ANALYSIS AND OPTIMIZATION OF BUILDING
ENERGY CONSUMPTION

JUN WEI CHUAH

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
ELECTRICAL ENGINEERING
ADVISER: NIRAJ K. JHA

SEPTEMBER 2013
Abstract

Energy is one of the most important resources required by modern human society. In 2010, energy expenditures represented 10% of global gross domestic product (GDP). By 2035, global energy consumption is expected to increase by more than 50% from current levels. The increased pace of global energy consumption leads to significant environmental and socioeconomic issues: (i) carbon emissions, from the burning of fossil fuels for energy, contribute to global warming, and (ii) increased energy expenditures lead to reduced standard of living. Efficient use of energy, through energy conservation measures, is an important step toward mitigating these effects. Residential and commercial buildings represent a prime target for energy conservation, comprising 21% of global energy consumption and 40% of the total energy consumption in the United States.

This thesis describes techniques for the analysis and optimization of building energy consumption. The thesis focuses on building retrofits and building energy simulation as key areas in building energy optimization and analysis. The thesis first discusses and evaluates building-level renewable energy generation as a solution toward building energy optimization. The thesis next describes a novel heating system, called localized heating. Under localized heating, building occupants are heated individually by directed radiant heaters, resulting in a considerably reduced heated space and significant heating energy savings. To support localized heating, a minimally-intrusive indoor occupant positioning system is described. The thesis then discusses occupant-level sensing (OLS) as the next frontier in building energy optimization. OLS captures the exact environmental conditions faced by each building occupant, using sensors that are carried by all building occupants. The information provided by OLS enables fine-grained optimization for unprecedented levels of energy efficiency and occupant comfort.
The thesis also describes a retrofit-oriented building energy simulator, ROBESim, that natively supports building retrofits. ROBESim extends existing building energy simulators by providing a platform for the analysis of novel retrofits, in addition to simulating existing retrofits. Using ROBESim, retrofits can be automatically applied to buildings, obviating the need for users to manually update building characteristics for comparisons between different building retrofits. ROBESim also includes several ease-of-use enhancements to support users of all experience levels.
Acknowledgments

This thesis represents several years of hard work. I am blessed to have received much support throughout the course of this work. To all who have helped make this thesis possible, I would like to express my sincere gratitude.

First and foremost, I would like to express my gratitude to the Agency for Science, Technology and Research, Singapore (A*STAR) and the people of Singapore for providing funding for both my undergraduate and graduate education. I will do my best to contribute to the country in any way I can.

A large part of who I am today is a direct result of the guidance provided by my adviser, Prof. Niraj K. Jha. His insight and attention to detail has made me a better researcher. His relentless thirst for knowledge has made me a better student. Most importantly, his focus on family values has made me a better husband and parent. I will forever cherish the advice that I have received from Prof. Jha.

I would also like to express my gratitude to Prof. Anand Raghunathan. He has been a key contributor to my research work. His creativity has helped in the pursuit of new research directions. His encouragement has helped me move on from numerous setbacks. I will strive to be as capable a researcher as Prof. Raghunathan.

I would like to thank my dissertation readers, Profs. Niraj K. Jha, Anand Raghunathan, and Elie Bou-Zeid, for their help in revising and improving upon this thesis. I would also like to thank Profs. Niraj K. Jha, Sigurd Wagner, and Naveen Verma for agreeing to be part of my thesis defense committee.

I would like to express my thanks to the administrators at the A*STAR Graduate Academy, especially Rinawati Rahmat and Jamie Tan, for dealing with all the administrative issues along the way. In addition, I would like to thank all the administrators at the Department of Electrical Engineering, Princeton University. In particular, I would like to thank Colleen C. Conrad, Sarah M. McGovern, Stacey A. Weber, Lori A. Bailey, and Roelie Abdi-Stoffers for their help throughout the last five years.
I have had the help and support of several friends during the course of my graduate work. I would like to thank them for being there for me along the way: Chunxiao Li, Najwa Aaraj, Wei Zhang, Ting-Jung Lin, Chia-Chun Lin, Chun-Yi Lee, Meng Zhang, Aoxiang Tang, Xianmin Chen, Debjit Bhattacharya, Wenhao Jia, Yavuz Yetim, Ozlem Bilgir Yetim, Carole-Jean Wu, Yu-Yuan Chen, Jakub M. Szefer, Yin Wang, and An Dai. I wish them the very best in the future. I am particularly grateful to Chunxiao Li for his help over the years. His critical thinking has helped me polish many of my research ideas. His dedication has helped make me a more hardworking person. I owe much of my development as a researcher to Chunxiao Li.

I have had immense support from my family in the past five years. I am grateful to my father Hock Seong Chuah, my mother Siew Huay Phua, my two brothers Jun Jie Chuah and Jun Yong Chuah, and my in-laws Tak Kwong Wong and Yuet Kwun Ho, for their support and encouragement in my graduate work. I would also like to express my appreciation to my daughter Christine Natalie Chuah, for giving me a few months of peace to work on this thesis.

Finally, I would like to dedicate this dissertation to the love of my life, my wife Kit Wan Michelle Wong. She has been my greatest supporter in the last few years and has been there for me the whole time. I look forward to our future together.
Contents

Abstract ................................................................. iii
Acknowledgments ...................................................... v
List of Tables ........................................................... xii
List of Figures .......................................................... xiii

1 Introduction ......................................................... 1
  1.1 Building Energy End-uses ...................................... 4
  1.2 Building Energy Optimization ................................. 6
    1.2.1 Renewable Energy Generation ............................ 6
    1.2.2 Fault Detection and Rectification ......................... 7
    1.2.3 Building Systems and Retrofits ............................ 8
  1.3 Building Energy Analysis ..................................... 10
    1.3.1 Building Energy Metering ................................ 11
    1.3.2 Building Energy Simulation ............................... 12
  1.4 Contributions of the Thesis ................................ 13
  1.5 Thesis Outline ................................................ 15

2 Related Work ....................................................... 16
  2.1 Building Energy Efficiency Approaches ...................... 16
    2.1.1 Heating, Ventilation, and Air Conditioning (HVAC) .... 17
    2.1.2 Lighting .................................................. 19
3 Renewable Energy Generation

3.1 Introduction .................................................. 26
3.2 Background .................................................. 28
3.3 Energy Generation Characteristics ................................................. 30
  3.3.1 Solar Energy ............................................. 30
  3.3.2 Wind Energy ............................................. 33
3.4 Evaluation .................................................. 35
  3.4.1 Solar Energy ............................................. 36
  3.4.2 Wind Energy ............................................. 41
3.5 Issues in Renewable Energy Generation ................................................. 44
3.6 Chapter Summary .................................................. 47

4 Localized Heating

4.1 Introduction .................................................. 50
4.2 Background .................................................. 53
4.3 Localized Heating System .................................................. 53
  4.3.1 System Outline ............................................. 54
  4.3.2 Control Algorithm ............................................. 56
  4.3.3 Thermal Comfort ............................................. 59
4.4 Occupant Positioning .................................................. 62
  4.4.1 System Outline ............................................. 62
6  ROBESim: A Retrofit-oriented Building Energy Simulator

6.1 Introduction .............................................. 109

6.2 Background .............................................. 113

6.3 Simulator Overview ..................................... 117
   6.3.1 Simulator Workflow ................................. 117
   6.3.2 Inputs .............................................. 119
   6.3.3 Outputs ............................................ 121
   6.3.4 Summary ............................................ 123

