Abstract—One of the primary energy consumers in buildings are the Heating, Ventilation, and Air-Conditioning (HVAC) systems, which usually operate on a fixed schedule, i.e., running from early morning until late evening during the weekdays. This fixed operation schedule does not take the dynamics of occupancy level in the building into consideration, therefore may lead to waste of energy. An estimate of the number of occupants in the building with time can contribute to improving the control policy of the building’s HVAC system by reducing energy consumption. In this paper, the auto-regressive hidden Markov model (ARHMM), is investigated to estimate the number of occupants in a research laboratory in a building using a wireless sensor network deployed. The network is composed of stand-alone sensing nodes with wireless data transmission capability, a base station that collects data from the sensing nodes, and a server to analyze the data from the base station. Experimental results and numerical simulation demonstrate that the ARHMM is more effective in estimating the number of occupants in the laboratory than the HMM algorithm, especially when the occupancy level fluctuates frequently.

I. INTRODUCTION

Buildings account for nearly 40% of global energy consumption [1], and 43.4% and 38% of the energy has shown to be consumed by commercial and residential buildings, respectively [2]. Most of these buildings are conditioned on fixed schedules, such as performing heating/cooling, ventilation, and air-conditioning (HVAC) from early morning until late evening during weekdays. However, this policy may lead to a significant waste in energy consumption, because the HVAC schedule does not consider the actual conditions of the building. In light of increasing energy costs and government mandates for energy efficiency [3], increasing the energy efficiency of buildings is of great interest. Some researchers have proposed the control and use of shading blinds and natural ventilation to make effective use of the available natural resources to reduce building energy consumption [4]-[5], while others have proposed optimum control of HVAC schedules [6]. The above-mentioned methods do not consider the actual number of occupants in the building, while it has been shown that annual energy savings of 10%-42% can be achieved if a proper HVAC strategy that accounts for actual occupancy levels [7].

Since the occupancy level cannot be readily obtained directly by using sensors or imaging from cameras due to communication bandwidth and privacy concerns [8], accurate and efficient estimation of occupancy level in the building become one of the key problems for smart buildings. In [2], inexpensive magnetic reed switch and passive infra-red (PIR) sensors were used to monitor the occupants of a building and guide the operation of the HVAC system. However, this direct sensing method, utilizing magnetic reed switches and PIR sensors, can only provide information of whether the building is occupied or not, and cannot provide the number of the occupants in the building. In addition, detection error associated with PIR sensors can occur when the occupant is performing common stationary tasks in the office space, such as talking or writing. Page developed a stochastic model of an occupant present state based on an inhomogeneous Markov Chain, interrupted by occasional periods of long absence. This model can generate a time series of the states of presence of each occupant for each space in the building [9], but the accuracy is limited when the vacancy intervals follow an exponential distribution, since the distribution of occupancy intervals is time varying. In [10], the authors assumed that the occupants affect the indoor environment by emitting CO$_2$, heat, moisture and sound, and therefore these indoor environmental variables were combined to measure the environmental conditions and infer the occupancy level. In [11], by using the data from CO$_2$ sensors, acoustic sensors, and the PIR sensors, three machine learning methods ANN, SVM and HMM were investigated to estimate occupancy in an office. However, these environmental variables are correlated with their respective previous measurements. For example, the measurement of CO$_2$ in the room by a CO$_2$ sensor is not only dependent on the occupancy at the present time, but also the previous state measurement and its first order derivative [12] due to air flow dynamics when the occupancy changes. Similar situation also apply to temperature and humidity measurement. In addition, since all of these environmental variables are correlated with the occupancy, they are correlated with each other. Therefore, when the environmental variables are correlated, all three of these machine learning methods lose their ability to accurately estimate occupancy. In [8], the authors took the correlation of data into consideration, and proposed the ARHMM method to estimate the number of occupants.

