and EnergyPlus simulations. However, the results do not take into account ventilation, which can have a significant impact on energy efficiency. Also, the modeling approach does not account for different daily schedules. If a person stays out late on Friday on a semi-regular basis, the model would not be able to predictively condition for this scenario.

Dodier et al. [2006] propose using a belief network for occupancy detection within buildings. The authors use multiple sensory input to probabilistically infer occupancy. By evaluating multiple sensory inputs, they determine the probability that a particular area is occupied. In each office, PIR and telephone on/off hook sensors were used to determine if rooms are in occupied states. The authors model the occupied state of individual rooms with a Markov Chain, where the transition matrix probabilities are calculated by examining the exponential distribution of the sojourn times of the observed states. While these strategies are more suitable for predictive demand control strategies, there are limitations that diminish their usefulness. The strategies are aimed for modeling occupancy for individual offices and cannot be applied to spaces with larger occupancies. Though the multiple telephone sensors could potentially be used to help determine a lower bound of occupancy, it is difficult to determine when someone leaves since the presence of multiple people essentially nullifies the usefulness of the PIR sensor. The approach does not consider room interdependences and only focuses on occupancy detection.

Through proper framing of the problem, it is possible to create prediction models using a classification framework. Scott et al. [2011] developed a prediction-based conditioning strategy for residential buildings using *k*-nearest-neighbors (KNN) to predict occupancy. The use of KNN creates several drawbacks. KNN works best if the number of days in the dataset is small since KNN is simply a comparison among all points in the dataset. If the dataset of days is large, then the time required to compute the prediction may be prohibitively long.

In our previous work [Erickson et al. 2009], we developed an agent-based model (ABM) and a multivariate Gaussian model. The ABM simulates occupancy by modeling the behavior of the individual. Agents are given paths, walking speed, and itineraries based on the occupancy changes seen in the training data. Occupancy is simulated by creating multiple agents that follow probabilistically generated instructions. Based on their simulated movement, room occupancies over the course of the day can be estimated. While this is useful for estimating daily occupancy changes that are possible, it is difficult to use this model to predict future room usage. We discussed these models in Section 3.

Liao and Barooah [2011] developed a stochastic agent-based model. Rather than creating a variety of precomputed schedules based on a variety of parameters, the authors took into account the state of occupancy in the prior state. The main drawback to this approach is the difficulty of training such a model. An agent-based model requires detailed information at the individual occupant level. In this case, the only feasible method of collecting such data was for the authors to give surveys; this method did not have a system-based approach for gathering individual movement dynamics. This can potentially have a large amount of error, leading to inaccuracy in the model.

10. CONCLUSIONS AND FUTURE WORK

We propose a statistical model of the temporal occupancy of a building based on an inhomogeneous Markov chain estimated from occupancy data collected from a sensor network. We control the complexity of the state space by using only states observed during training, and prevent discontinuities in the chain by a closest distance Markov chain (CDMC) approach or by continuously blending the transition matrices over time (BMC). Comparing samples generated from the model with ground-truth data using various metrics shows that the model captures the occupancy dynamics accurately. We have also shown how the model can be integrated with a WSN for demand-control-based conditioning strategies. We propose the OBSERVE predictive demand control strategy and test the energy savings and conditioning performance. We learned several lessons from our results. First, in order to achieve energy savings, real-time occupancy data is critical. This can be seen by the average 42% annual energy savings compared to the current state-of-the-art baseline strategy. Second, predictive strategies show better energy savings performance and even significantly better quality of service conditioning than reactive strategies. Finally, in order to achieve maximum energy savings, actual level of occupancy is required in order to optimize ventilation levels. For future work, we plan to deploy a new occupancy estimation system on two floors and train the model in real time over longer period of time. Once deployed, we can integrate OBSERVE into the HVAC system and refine the OBSERVE algorithm to optimally determine precondition times.

ACKNOWLEDGMENTS

Special thanks to Ankur Kamthe for valuable discussions of the MC models and OBSERVE, J. Lindblom, P. Felkai, D. Burch, A. Magnana and M. Torio for processing the ground truth data.

REFERENCES

ASHRAE. 2007a. ASHRAE standard 55: Thermal environmental conditions for human occupancy. American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc.

ASHRAE. 2007b. ASHRAE standard 62.1: Ventilation for acceptable indoor air quality. American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc.

ASHRAE. 2007c. ASHRAE standard 90.1: Energy standard for buildings except low-rise residential buildings. American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc.