6.4 Simulator Usage Scenarios .............................. 124
   6.4.1 Building Owner ...................................... 124
   6.4.2 Retrofit Developer ................................. 125
   6.4.3 Additional Usage Scenarios ...................... 127

6.5 Retrofit Description and Management ............... 128
   6.5.1 Basic Modules ..................................... 128
   6.5.2 Retrofits Database .................................. 130
   6.5.3 Retrofit Processing ................................ 134
   6.5.4 Case Study: The Localized Heating Retrofit Module .................................. 137
   6.5.5 Summary ............................................ 139

6.6 ROBESim Components .................................. 139
   6.6.1 Simulation Manager ................................. 139
   6.6.2 Preprocessor ...................................... 140
   6.6.3 Co-Simulator ...................................... 141
   6.6.4 Postprocessor ...................................... 141

6.7 Input File Generators .................................. 143
   6.7.1 Building Model Generator ...................... 143
   6.7.2 Occupant Profile Generator .................... 145

6.8 Simulation Examples .................................. 147
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.8.1 Building Owner Usage Scenario</td>
<td>147</td>
</tr>
<tr>
<td>6.8.2 Parametric Analysis Example</td>
<td>149</td>
</tr>
<tr>
<td>6.8.3 Retrofit Developer Usage Scenario</td>
<td>150</td>
</tr>
<tr>
<td>6.8.4 Simulation Overhead Analysis</td>
<td>152</td>
</tr>
<tr>
<td>6.9 Chapter Summary</td>
<td>153</td>
</tr>
<tr>
<td>7 Conclusions and Future Research</td>
<td>155</td>
</tr>
<tr>
<td>Bibliography</td>
<td>160</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Baseline parameters for solar energy evaluation</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Baseline parameters for wind energy evaluation</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>The ASHRAE thermal sensation scale</td>
<td>60</td>
</tr>
<tr>
<td>4.2</td>
<td>Results of occupant positioning system evaluation</td>
<td>73</td>
</tr>
<tr>
<td>4.3</td>
<td>Parameters used in savings analysis</td>
<td>76</td>
</tr>
<tr>
<td>4.4</td>
<td>Return on investment (ROI) computations</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>Potential savings for localized heating vs. programmable thermostat</td>
<td>83</td>
</tr>
<tr>
<td>5.1</td>
<td>Simulation parameters</td>
<td>105</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 United States historical energy consumption and expenditures. 2
1.2 United States building energy end-use splits. 5
1.3 Building system types. 9

3.1 Average retail electricity prices in the United States, 1995-2012. 29
3.2 Solar map of the United States. Darker shading reflects higher solar radiation. 31
3.3 Solar radiation by month of year for selected locations. 32
3.4 Wind map of the United States. 35
3.5 ROI for solar panel compared to solar radiation at location. 37
3.6 ROI for solar panel compared to electricity prices. 38
3.7 ROI for solar panel compared to electricity prices at selected locations. 38
3.8 Solar installation area required to cover selected proportions of monthly electricity consumption. 40
3.9 ROI for wind turbine compared to annual mean wind speed. 42
3.10 ROI for wind turbine compared to electricity prices. 43
3.11 Monthly energy generation compared to annual mean wind speed. 43
3.12 Global horizontal solar irradiance (in W/m$^2$) for two different days, measured in Kailua-Kona (Hawaii). 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Heating systems with different degrees of specificity, as implemented on a model 111 m² apartment.</td>
</tr>
<tr>
<td>4.2</td>
<td>Localized heating system overview.</td>
</tr>
<tr>
<td>4.3</td>
<td>Alternative localized heating setups.</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary of the direction algorithm for localized heating.</td>
</tr>
<tr>
<td>4.5</td>
<td>Overview of the proposed ultrasonic sensor-based positioning system.</td>
</tr>
<tr>
<td>4.6</td>
<td>Overview of the positioning algorithm for the proposed ultrasonic distance sensor-based positioning system.</td>
</tr>
<tr>
<td>4.7</td>
<td>Overview of the proposed motion sensor-based tracking system. Darker area denotes more recent sensor activation (at the end of movement).</td>
</tr>
<tr>
<td>4.8</td>
<td>Overview of the positioning algorithm for the proposed motion sensor-based positioning system.</td>
</tr>
<tr>
<td>4.9</td>
<td>Occupant positioning system components.</td>
</tr>
<tr>
<td>4.10</td>
<td>Motion sensor-based positioning system implemented in a 5.4 m × 3.6 m section of a living room, using a 9×6 sensor array.</td>
</tr>
<tr>
<td>4.11</td>
<td>Paths taken in positioning system tests.</td>
</tr>
<tr>
<td>4.12</td>
<td>Metrics used in the evaluation of the positioning system.</td>
</tr>
<tr>
<td>4.13</td>
<td>Sensor and heater placement for a 111 m² apartment.</td>
</tr>
<tr>
<td>4.14</td>
<td>Savings from localized heating according to building category.</td>
</tr>
<tr>
<td>4.15</td>
<td>Savings from localized heating for different central heating settings.</td>
</tr>
<tr>
<td></td>
<td>Savings from a dual radiant heater setup is also shown.</td>
</tr>
<tr>
<td>5.1</td>
<td>Placement of temperature sensors in the apartment.</td>
</tr>
<tr>
<td>5.2</td>
<td>Temperature variations in the building, each timestep represents thirty seconds. External temperature between 0°C and 18°C.</td>
</tr>
<tr>
<td>5.3</td>
<td>User device components and functions.</td>
</tr>
<tr>
<td>5.4</td>
<td>Infrastructure support components and functions.</td>
</tr>
<tr>
<td>5.5</td>
<td>OLS system implementation in a 111 m² apartment.</td>
</tr>
</tbody>
</table>
Modern human society is increasingly dependent on energy to function. Every aspect of modern society, from building conditioning to transportation, manufacturing, and communications, requires energy either directly or indirectly. This reliance on energy has significant costs. Globally, energy expenditures represented 10% of gross domestic product (GDP) in 2010, up from 8% in 1990 [1]. Currently, energy expenditures are second only to healthcare costs in many countries.

In the United States, the total energy consumption in 2010 was 97.7 quadrillion BTU (quads), representing energy expenditures of around $1.2 trillion, or 8.3% of the country’s GDP [2]. Figure 1.1(a) illustrates the historical trend in primary energy consumption in the United States. From the figure, we observe a gradual rising trend in energy consumption. Figure 1.1(b) illustrates the historical trend in energy expenditures. From the figure, we observe a steep increase in energy expenditures between 2002 and 2008, followed by a drop in energy expenditures in 2009 (possibly due to uncertainties in the global economy). As the economy began to pick up in 2010, energy expenditures resumed their rising trend. Between 1970 and 2010, the country’s energy expenditures as a proportion of GDP ranged from 5.9% to 13.7%. Globally, energy consumption is expected to increase by more than 50%, compared
to current levels, by 2035 [3]. In 2011, approximately 84% of global energy consumption was derived from fossil fuels [3]. The burning of fossil fuels results in carbon emissions, a major contributor to global warming. In addition, increased household energy expenditures lead to a reduced standard of living, as a larger proportion of
the household income is spent on energy. To mitigate these effects, the efficient use of energy, through energy conservation measures, is critical.

Buildings represent a prime target for energy conservation. Globally, buildings represent 21% of total energy consumption [3]. In the United States, based on data collected in 2011, 40% of the country’s energy consumption is attributed to commercial and residential buildings, with transportation taking up 28% and the industrial sector, 32% [4]. Thus, reducing building energy consumption can have a significant impact on the global energy footprint and, consequently, global energy expenditures.