In this paper, we continue the work done by Han [8], and present a method to derive the coefficients of the autoregressive part of the ARHMM. We also set up a wireless sensor network in a research laboratory to conduct experiments to validate the effectiveness of our proposed method. In section II, we will discuss details about the estimation methods, in section III, a sensor network scenario is introduced, an experimental study is outlined in section IV, and the conclusion is given in section V.
II. OCCUPANCY ESTIMATION ALGORITHMS

In this section, we discuss the HMM and ARHMM models and their application to occupancy estimation.

A. The Hidden Markov Model

Since each state does not necessarily correspond to an observable (physical) event in many scenarios, the Markov Models are restrictive in this application. If the observation is a probabilistic function of the state, i.e., a Hidden Markov Model (HMM), then the model is a double embedded stochastic process with an underlying stochastic process which is not observable, thus hidden, but can be estimated from other stochastic processes, such as observations. HMM became popular following its introduction by Baum and colleagues [13]-[15] in the late 1960s. The richness of this mathematical model forms the theoretical basis for a variety of applications, especially in speech recognition [16]-[17].

Assume that a HMM has $N$ states, and denote the state at time $t$ as $S_t$, which belongs to $\{1, 2, \ldots, N\}$. The initial distribution of the HMM is denoted as $\pi = \{\pi_i, i = 1, 2, \ldots, N\}$. The state transition matrix between two successive time periods is expressed as $A$, $y_i \in \mathbb{R}^k$ are $k$ observations at time $t$, then $b(S) = P(y_t | S_t = j)$ denotes the probability of observation $y_t$ at time $t$ when the system is in state $j$. Let $B = \{b(S_j), S_j = 1, 2, \ldots, N\}$, then the parameter of the HMM can be denoted as $\lambda = \{\pi, A, B\}$.

If we make two assumptions: first, the $t^{th}$ hidden state, given the $(t-1)^{th}$ hidden state, is independent of previous states or observations; second, the $t^{th}$ observation, given the $t^{th}$ hidden state, is independent of other variables. Then, we have following three matters to address for the HMM [18]:

Matter 1: given a model $\lambda = \{\pi, A, B\}$ and the observation sequence $Y = y_1, y_2, \ldots, y_T$, how can $P(Y | \lambda)$ be computed efficiently?

This is a model evaluation problem. For each $t$, there are $N$ possible states, and for $t = 1, 2, \ldots, T$, there are $N^T$ possible state sequences. In addition, for each sequence, there are $2T$ multiplications and $N^T$ additions. Therefore, the computation of $P(Y | \lambda)$ requires $(2T+1)N^T$ calculations if we use the exhaustive enumeration method, which is time consuming. The solution to this problem is to solve by a Forward Algorithm or a Backward Algorithm, both of which reduce the calculations into $N^T$.

Matter 2: given a model $\lambda = \{\pi, A, B\}$ and the observation sequence $Y = y_1, y_2, \ldots, y_T$, how can the corresponding optimal state sequence $S = S_1, S_2, \ldots, S_T$ be chosen?

This is a path decoding problem and the solution is through the Viterbi Algorithm.

Matter 3: how can the model $\lambda = \{\pi, A, B\}$ be tuned to maximize $P(Y | \lambda)$?

This is a model training problem, and there is no analytical solution, but the Baum-Welch re-estimation (Expectation Maximization) algorithm can be used to find the local optimal solution [18].

B. The Hidden Markov Model Applied to Occupancy estimation

Assume that the occupancy level of a room can be modeled by a Markov Chain with $N + 1$ states, such that the number of the people in the room varies from 0 to $N$. However, we do not know directly the number of the occupants at any time, that is, the states are hidden. There are various kinds of sensors installed in the room to detect the CO2 level, relative humidity, and temperature of the room, and the data from these sensors are available anytime, i.e., they are observable. Therefore, the problem of estimating the occupancy number can be modeled as a HMM with $N + 1$ states and the observation sequences are data from CO2, RH and Temperature sensors.