EIA. 2010. EIA-Energy Information Administration. http://www.eia.doe.gov/.

DOE. 2012. DOE-2 - building energy analysis tool and cost analysis tool. http://www.doe2.com/DOE2.

B. Abushakra and D. Claridge. 2001. Accounting for the occupancy variable in inverse building energy baselining models. In *Proceedings of the 1st International Conference for Enhanced Building Operations (ICEBO 2001)*.

Y. Agarwal, B. Balaji, S. Dutta, R. K. Gupta, and T. Weng. 2011. Duty-cycling buildings aggressively: The next frontier in HVAC control. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'11)*. ACM/IEEE, 246–257.

M. J. Brandemuehl and J. E. Braun. 1999. The impact of demand-controlled and economizer ventilation strategies on energy use in buildings. *ASHRAE Trans.* 105, 2.

B. Burke and M. Keeler. 2009. *Fundamentals of Integrated Design for Sustainable Building*. Wiley.

R. H. Dodier, G. P. Henze, D. K. Tiller, and X. Guo. 2006. Building occupancy detection through sensor belief networks. *Ener. Build.* 38, 9, 1033–1043.

eQuest Building Energy Analysis Tool. http://www.doe2.com/.

V. Erickson, S. Achleitner, and A. E. Cerpa. 2013. POEM: Power-efficient occupancy-based energy management system. In *Proceedings of the 12th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'13)*. ACM/IEEE, 203–216.

V. Erickson, M. A. Carreira-Perpinan, and A. Cerpa. 2011. OBSERVE: Occupancy-based system for efficient reduction of HVAC energy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'11)*. ACM/IEEE, 258–269.

V. Erickson and A. Cerpa. 2010. Occupancy-based demand response HVAC control strategy. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building (BuildSys'10)*. ACM, New York, 7–12.

V. L. Erickson, Y. Lin, A. Kamthe, R. Brahme, A. Cerpa, M. D. Sohn, and S. Narayanan. 2009. Energy efficient building environment control strategies using real-time occupancy measurements. In *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys'09)* in conjunction with *ACM SenSys '09*. ACM, 19–24.

W. J. Fisk, D. Faulkner, and D. P. Sullivan. 2006. Accuracy of CO2 sensors in commercial buildings: A pilot study. Tech. Rep. LBNL-61862, Lawrence Berkeley National Laboratory.

D. Herron, J. Eidsmore, R. O'Brien, and D. Leverenz. 1983. Use of simplified input for BLAST energy analysis. Tech. Rep. E-185. Construction Engineering Research Laboratory, U.S. Army, Corps of Engineers, Champaign, IL, 1–110.

A. Kamthe, L. Jiang, M. Dudys, and A. Cerpa 2009. SCOPES: Smart cameras object position estimation system. In *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN'09)*. Springer-Verlag, Berlin, 279–295.

C. Liao and P. Barooah. 2011. A novel stochastic agent-based model of building occupancy. In *Proceedings of the IEEE American Control Conference (ACC'11)*. IEEE, 2095–2100.

R. Lienhart and J. Maydt. 2002. An extended set of Haar-like features for rapid object detection. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2002)*. IEEE, I-900–I-903 vol. 1.

J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. 2010. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*. ACM, 211–224.

B. W. Olesen and K. C. Parsons. 2002. Introduction to thermal comfort standards and to the proposed new version of EN ISO 7730. *Ener. Build*. 34, 6, 537–548.

OpenCV. http://opencvlibrary.sourceforge.net/.

J. U. Pfafferott, S. Herkel, D. E. Kalz, and A. Zeuschner. 2007. Comparison of low-energy office buildings in summer using different thermal comfort criteria. *Ener. Build.* 39, 7, 750–757.

J. Ploennigs, B. Hensel, and K. Kabitzsch. 2010. Wireless, collaborative virtual sensors for thermal comfort. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building (BuildSys'10)*. ACM, New York, 79–84.

M. Schumann, A. Burillo, and N. Wilson. 2010. Predicting the desired thermal comfort conditions for shared offices. In *Proceedings of the International Conference on Computing in Civil and Building Engineering (ICCCBE-10)* (Nottingham, UK).

J. Scott, A. J. B. Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar. 2011. Preheat: Controlling home heating using occupancy prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp 2011)* (Beijing, China). 281–290.

P. Viola and M. Jones. 2001. Robust real-time object detection. In *Int. J. Comput. Vis*. 57, 2, 137–154.

Received August 2012; revised June 2013; accepted May 2013