The relatively long life cycle of buildings is a major challenge and contributes to their lower energy efficiency. In the United States, 66% of vehicles on the road were manufactured less than 10 years ago [5]. In contrast, only 26% of existing commercial buildings were constructed less than 10 years ago [6]. Due to the long life cycle of buildings (and their corresponding slow replacement rate), technological improvements in new building design and construction, although important, would require several decades before becoming ubiquitous. Moreover, demolishing and reconstructing existing buildings represent a significant financial investment, along with considerable environmental impact. Based on recent studies, it would take between 10 to 80 years for a newly constructed, energy-efficient building to compensate for the carbon footprint of the construction process [7]. Thus, to achieve building energy conservation in an environmentally sound manner, retrofitting existing buildings is an equally important step. Building retrofits are systems installed to augment or replace existing building systems, resulting in improved energy efficiency and/or occupant comfort. Some examples of building retrofits include light-emitting diode (LED) lighting, small-scale renewable energy generation through solar panels or wind turbines, and self-programming thermostats. These building retrofits can be implemented in existing buildings, yielding immediate energy savings, cost savings, and carbon emissions reductions.
This dissertation discusses techniques in the analysis and optimization of building energy consumption, focusing on retrofits that can be applied to existing buildings. In addition to the analysis of currently available retrofits, this dissertation discusses novel retrofit concepts based on recent advancements in intelligent sensing and control. Wireless sensor systems are now a mature technology, and are ready for deployment in novel building retrofits. This dissertation also discusses retrofit-oriented simulation tools that can assist in the analysis of building energy consumption. These simulation tools are valuable in estimating the impact of building retrofits before actual implementation.

1.1 Building Energy End-uses

In order to identify target areas for improvement, an understanding of the building energy end-uses is necessary. In the United States, much of the building energy is consumed in building systems that cater to occupant comfort, namely, heating, ventilation, air conditioning (HVAC), and lighting. The building energy end-use splits for the United States are illustrated in Figure 1.2 based on information from [6] and [8]. In commercial buildings, the key energy end-uses are as follows: space heating (38%), lighting (20%), water heating (8%), space cooling (7%), and ventilation (7%). In residential buildings, the key energy end-uses are as follows: space heating (43%), water heating (17%), space cooling (10%), and lighting (6%); energy consumption in ventilation is reflected in the space heating and space cooling values.

From Figure 1.2, we observe that HVAC, alone, represents 52% of total energy consumption in commercial buildings and 53% of total energy consumption in residential buildings. Thus, improvements in the building HVAC systems can result in significant energy and cost savings.
(a) Energy end-use splits for non-mall commercial buildings in the United States.

(b) Energy end-use splits for residential buildings in the United States.

Figure 1.2: United States building energy end-use splits.

The end-use splits presented in Figure 1.2 do not take into account variations in building function and climate. In a warmer climate, such as that of California
or Texas, space cooling will represent the bulk of energy consumption rather than space heating. Further, in most modern office buildings, we expect computing to represent a significantly larger proportion of total energy consumption, in some cases, over 25% [9]. Thus, depending on the buildings in question, the results from energy conservation techniques may vary significantly.

1.2 Building Energy Optimization

Building owners looking to reduce their energy footprint and, consequently, energy expenditures can look to options in three broad categories: renewable energy generation, building fault detection and rectification, and building retrofits. Renewable energy generation involves the generation of electricity, using solar panels or wind turbines, to offset the electricity consumption in the building. Building fault detection and rectification finds and fixes faults in building systems and the building envelope, reducing energy wastage due to faulty systems or drafty buildings. Building retrofits augment or replace existing building systems to improve performance and/or energy efficiency. In this section, we discuss some of the building optimization options available to building owners.

1.2.1 Renewable Energy Generation

Solar and wind energy is typically what comes to mind when we consider renewable energy sources. At the building level, solar energy is currently being used in electricity generation (through the use of photovoltaic cells) and in water heating, and wind energy is being converted to electricity through small-scale wind turbines [10].

Electricity generation from renewable sources is an attractive option for building owners, with the promise of clean and free energy. However, the large initial investment serves as a significant barrier to entry. The cost of entry-level solar and wind
installations ranges from several thousand dollars for single-family homes to tens of thousands of dollars for large buildings. In addition, renewable energy generation is heavily dependent on the climate and, thus, might not be a viable option for all geographical regions. Nonetheless, combined with suitable energy conservation measures, renewable energy generation provides protection from rising energy expenditures.

1.2.2 Fault Detection and Rectification

Fault detection and rectification involve checking the building envelope and/or building systems for faults, then rectifying them accordingly. This reduces energy wastage in faulty systems or drafty buildings, representing a low-cost, high-impact approach to energy conservation. Finding and fixing air leaks in the building envelope (windows, walls, floors, and roofs) is an example of fault detection and rectification. Although not necessarily a part of fault detection, adding insulation to the building envelope (e.g., additional insulation in the attic) is often performed in conjunction with fixing air leaks. HVAC energy savings of up to 20% can be achieved from a combination of sealing air leaks and additional building envelope insulation [11].

Detecting and fixing faults in building systems, such as the HVAC system, requires advanced techniques. In HVAC fault detection [12, 13], a set of rules are formulated for each HVAC system component based on expected system behavior. When one or more of these rules are broken, the fault detection system indicates a possible fault in the corresponding system components. For example, if the supply (output) air temperature from the heating system is lower than expected, the fault detection system will signal a fault in the heating system. By replacing faulty HVAC components, energy savings of up to 20% can be achieved [13].
1.2.3 Building Systems and Retrofits

Buildings are constructed with a relatively long expected life cycle, up to several decades. Building systems, in general, follow the same principle and, with proper maintenance, are expected to last as long as the buildings in which they are deployed. These building systems can be augmented, using updated system components, to improve their performance and/or energy efficiency, e.g., incandescent light bulbs can be replaced with LED lamps while preserving the original light sockets and electrical wiring.

The system illustrated in Figure 1.3(a) represents the large majority of existing building systems. These are usually central systems, in which a single system, controlled using feedback from a sensor, is used to condition a large space, e.g., a typical central heating system heats up an entire building at once, based on input from a single thermostat, and is unable to condition spaces individually. In addition, these systems are usually occupant-unaware, functioning independent of the occupants’ positions and surrounding environment (e.g., the central heating system has no knowledge of where the occupants are, and the actual temperatures at the occupants’ positions).

Currently available solutions improve on these building systems by deploying additional sensors and advanced controls, as illustrated in Figure 1.3(b), while preserving the underlying central system. The additional information available to the system enables better decision making by the controller. The Nest thermostat [14] and smart thermostat [15] [16] [17] are examples of such solutions, relying on both occupancy prediction and coarse-grain occupancy sensing, via motion sensors, to automatically manage thermostat settings. These thermostats can determine when the building is vacant and reduce the HVAC output accordingly, enabling energy savings. Another solution, motion-controlled lighting [18], is commonly deployed in areas that are sporadically occupied, e.g., office bathrooms. The scope of these solutions, however, is
limited to reducing vacancy energy wastage, where building systems are left on in vacant spaces. With the Nest thermostat, for example, energy savings can only be realized when the entire building is vacant, since the central heating system does not allow finer-grain control, *e.g.*, at the room or zone level. This limited scope is a direct result of preserving the existing central system, since the all-or-nothing nature of these systems does not afford much control options.