In order to estimate the occupancy number of the room we have to complete the following three steps:

The first step is to build individual models for different occupancy numbers. This is accomplished by using the solution of previously mentioned Matter 3.

The second step is to refine the models which have been made. This can be accomplished by using the solution of Matter 2 discussed previously.

Once the set of $N$ HMMs has been obtained and optimized, then the estimation of the occupancy number in the room can be achieved by using the solution of Matter 1 to score each model based on the given test observation sequences, and the select the number of occupants whose model has the highest score.

C. The Auto-Regressive Hidden Markov Model

As pointed out by Wang in [12], the data obtained from the sensors are usually correlated, therefore this relationship should be taken into consideration when we build our models, namely, ARHMM. We assume that $\{c_i(i), i = 1, 2, \ldots, p\}$ are the coefficients of a $p^{th}$ order autoregressive (AR(p)) process at time $t$. $W_\lambda$ is an independent identically distributed sequence of innovation for this AR(p) process and its variance is $\sigma^2_\lambda$. Let $\theta_\lambda = \{c_i(i), i = 1, 2, \ldots, p\}$ be the parameters of this AR(p) process at time $t$. Then the observations can be written in the following form:

$$y_t = \sum_{i=1}^{p} c_i(i) \cdot y_{t-i} + W_\lambda$$  \hspace{1cm} (1)

Let $y_{ij} = \{y_i, y_{i+1}, \ldots, y_j\}$ for $i < j$ and $b\{y_i, y_{i-1}, \theta_\lambda\}$ denote the conditional density of observation $y_i$, given the parameters of $\theta_\lambda$ and the previous observations $y_{i-1}$. Practically, initial values, e.g. $c_0(i) = 1$, are provided for Eq (1). The parameters were updated for each step based on the measured data and ground truth occupancy manually measured. Let $\theta = \{\theta_\lambda, S = 1, 2, \ldots, N\}$ then the parameters of the switching ARHMM can be denoted as $\lambda = \{\pi, A, \theta\}$. 2235
Suppose that some initial conditions \( y_{0-p+1} = \{y_{-p+1}, y_{-p+2}, \ldots, y_0\} \) are given then the conditional n-dimensional density of \( y^p \) is

\[
P(y^p | y_{-p+1}, \lambda) = \sum_{y^p} P(y^p, S^p | y_{-p+1}, \lambda) \tag{2}
\]

where

\[
P(y^p, S^p | y_{-p+1}, \lambda) = \prod_{i=1}^{k} a_{S_{i-1}, S_{i}} \cdot b(y_i | y_{-p+1}, \theta_{S_{i}}) \tag{3}
\]

And \( S^p = \{S_1, S_2, \ldots, S_n\} \) is the state sequence corresponding to \( y^p \); \( b(y_i | y_{-p+1}, \theta_{S_{i}}) \) is the observation conditional density, with mean \( \sum_{i=1}^{m} c_i \cdot y_{-i} \) and variance \( \sigma_i^2 \). The proof of Eq. (2) is provided in Appendix A.

If we denote the coefficients of the AR(p) process at state \( j \) as \( C_j^p = \left[ c_{1}(j), c_{2}(j), \ldots, c_{p}(j) \right] \), and the probability of the state in \( j \) at time \( t \), given the observation \( y^p \) is:

\[
q_j(t) = \frac{P(S_t = j | y^p, \lambda)}{\sum_{i=1}^{n} P(S_t = j | y^p, \lambda)} \tag{4}
\]

The probability of the state in \( j \) given the observation \( y^p \) is:

\[
q = \sum_{i=1}^{n} q_j(t) \tag{5}
\]

then we have the following theorem:

Given the observation \( y^p \), the coefficients and variance for the AR(p) process are:

\[
C_j^p = -d_j \cdot (D_j)^{-1} \quad \text{and} \quad \sigma_j^2 = \left[ 1 \quad C_j^p \right] \cdot R_j \cdot \left[ 1 \quad C_j^p \right]^{-1}, \quad \text{where}
\]

\[
C_j^p = \left[ c_{1}(j), c_{2}(j), \ldots, c_{p}(j) \right], \quad \rho_j = \left( \sum_{i=1}^{n} q_j(t) \cdot y_{i} \right),
\]

\[
D_j^p = \left[ \sum_{i=1}^{n} q_j(t) \cdot y_{i} \cdot y_{i-1} \right] \quad \sum_{i=1}^{n} q_j(t) \cdot y_{i} \cdot y_{i-2} \ldots \sum_{i=1}^{n} q_j(t) \cdot y_{i} \cdot y_{i-p}, \quad \text{and}
\]

\[
E_j = \left[ \sum_{i=1}^{n} q_j(t) \cdot y_{i} \cdot y_{i-1} \right] \quad \sum_{i=1}^{n} q_j(t) \cdot y_{i} \cdot y_{i-2} \ldots \sum_{i=1}^{n} q_j(t) \cdot y_{i} \cdot y_{i-p}.
\]

The proof of Eq. (4) is provided in Appendix B.

After these parameters are obtained, the Baum-Welch re-estimation (Expectation Maximization) algorithm was applied to find the local optimal solution for the local optimal models of the ARHMM. Detailed procedures of the Expectation Maximization algorithm can be found in the research [19] and [20].

III. ALGORITHMS IMPLEMENTATION IN A SENSOR NETWORK

A wireless sensor network was set up in a research laboratory of the first floor of a three-story building at the University of Connecticut. The network is composed of three parts: stand-alone sensors and wireless measurement nodes, a base station deployed to collect the data from the nodes, and a server to analyze the data from the base station. Seven types of sensors: passive infra-red (PIR) sensors, carbon dioxide (CO2) sensors, temperature sensors, relative humidity (RH) sensors, air-velocity sensors, global thermometer, and reed switches, were deployed in different places throughout the laboratory.

Four PIR sensors were mounted on the ceiling of both the front and back doors of the laboratory. For each door, one PIR sensor with a wide detection angle was installed facing the interior of the room to detect the occupants’ movement in the room, the other PIR sensor with a narrow detection angle was installed facing the floor to detection the occupants’ status near the door. Due to the sensitivity of PIR sensors, we also installed reed switches for each door to avoid incorrect information provided by false triggering of PIR sensors. Combining the various triggering time sequences of the PIR sensors and reed switches, we can obtain the occupants’ direction of movement in the room.

Four CO2, Temperature and RH sensors were distributed in the places where the occupants are most concentrated. Two air-velocity sensors were installed near both doors, and one global thermometer was installed in the center of the room. Since the occupants emit CO2 and moisture in the room through exhaled air, and movement of the occupants causes vibration in the air-velocity, and different numbers of occupants will have differing effects on the temperature and radiant temperature in the environment, the data collected from these sensors may reveal the occupancy level of the room.

IV. SENSOR NETWORK EXPERIMENT AND RESULTS

An evaluation of the wireless sensor network was performed through experiment in the research laboratory. The occupants in the lab who work from 9:30am until 6:00pm on weekdays will come and go during the day, creating a dynamic occupancy pattern. To provide a ground truth for the study, the occupancy of the lab was manually recorded during the five day work week over two periods of testing. The manually recorded occupancy data could then be compared with the result from our algorithm. In the experimental study, we targeted at small offices where the occupancy level is below 10. As compared with rooms of higher occupancy levels, such as auditorium or conference rooms where the occupancy can reach 100 or above, the low level occupancy has high variance in the measured data, since any single occupancy increase/decrease will lead to significant changes in the overall occupancy level, therefore requiring higher level of stability in the occupancy detection algorithm to maintain estimation accuracy and robustness.