Whereas multiple sensors offer better information on the indoor environment, distributed systems offer better control over building systems. A distributed system, illustrated in Figure 1.3(c), comprises several homogeneous or heterogeneous systems that can be individually controlled, *e.g.*, zonal heating divides the target region into zones that can be individually heated; heating in each zone is controlled by independent thermostats. Distributed systems offer better energy utilization by reducing the...
conditioned space. Under zonal heating, for example, only occupied rooms or zones are heated, resulting in energy savings even when the building is occupied.

Multi-sensor, distributed systems represent the current state-of-the-art for energy-efficient building systems. By combining better information (through the increased use of sensors) and finer-grain control (through distributed systems), these systems allow more avenues for efficient energy management than traditional central systems. In order to further optimize building energy, we believe that the next frontier in building technologies requires sensing at the level of the occupant, \textit{i.e.}, building systems must be made \textit{occupant-aware}. An occupant-aware system requires two key components: occupant positioning and occupant-level sensing. Occupant positioning provides information on the number of occupants in the building and their positions (coordinates in the building). Occupant-level sensing refers to the deployment of low-power wireless sensors, carried by the building occupants, to detect the exact environmental conditions that each occupant is experiencing. Some examples of detectable environmental conditions include temperature, humidity, and indoor air quality (through carbon dioxide concentration levels). The combination of occupant positioning and occupant-level sensing enables the building systems to know where the occupants are in the building and the actual environmental conditions experienced by each occupant.

### 1.3 Building Energy Analysis

Building owners looking to reduce their energy expenditures have several choices. To help decide among these choices, an energy analysis of the current building can be performed. Building energy analysis provides information on the building’s energy consumption, \textit{i.e.}, the energy consumed in building systems, such as HVAC and lighting (information similar to Figure 1.2, but tailored to the specific building under
analysis). This can be achieved through building energy metering and/or building energy simulation. Using this information, building owners can then select the retrofits that are best suited for their respective buildings, e.g., a building with a significant proportion of energy consumed through space heating will benefit more from improvements to the heating system, as opposed to another building with minimal energy consumption in space heating.

Building energy simulation enables the comparison of building retrofits before actual implementation. Multiple building models, each representing the target building with a different set of building retrofits, can be created and simulated. A building model captures the characteristics of a specific building, e.g., the surface dimensions, materials used in its construction, occupancy information, and building system parameters such as HVAC settings. The simulator reads the building model and performs computations to determine the expected energy consumption for the building. This process is repeated for each set of building retrofits. The results from building energy simulation help building owners decide which retrofits are most suited to their buildings.

1.3.1 Building Energy Metering

Building energy metering involves the measurement and classification of energy consumption within the building. This is an important first step in improving the energy efficiency of buildings, providing building owners with feedback for further energy conservation measures. Basic real-time energy information, at the building level, can be provided by smart meters or current sensors attached to the main power line into the building [19]. This metering information, although useful, is very coarse-grained, and does not provide information on specific end-uses. Advanced energy meters, such as ElectriSense [20], provides energy consumption information at the appliance-level, i.e., the energy consumption for each appliance or end-use is measured. These ad-
vanced metering techniques provide a better picture of energy consumption in the building.

In addition to their role in building energy analysis, energy metering can also indirectly enable energy savings. Studies have indicated that, armed with better information on their energy consumption, motivated occupants were able to achieve up to 15% in energy savings through behavior modification [21]. Clearly, energy savings from energy metering alone are occupant-dependent and not guaranteed. Nonetheless, the additional information provided by metering can help building owners in both identifying potential areas for energy conservation and evaluating already-implemented energy conservation techniques.

### 1.3.2 Building Energy Simulation

Building energy simulation is an important part of building energy analysis. It has three key phases: modeling, simulation, and results analysis. In the modeling phase, a baseline building model is first generated, based on the current state of the target building. Additional building models, featuring various combinations of retrofits, can also be set up for comparison. The effort spent in this phase varies according to the required accuracy of simulation; more effort is required to develop highly specific, complex building models that produce more accurate simulation results. In the simulation phase, the simulators take the building models as input and perform simulations over a specified time period and climate. In this phase, the performance of the building systems is evaluated. In addition, the energy consumption profile of the building systems is computed. In the final phase, results analysis, the results of the simulation are presented to the user. This can be in the form of text-based output or more advanced graphical output. Additional computations can also be performed in the results analysis phase, based on the raw data outputted from the simulation phase.
Building energy simulation provides a quick and easy estimate of the energy consumption in a given building, based on the corresponding climate and building characteristics. Using the information derived from building energy simulation, the building owner can make better decisions on which retrofits to install. Some popular building energy simulators include EnergyPlus [22, 23], DOE-2 [24], and TRNSYS [25]. Another popular tool, eQuest [26], provides a graphical user interface and several usability enhancements to the DOE-2 simulation engine.

1.4 Contributions of the Thesis

This thesis provides solutions in the analysis and optimization of building energy consumption. In terms of building energy optimization, the thesis provides two occupant-aware solutions: localized heating and occupant-level sensing. The occupant-aware nature of these solutions ensures that conditioning can be performed according to the occupants' positions and exact environmental conditions, leading to a significantly reduced indoor conditioned space. The reduced conditioned space enables considerable energy savings. To assist in the analysis of building retrofits, the thesis describes a building energy simulator targeted at the comparison of retrofits. The simulator supports novel, occupant-aware solutions, in addition to currently available retrofits. Using the simulator, building owners can evaluate the performance of retrofits and decide which retrofit combinations to implement in their buildings.

The thesis first considers renewable energy generation as a solution to building energy consumption. The generation characteristics for both solar and wind energy generation are presented, along with a discussion on the impact of climate and location on actual renewable energy generation. Renewable energy generation, at the building level, is then evaluated in terms of return on investment and actual electricity generation.
The thesis then describes localized heating, a novel occupant-aware heating solution. Under localized heating, the central heating setting is reduced (e.g., to 10°C); radiant heaters are then directed at the occupants to cover the remaining heating requirements. To enable localized heating, an occupant positioning system is required. Two possible solutions, using ultrasonic distance sensors and passive infrared motion sensors, are described and evaluated. Localized heating is evaluated under the EnergyPlus simulation framework [22, 23]. The simulation results indicate average savings of 52% in heating energy (maximum of 76%) and 46% in heating cost (maximum of 59%) when compared to central heating alone. Note that localized heating uses electricity as well as natural gas, whereas central heating uses only natural gas. Since electricity costs approximately 2.5 times as much as natural gas, there is a disparity between energy savings and cost savings from localized heating.

The thesis thereafter presents occupant-level sensing (OLS), a comprehensive system to provide information on the actual environmental conditions that building occupants are experiencing. The OLS system comprises two parts: a user-carried device, and the corresponding infrastructure support. Sensors are deployed on the user device to monitor environmental parameters such as temperature, humidity, and carbon dioxide (CO₂) concentration. The OLS system can also be used for occupant positioning with the addition of appropriate sensors. The information provided by the user devices are then used to control the corresponding building system. An improved localized heating solution, using OLS for feedback, is described. In addition, two other applications of OLS, occupant-level air quality management and occupant-level smart lighting, are discussed.

In addition to contributions in optimizing building energy consumption, the thesis describes ROBESim, a retrofit-oriented building energy simulator, based on EnergyPlus [22, 23], that can be used in the analysis and evaluation of building retrofits. ROBESim was developed as part of this dissertation. It provides definitions for
retrofit modules that can be automatically applied to existing building models. Previously, building owners, who wanted to compare different retrofits, were required to implement the retrofits manually in their building models. ROBESim significantly reduces the effort required on the part of the building owner. In addition, to support the evaluation of occupant-aware building solutions, ROBESim implements occupant profiles, *i.e.*, occupant positioning information and preferences. This information can be used by retrofit developers to evaluate their occupant-aware retrofits across a range of building models and climate zones. To simplify this process, a batch simulation and post-processing tool and a building model generator, both included with ROBESim, are described.

### 1.5 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 discusses related research efforts in the analysis and optimization of building energy consumption. Chapter 3 discusses renewable energy generation as a solution to reducing building energy expenditures. Chapter 4 describes localized heating, a novel occupant-aware heating solution that significantly reduces the indoor heating space. Chapter 5 describes an OLS system that provides information on the occupants’ exact environmental conditions. Chapter 6 describes ROBESim, a retrofit-oriented building energy simulator that can be used to provide a quick analysis across retrofit choices. Chapter 7 concludes the thesis and presents ideas for future research.
Chapter 2

Related Work

This thesis presents solutions toward energy conservation and the facilitation of building energy analysis. Both of these areas are well studied. This chapter provides an overview of the related work in these areas. In Section 2.1 building energy efficiency techniques are described and classified according to the corresponding building energy end-uses. In Section 2.2 previous work in the area of building energy analysis is discussed.

2.1 Building Energy Efficiency Approaches

In this section, various current and ongoing efforts directed toward improving building energy efficiency are discussed, focusing on technological improvements to the following building subsystems: HVAC, lighting, water heating, and computing. These subsystems represent a significant proportion of building energy consumption and present several opportunities for energy efficiency measures.
2.1.1 Heating, Ventilation, and Air Conditioning (HVAC)

Due to the large fraction of energy consumption attributed to HVAC, it has a great potential for energy conservation. An important first step in reducing HVAC energy consumption is the optimization of existing HVAC systems. Techniques to achieve HVAC optimization are focused around two key areas: optimized HVAC controls and HVAC fault detection. Optimized HVAC controls [27, 28] consider the interactions between the different HVAC components and control these components accordingly to achieve energy savings; up to 60% HVAC energy savings have been estimated [27], although actual savings depend on the existing HVAC system in the building. HVAC fault detection [12, 13] can also lead to considerable energy savings. By replacing HVAC components that have deteriorated, energy savings of up to 20% can be achieved [13]. These HVAC optimization techniques are attractive solutions, primarily due to their low implementation cost and high impact, although the actual energy savings that can be achieved depend heavily on the existing HVAC system.

In terms of HVAC energy efficiency, existing solutions target vacancy wastage (HVAC left on in a vacant room or building), primarily through programmable or smart thermostats. Although programmable thermostats are cheap, easy to install, and can lead to significant HVAC energy savings, their reliance on the occupants for programming is not an ideal situation: of the 30% of homes in the United States with programmable thermostats, only 36% of the homeowners were reported to have programmed their thermostats [8], whereas the remainder used them like regular thermostats. We observe that systems requiring extensive user participation are often ineffective, as the average user is either not sufficiently motivated or not diligent enough to use them. Thus, to reduce energy consumption, we need proactive, automated systems that lower the burden on users, while not compromising user comfort. The Nest learning thermostat [14, 29] solves this problem through self-programming, after a learning period. Occupant settings during this learning period are recorded
and used to predict future thermostat settings. The Nest thermostat also features an on-device motion sensor for automated learning.

Whereas the Nest thermostat features a single, on-device motion sensor, using multiple sensors to track occupancy could enable better prediction. This has been the focus of smart thermostat technology. The smart thermostat \cite{15, 16} utilizes a large number of motion sensors to detect occupancy in all areas of the building, enabling the system to be fully automatic, \textit{i.e.}, the system does not rely on the users to input their occupancy schedules, resulting in a reduced learning period. The PreHeat system \cite{30} uses radio-frequency identification (RFID) tags attached to the occupants’ keys to monitor their occupancy status. Both groups of solutions, although implementable on a larger scale, focus on single-family homes or apartments. In \cite{17}, a large-scale occupancy-based HVAC solution, especially suited for office buildings, is presented. This solution augments automated building management systems, available in many existing office buildings, with a combination of motion sensors and door sensors, deployed in various rooms of the building. A similar solution, presented in \cite{31}, relies instead on low-power cameras to monitor room occupancy. All these solutions function, essentially, as self-programming thermostats, \textit{i.e.}, thermostats that can be automatically programmed based on the detected and predicted occupancy status of the building.

One novel HVAC system improvement, beyond thermostats, is localized air-conditioning \cite{32}. Localized air-conditioning improves existing office air-conditioning systems by controlling the direction in which air is output from the vents. The direction of the outputted air is based on the occupancy of the space beneath the vents, \textit{i.e.}, the vents above unoccupied spaces are automatically closed. The localized air-conditioning system saves energy by only conditioning occupied spaces. Another solution, demand-controlled ventilation, can be implemented to reduce energy wastage due to over-ventilation \cite{33, 34}. This ventilation system relies on
carbon dioxide (CO$_2$) concentration information from CO$_2$ sensors placed within the building; building ventilation is switched on only when the CO$_2$ concentration exceeds the acceptable levels. Ventilation energy savings of up to 80% can be achieved under demand-controlled ventilation [33].

2.1.2 Lighting

Lighting represents 20% of total energy consumption in commercial buildings and 6% in residential buildings [6, 8]. Switching to energy-efficient lamps is the quickest path to lighting energy savings. In residential buildings, compact fluorescent and light-emitting diode lamps are four times more energy-efficient than incandescent bulbs. In commercial buildings, although much of the lighting is already provided by fluorescent lamps, switching from less efficient T12 fluorescent bulbs to more efficient T8 and T5 bulbs can yield lighting energy savings of 10% and 40%, respectively [35]. In addition, improvements to lamp reflector design and lamp controls can enable further energy savings [36]. In the United States, T12 lamps have been phased out since July 2012, although these lamps are still widely used globally.

Apart from switching to more efficient lamps, other solutions for reducing building lighting energy include motion-controlled lighting and task lighting. Motion-controlled lighting, such as the GE Aware system [18], automatically switches lights on when the space is occupied and switches them off after a period of vacancy. This reduces vacancy wastage in illuminating unoccupied spaces. Potential lighting energy savings, for most scenarios, range from 20% to 35% [35]. However, large variances in energy savings have been reported. In [37], a survey of motion-controlled lighting experiments indicates savings from 3% to 86%. This large variance is primarily due to different building characteristics and functions among the buildings surveyed. Task lighting involves the use of low-power lamps, placed on desks, to provide additional lighting for various tasks, such as reading and typing, as opposed to higher-output
ambient lighting for the entire space. Increased reliance on task lighting can lead to lighting energy savings of around 25% [35].

Daylight harvesting involves the smart design of buildings to capture as much natural sunlight as possible. Dynamic lighting tuning is an important component of daylight harvesting, representing the state-of-the-art in current building lighting technology [38, 39]. Under dynamic tuning, indoor lighting fixtures are dimmed based on the amount of natural lighting available from the sun. This results in lighting energy savings from 25% to 60% [35]. Smart lighting [40, 41, 42] combines both occupancy sensing and dynamic tuning. Under smart lighting, light and occupancy sensors are placed on workstations in an office. Using these sensors, the smart lighting system can adjust the indoor illumination based on the occupancy status of each workstation and the specified lighting requirements. Under smart lighting, energy savings of 50% to 70% have been reported [42].