Experiment 1 was done when the lab contained 6 occupants. We choose one work day’s data as the training data set, and any other day’s data as testing data. For all the models, the number of the state is 7, since we have 6 occupants, and the number of occupants in the room varies from 0 to 6. The time steps for the sensors sampling data are all 20 seconds. Figs. 1 and 2 show the estimation result of a particular day by using the HMM and the ARHMM algorithms respectively.

We define the estimation accuracy as the total number of correct estimations divided by the total number of estimations, and the estimation accuracy by using the HMM and the
ARHMM algorithms are listed in Table 1. The estimation accuracy obtained by using the ARHMM algorithm shows a 9% improvement over the estimation results obtained using the HMM algorithm. The standard deviation of the estimation error, which reveals the variation of estimation error, by the two methods are also listed in Table 1. We can see that by using the ARHMM algorithm, the estimation error deviation of the ARHMM has 14.4% improvement over that of the HMM. Therefore, the ARHMM estimation method is better than the HMM estimation method in detecting the number of occupants. In addition, from the figures, we can see that when the number of occupants changes in a relatively short period, that both algorithms may fail to estimate the actual occupant profile, with a larger error for the HMM algorithm.

In Experiment 2, the lab contained 10 occupants who followed the same work schedule as in Experiment 1. However, the occupancy level changed much more frequently than that of Experiment 1. In this experiment, we set the sampling time of sensors to 5 seconds to overcome the delayed response of the reed switches and PIR sensors. From Table II, we can see that by using the HMM algorithm, the estimation accuracy is only 25.2%, which is not satisfactory, and the HMM failed in estimating the number of occupants for this case. From Figure 3, we can also easily notice that even when the actual number of occupants is 0, such as in the evening, for a long period of time, the HMM algorithm still cannot track the occupancy level. However, if we consider the correlations of the observed data, then by using the ARHMM algorithm, the estimation accuracy is 80.1%, which is a 68.5% improvement over the result obtained by using the HMM algorithm. The standard deviations of estimation error by using the HMM and the ARHMM algorithms are also listed in Table II, from which we know that the ARHMM algorithm has a 99.1% improvement over the HMM algorithm in occupancy estimation, and this verifies that the ARHMM is still valid in estimating the number of occupants in this case. From Fig. 4, we can see that although the ARHMM is not efficient in capturing dynamic changes of the occupancy level, it is good for tracking the occupant profile when the lab is vacant.

TABLE I. ESTIMATION RESULTS BY USING THE HMM AND ARHMM ALGORITHMS: EXPERIMENT 1

<table>
<thead>
<tr>
<th>Estimation Method</th>
<th>Estimation Accuracy</th>
<th>Estimation Error Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>76.2%</td>
<td>1.003</td>
</tr>
<tr>
<td>ARHMM</td>
<td>84.0%</td>
<td>0.859</td>
</tr>
</tbody>
</table>

TABLE II. ESTIMATION RESULTS BY USING THE HMM AND THE ARHMM ALGORITHMS: EXPERIMENT 2

<table>
<thead>
<tr>
<th>Estimation Method</th>
<th>Estimation Accuracy</th>
<th>Estimation Error Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>25.2%</td>
<td>2.34</td>
</tr>
<tr>
<td>ARHMM</td>
<td>80.1%</td>
<td>0.02</td>
</tr>
</tbody>
</table>

The possible reason for the differing estimation results for these two experiments may be that the data we used as the observations of the model contains the CO2 data. This data is correlated, since the data from the sensor is calculated based on previous data [11], and this correlation is enhanced when the sampling interval is narrowed.
V. CONCLUSION

In this paper we present the result of building occupancy level estimation based on the ARHMM algorithm [8]. We developed a method to derive the coefficients of the autoregressive part of the ARHMM, and analyzed data obtained by a wireless sensor network in a research laboratory to evaluate the effectiveness of the developed method. We found that by taking correlation of observation data into consideration using the ARHMM method, the estimation results are consistently better than using the conventional HMM method. This is especially true when the number of occupants changes frequently in a relatively short period of time. The estimation results show that for the case when the occupancy level does not change frequently, the ARHMM estimation method has a 9% improvement over the HMM estimation method. However, for the case when the occupancy level changes frequently in a short period, the estimation result by using the AHMM method has a 68.5% improvement over the HMM method.