2.1.3 Water Heating

After HVAC, water heating is the second largest consumer of energy in residential buildings [8]. An overwhelming majority, 97%, of homes in the United States rely on hot water tank heaters [8]. The remaining 3% rely on tankless water heating. In a tankless water heating system, hot water is not stored in a tank. Instead, water is heated at the point of use (the tap or shower), using either a gas or electric water heater installed at that location. This is generally regarded as a more efficient water heating solution, since no heat is lost in hot water storage under tankless water heating.

Currently, energy conservation techniques in water heating typically involve upgrading to more efficient heating systems or reducing hot water consumption [43, 44]. Switching to high efficiency water tanks (hot water tanks with better insulation and reduced heat loss to the surroundings) can lead to water heating energy savings of 10-
20%, whereas switching to tankless water heaters can lead to savings of 45-60% \[43\]. Heat pumps and solar water heaters, technologies particularly well-suited to warmer climates, can lead to energy savings of 65% and 70-90%, respectively \[43\]. This list represents much of the currently available technologies with regard to water heating. Understandably, the implementation cost and effort increases with the complexity of the technology. For example, a home solar water heating system costs $5,000 to $10,000, excluding government incentives, whereas an energy-efficient water tank costs less than $1,000 \[45, 46\].

Although much work has been done in evaluating solar water heating systems and their economic considerations \[45, 46\], there do exist a few novel water heating solutions. The PreHeat system \[30\], a smart thermostat system, also considers water heater control. Using the same algorithm as the one used for occupancy prediction, the system enables deeper temperature setbacks in the hot water tank when the home is vacant, and preheats the water tank when the occupants are expected to return. Nonetheless, simply switching to a tankless water heater can yield greater energy savings, with guaranteed hot water availability, at a slightly higher implementation cost.

### 2.1.4 Computing

On an average, computing represents a small portion (2%) of overall energy consumption in buildings \[6, 8\]. However, for some types of buildings, such as offices and data centers, the proportion of energy consumed in computing can be much higher, e.g., computing represents 8% of total energy consumption in educational facilities and 10% in office buildings \[6\]. Currently, power management features are built into most modern operating systems. Although effective in reducing energy consumption, these features are often neglected, and computers remain left on when not in use \[9, 47\].
Many users leave their computers on to support remote access and to run background applications, such as Internet messaging (IM), email, and file sharing [48]. The Somniloquy system [48] involves augmenting computers with networked low-power embedded systems, allowing the computers to be put into sleep mode without affecting regular use. The system supports on-demand remote access and background applications through the use of application-specific “stubs” (applications designed to run on the low-power system), resulting in estimated energy savings of 65% without affecting system availability. An alternative solution, Sleepserver, relies on a central server to automatically manage the switching of sleep/wake modes for networked enterprise workstations [49].

A more readily available solution, cloud computing, can help reduce energy consumption in unused computers. Cloud storage services, such as Dropbox [50], automatically synchronize files to a central server, allowing remote access even when the workstations are switched off. Cloud computing services, such as those provided by Amazon Web Services [51], remove the need for desktops and workstations, and enable remote access of files and applications. Under cloud computing, energy consumption is transferred to the data centers that host the cloud infrastructure, where it can be better managed.

2.1.5 Challenges and Concerns

In this section, we discuss some key challenges and concerns in the field of energy-efficient building technologies.

Security and privacy: Energy-efficient building technologies are increasingly dependent on building automation through intelligent sensing and control, often via wireless sensors and controllers. Attacks on these systems have been demonstrated, typically targeting user privacy [52]. Additional security concerns for future smart
buildings are discussed in [53]. Fortunately, mitigation techniques are available for these attacks [52], and should be implemented whenever possible.

*Building codes:* Building technologies must adhere to strict building codes and design guidelines, *e.g.*, standards and guidelines for HVAC systems [54, 55]. Although building codes are an important means toward energy conservation [56], these codes sometimes lag behind technological innovations. Nonetheless, it is important that new building technologies adhere to existing building codes and guidelines to ensure a safe and comfortable indoor environment.

*Lack of motivation:* The most important concern for energy-efficient building technologies is the lack of user motivation or awareness. This is especially true for residential homeowners. Energy costs remain a relatively small fraction of the average homeowner’s spending (around $2,000 annually [8]). Thus, motivating homeowners to pursue energy-efficient building retrofits is a difficult, but important, task. Further, as the low usage statistics for programmable thermostats indicate, expecting occupants to modify their habits is too optimistic. This motivates the need for automated solutions that require minimal user effort after implementation.

### 2.2 Building Energy Analysis Tools

Approaches toward building energy analysis fall into two broad categories: building energy metering and building energy simulation. The former category of approaches involves actively measuring energy consumption in a building. The latter involves the creation of accurate simulation models to provide an estimation of the energy consumption in a building. Often, a combination of these two approaches is used to provide a clearer picture of building energy consumption. In this section, we highlight some of the tools for building energy analysis that are currently available.
2.2.1 Metering and Energy Audits

There are currently a number of approaches to building energy analysis, varying in terms of cost and precision of information. Smart meters can be used to capture the total energy consumption in a building. These meters can be provided by the utility or self-installed [19]. However, the granularity of information provided by smart meters is often insufficient. Building owners seeking to identify the major end-uses within their buildings are unable to do so with basic smart meters. Nonetheless, smart meters represent a cost-effective means to measure and track building energy consumption before and after retrofitting.

In order to obtain more fine-grained information through metering, more advanced techniques have to be implemented. These approaches can be classified into two broad categories: single-point metering [20, 57] and multi-point metering [58, 59, 60, 61, 62, 63]. Under single-point metering, appliances are classified based on their signatures, using a sensing system deployed at a single location within the building. In [20], the ElectriSense system is plugged into any power socket in the building. The sensor then tracks the electromagnetic interference in the power lines within the building, to determine which appliances are switched on. Under multi-point metering, sensors are deployed on all the appliances in the building. The sensors detect whether the appliances are switched on or off. Both metering systems result in information on the state (on/off) of all the appliances in the building. By correlating this information with a smart meter, the energy consumption attributed to each appliance can then be determined.

For a comprehensive evaluation of building energy consumption, an in-depth building energy audit can be performed [64, 65, 66]. Building energy audits involve a combination of metering and simulation to provide more accurate information. Understandably, this option is more costly than metering or simulation alone. However, most building energy audits include checks for buildings faults, such as air leakages
and areas of poor or deteriorating insulation. Fixing these faults is often the most cost-effective means toward improving building energy performance. The downside of building energy audits is their additional cost. Building owners with limited retrofit budgets are often unable to take advantage of building energy audits.

2.2.2 Existing Building Energy Simulators

For building owners looking for a quick energy analysis, building energy simulation is an attractive option. Unlike metering, building energy simulation allows the building owners to estimate the impact of certain building retrofits before implementation. The results from multiple simulations, using different retrofit combinations, can be used to make comparisons between the different retrofit choices. Several building energy simulators are currently freely available. We highlight EnergyPlus \[22, 23\] and DOE-2 \[24\] as some of the most popular options in building energy simulation. TRNSYS \[25\], although not free software, is also commonly used for building energy simulations. eQuest \[26\] provides a graphical user interface and several enhancements that significantly improve the usability of the DOE-2 engine. These additions allow easy development of building models for simulation using the DOE-2 engine.

The information provided through building energy simulation is often sufficient for the building owner to make his retrofit choices. Unfortunately, existing building energy simulators are primarily focused on accurate computations for a single input set (a building model and weather data file) and do not have built-in functionality to perform comparisons. Further, each building model must be manually modified to consider additional retrofits. This additional effort, on the part of the building owner, represents a significant barrier of entry to building energy simulation. In addition, the steep learning curve of these building energy simulators serves as another barrier to their use.
Chapter 3

Renewable Energy Generation

Renewable energy generation is often considered by building owners seeking to reduce their energy expenditures. This chapter discusses small-scale renewable energy generation as a solution for such reductions. The energy generation properties of solar panels and small-scale wind turbines are discussed along with an evaluation of small-scale renewable energy generation. In addition, issues related to renewable energy generation are considered. Renewable energy generation primarily targets electricity expenditures. Solutions for reducing natural gas energy expenditures will be presented in later chapters.

This chapter is based on work originally published in 2010 by Chuah et al. [67], updated to include recent advancements in renewable energy technology.

3.1 Introduction

In recent years, there has been increased public awareness of the need to develop sustainable energy sources. More people around the world are becoming aware of and participating in “green” living by committing to reduce their impact on the environment. Renewable energy generation promises free and clean energy and, thus, presents an attractive option for environmentally conscious building owners or those
simply looking to reduce their energy expenditures. In addition, commercial building owners can adopt renewable energy generation as part of the green building certification process (such as LEED [68]). The marketing and goodwill generated as a result can outweigh the cost of the initial investment in renewable energy generation. Although previously limited to large-scale solar of wind power plants, renewable energy generation products that can be installed at the building-level are now available, e.g., solar photovoltaic (PV) panels and small-scale wind turbines [69] [70]. Within the last few years, these renewable energy generation products have become cheaper and more efficient.

The potential for renewable energy generation varies according to the location and corresponding climate. In general, sunny climates are more suitable for solar energy generation, whereas windy climates are more suitable for wind energy generation. Due to the limited space available on and around buildings, options for solar energy generation are restricted to rooftop or ground PV panels, and wind energy generation is restricted to rooftop wind turbines or small wind towers. Thus, the renewable energy generation potential of a building is limited by the available roof and/or ground area.

In this chapter, some key problems in the area of small-scale renewable energy generation are explored. First, the energy generation characteristics, for both solar panels and wind turbines, are described. This information can be used to determine the renewable energy potential for any given location in the world. Second, building-level renewable energy generation is evaluated to determine its feasibility, from an investment perspective. This evaluation is based on a number of different parameters, such as location and electricity prices. Finally, the issues related to renewable energy generation are discussed. Our evaluation indicates that building-level solar energy generation, in particular, is a feasible solution, and can be included as an option when considering building retrofits. However, the large initial investment in small-scale
renewable energy generation motivates the need for less expensive energy conservation measures; after the energy consumption in a building is reduced, it would be easier to satisfy the energy requirements of the building through renewable energy generation.

### 3.2 Background

Based on a 2009 survey of residential buildings, the average American household uses 943 kilowatt-hours (kWh) of electricity every month [8]. Since the average household consists of three members [8], the average monthly electricity consumption per occupant is around 314 kWh. Based on similar survey data for commercial buildings, the average monthly electricity consumption per office worker is around 624 kWh [6]. Note that this value is obtained by dividing the total site electricity consumption, for all offices in the United States, by the total number of office workers. Thus, the electricity consumption per office worker includes electricity consumed in common areas, such as corridors and bathrooms.

The average retail electricity prices in the United States from 1995 to 2012 are presented in Figure 3.1 based on data obtained from [71]. These prices do not include delivery charges typically levied by the power company, which could be significant in some areas. From the figure, we observe relatively little variation in electricity prices from 1995 to 2002 followed by a gradual increase in prices in the last decade, especially for residential building owners; electricity prices for the commercial sector have stabilized after 2008. As a result, residential homeowners currently pay around 41% more for electricity than they were paying a decade ago, and commercial building owners experienced a 28% increase in electricity prices in the same time period.

Based on the energy consumption and pricing information presented, the average American household spends $112 on electricity expenditures each month (around $37 per occupant), or $1,344 annually ($448 per occupant), in 2012. Similarly, in a
commercial office building, the average electricity expenditures per office worker is approximately $63 per month, or $756 per year. Based on the electricity prices in Figure 3.1, the average household in 2012 is incurring an additional $389 in annual electricity expenditures compared to 2002. This increased expenditure motivates the need for renewable energy generation and energy conservation measures to protect against future increases in energy prices.

Prices of renewable energy generation products, especially solar PV panels, have decreased in recent years, further enhancing the feasibility of small-scale renewable energy generation. In addition, the performance characteristics of these products have also improved considerably. In 2009, a top-of-the-line solar panel, the Sharp ND-N2ECUC, cost $650 per 1.15 m$^2$ panel, with a 12.5% energy conversion efficiency [70]. In comparison, in 2013, a similar top-of-the-line solar panel, the Sharp ND-250QCS, costs $345 per 1.63 m$^2$ panel, with a 15.3% energy conversion efficiency [70]. This represents a 76% increase in energy generation potential (from 142 Watts (W) to
250 W), along with a 47% reduction in solar panel cost. As a result, the feasibility of building-level renewable energy generation is greatly improved.

### 3.3 Energy Generation Characteristics

The renewable energy generation potential at a given location can be estimated using available data. In this section, we describe the steps involved in this estimation. We use some simple examples in this section to assist in our description. A more comprehensive evaluation, over a range of different parameters, is provided in the next section.

#### 3.3.1 Solar Energy

Solar energy is typically what comes to mind when considering renewable energy sources. At the building level, energy from the sun is currently being used in electricity generation (using solar PV panels) and in water heating [10]. In our analysis, we concentrate on electricity generation using solar energy.

Most of the solar panels currently available on the market are rated at around 10-20% energy conversion efficiency. This represents the current cost-effective solar efficiency level for manufacturers. Silicon-based solar PV cells are theoretically limited by the Shockley-Queisser limit of 30% [72]. The use of alternative materials and concentration techniques have resulted in solar cells exceeding this limit. The current limit for solar energy conversion efficiency is around 40% [73]. However, the high cost of manufacturing such panels makes them unsuitable for mass production today.

The daily electricity generation for a solar panel, at a given location, can be estimated as follows:

\[
E_s = H_s \times \eta \times A_s
\]  

(3.1)
where $E_s$ represents the estimated daily electricity generation (in kWh), $H_s$ is the average solar radiation (in kWh/m$^2$/day) at the specified location, $\eta$ is the energy conversion efficiency (%) of the solar panel and $A_s$ is the area of the solar panel (in m$^2$). The average solar radiation, $H_s$, can be obtained from available solar resource maps \cite{74,75}. A solar resource map of the United States, obtained from \cite{74}, is illustrated in Figure 3.2. Note that the solar radiation at a given location varies according to the month of the year. The solar radiation values presented in these maps are averaged across the year. In addition, weather information (e.g., cloud cover) is factored into these solar radiation values.

![Photovoltaic Solar Resource of the United States](image)

Figure 3.2: Solar map of the United States. Darker shading reflects higher solar radiation.