ACKNOWLEDGMENT

This work has been supported by the National Science Foundation under grant CNS-1239030. Editorial assistance by Donald Karg is appreciated.

VI. APPENDIXES

Proof of Theorem

Proof: By using the mathematical induction, we can obtain that:

\[ P(x^{s-1}, s-2) = \sum_{i=0}^{n} P(y_s, s_i), P(x^{s-1}, s_i) \cdot P(x^{s-2}, s_i) \]

From the above equation, we know that Theorem 1 holds.

Then

\[ w_s = y_s + \sum_{i=1}^{k} c_s(i)y_{s-i} \]  

(A.1)

\[ \sigma_j = E[w_j \cdot w_j' | S_j = j] \]

\[ = P(S_j = j) \cdot E(w_j \cdot w_j') \]

\[ = \sum_{i=1}^{k} q_j(i) \cdot [y_s + \sum_{i=1}^{k} c_j(i)y_{s-i}] \cdot [y_s + \sum_{i=1}^{k} c_j(i)y_{s-i}] \]

(A.2)

\[ = \sum_{i=1}^{k} q_j(i) \cdot \left[ \begin{array}{c} 1 \cdot c_j(1) \cdot c_j(2) \cdots c_j(p) \\ \vdots \\ y_{s-1} \\ y_{s-p} \end{array} \right] \]

\[ = \sum_{i=1}^{k} q_j(i) \cdot \left[ \begin{array}{c} 1 \cdot c_j(1) \\ \vdots \\ y_{s-1} \\ y_{s-p} \end{array} \right] \cdot \left[ \begin{array}{c} c_j(2) \\ \vdots \\ c_j(p) \end{array} \right] \]

(A.3)

Theorem 1 holds.

\[ R_j = \sum_{i=1}^{k} q_j(i) \cdot y_j \cdot y_{j-i} \cdot \sum_{i=1}^{k} q_j(i) \cdot y_j \cdot y_{j-i} \cdots \sum_{i=1}^{k} q_j(i) \cdot y_j \cdot y_{j-i} \]

(A.4)

If we partition the Matrix \( R_j \) into following parts:

\[ R_j = \begin{bmatrix} \rho_j & \rho_j' \\ D_j & D_j' \end{bmatrix} \]

(A.5)

where

\[ \rho_j = \sum_{i=1}^{k} q_j(i) \cdot y_j \cdot y_{j-i} \]

\[ D_j = \sum_{i=1}^{k} q_j(i) \cdot y_j \cdot y_{j-i} \cdot \sum_{i=1}^{k} q_j(i) \cdot y_j \cdot y_{j-i} \]

and
Then the following equation holds:

\[
E_j \cdot C_j^* = \left\{ \begin{array}{c}
\sum_{i=1}^{n} q(j) \cdot y_{i-1} \cdot y_{f-1} \\
\vdots \\
\sum_{i=1}^{n} q(j) \cdot y_{f-p} \cdot y_{f-1} \\
\end{array} \right\} \left\{ \begin{array}{c}
\sum_{i=1}^{n} q(j) \cdot y_{i-1} \\
\vdots \\
\sum_{i=1}^{n} q(j) \cdot y_{f-p} \\
\end{array} \right\} = \left\{ \begin{array}{c}
c_j(1) \\
\vdots \\
c_j(p) \\
\end{array} \right\}
\]

Therefore, \( C_j^* \) and \( \sigma_j^* \) are defined as follows:

\[
C_j^* = -D_j \cdot (E_j)^{-1} \quad \text{and} \quad \sigma_j^* = \left[ \begin{array}{c}
1 \\
\end{array} \right] \cdot R_j \cdot \left[ \begin{array}{c}
1 \\
\end{array} \right]
\]