Monthly solar radiation data, for selected cities worldwide, can be obtained from \cite{76}. Figure 3.3 illustrates the monthly variation in solar radiation for the
following locations: Phoenix (Arizona), San Francisco (California), Chicago (Illinois), Atlanta (Georgia), Houston (Texas), and Newark (New Jersey). From the figure, we observe that solar radiation is considerably higher in the summer months than during the winter, e.g., solar radiation in Chicago in July is more than double that in December. In addition, solar radiation varies significantly between different locations. Thus, some locations, such as Phoenix, are more suited to solar energy generation than others.

![Solar Radiation vs Month of Year](image)

**Figure 3.3**: Solar radiation by month of year for selected locations.

As an example, we analyze the solar energy potential of Newark (New Jersey), using the Sharp ND-250QCS solar panel. Recall that the Sharp ND-250QCS has an area of 1.63 m\(^2\) and an energy conversion efficiency of 15.3% \[70]. From \[76], Newark has an average solar radiation of 4.42 kWh/m\(^2\)/day. Thus, a Sharp ND-250QCS, on average, would generate 1.1 kWh of electricity daily, 402 kWh annually. From the energy consumption statistics discussed in Section 3.2, approximately ten solar panels are required to completely satisfy the annual electricity consumption of
each household member in a residential building. Note that this electricity is not generated at a constant rate; solar energy generation varies according to the actual weather and time of day. In addition, energy storage systems would be needed if the generated electricity is not immediately consumed. These issues will be considered in Section 3.5.

### 3.3.2 Wind Energy

The wind is another popular source of renewable energy. Wind energy has been consistently used in the past to drive sailboats for exploration and windmills for farming. Naturally, it is now being utilized in clean, renewable energy generation. In recent years, compact wind generation turbines have been made available for installation at the building level [69, 77, 78]. In general, these small-scale wind turbines have a smaller footprint than the wind towers used in wind farms. The small size of some of these turbines allows their placement on rooftops. One such turbine is the Honeywell Earthtronics WT6500 [69]. We base our analysis of building-level wind energy generation on this wind turbine.

The mean power generation of a single wind turbine, over the course of the year, can be estimated as follows (adapted from [79]):

\[
P_w = k_w \times V^3 \times A_w
\]  \hspace{1cm} (3.2)

where \( P_w \) denotes the mean power generation (in W), \( k_w \) is a constant factor (in kg/m\(^3\)) encompassing the density of air and a number of efficiency factors related to the characteristics of the wind turbine [79], \( V \) is the annual mean wind speed (in m/s) at the specified location, and \( A_w \) is the area swept by the wind turbine (in m\(^2\)). We note that the use of averaged \( V^3 \) values, from wind speeds recorded throughout the year, would provide a better estimation of the power generation.
of the wind turbine. However, the annual mean wind speed statistics are more readily available, and the equation presented is an acceptable method for the quick estimation of mean power generation [79]. The value of $k_w$ is typically around 0.365 kg/m$^3$ [79] and can be determined from the power curve of the wind turbine (usually provided by the manufacturer). This simplified formula allows for quick estimation of the potential electricity generation of a given wind turbine. A more specific formulation for electricity generation can be found in [80]. The annual energy generation of the wind turbine, $E_w$, in kWh, can be determined by multiplying $P_w$ with a conversion factor of 8.766 (to convert from W to kWh/year).

The most important contributing factor in wind energy generation is the annual mean wind speed at the specified location, $V$. This is due to the cubic relationship between electricity production and wind speed. Thus, wind energy generation varies considerably with the location. $V$ can be obtained from wind resource maps [74, 75, 81, 82, 83]. An example of a wind resource map of the United States, obtained from [74], is illustrated in Figure 3.4. Using this data, combined with the wind turbine specifications (to determine the value of $k_w$ and $A_w$), the expected wind energy generation can be determined. As a general guideline, locations with annual mean wind speeds above 5.6 m/s are considered suitable for wind energy generation [82].

To simplify our evaluations in Section 3.4.2, we assume a fixed wind speed (taken as the annual mean wind speed based on the wind power classification for the location) and evaluate the performance of the wind turbine based on this wind speed. For example, at a location with an annual mean wind speed of 6.0 m/s, using a Honeywell Earthtronics WT6500 wind turbine (with $k_w$ of 0.365 kg/m$^3$, $A_w$ of 2.63 m$^2$), the estimated annual electricity generation is 1,818 kWh. Thus, approximately two wind turbines are required to fully satisfy the annual electricity consumption of each household member. Similar to solar energy generation, this electricity is not generated at a constant rate. We consider these issues later in Section 3.5.
3.4 Evaluation

In the previous section, we described the energy generation characteristics of solar panels and wind turbines. In this section, we evaluate the feasibility of the chosen solar panel (Sharp ND-250QCS) and wind turbine (Honeywell Earthtronics WT6500) under different scenarios. In our evaluation, we use the return on investment (ROI) as an indicator of feasibility. The ROI represents the number of years required to recoup the initial investment in each of these renewable energy generation systems, and is computed by dividing the cost of the initial investment in the system by the annual cost savings (reduction in energy expenditures) enabled by the system. Since installation costs and government subsidies vary significantly depending on location and size of installation, we base our analysis strictly on the unit cost of each solar
panel and wind turbine, i.e., our analysis does not take into account installation costs and government subsidies. In addition, we note that our analysis does not cover every possible scenario; building owners looking to install renewable energy solutions should perform the same computation steps to determine the ROI for their particular situation. Our ROI analysis does not account for potential investment returns that can be achieved from investing the initial capital in avenues other than renewable energy generation systems. This is because the actual investment returns vary significantly between individuals and investment schemes. For a more accurate picture of ROI, the circumstances specific to the building owner need to be factored in individually.

3.4.1 Solar Energy

In our analysis of building-level solar energy generation, we use the baseline parameters listed in Table 3.1 based on the Sharp ND-250QCS solar panel [70]. The electricity price used in our baseline is derived from the United States average electricity retail price for residential consumers in 2012 [71]; location-specific electricity prices will be considered later in our evaluation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electricity price</td>
<td>$0.1188/kWh</td>
</tr>
<tr>
<td>Solar panel area, $A_s$</td>
<td>1.63 m$^2$</td>
</tr>
<tr>
<td>Solar panel efficiency, $\eta$</td>
<td>15.3%</td>
</tr>
<tr>
<td>Solar panel unit cost</td>
<td>$345</td>
</tr>
</tbody>
</table>

We first analyze how ROI is affected by the solar radiation at the location of implementation. Figure 3.5 illustrates the results of this analysis. From the figure, we observe that the ROI for the selected solar panel ranges from just under five years to around eight years. Thus, the Sharp ND-250QCS solar panel is expected to pay for itself within that time period. The ROI figures are fairly attractive; even at a
low solar radiation of 4.0 kWh/m²/day, the initial investment in the solar panel can be recouped in eight years. This is encouraging for building-level solar installations in the future, since prices for solar PV panels are expected to decrease, and energy conversion efficiencies are expected to increase.

In our next analysis, we seek to determine the sensitivity of the expected ROI to electricity prices. Figure 3.6 illustrates the results of this analysis. In this analysis, we assume a fixed solar radiation of 4.42 kWh/m²/day, corresponding to the solar radiation levels in Newark. From the figure, we observe that, as expected, ROI decreases significantly with increasing electricity prices. Thus, solar energy generation is more attractive in areas with high prices.

In our third analysis, we analyze the ROI for the solar panel when implemented at selected locations (see Figure 3.3). In this analysis, we used specific electricity prices for the selected locations, obtained from [71]. Figure 3.7 illustrates the results of our analysis.
Figure 3.6: ROI for solar panel compared to electricity prices.

Figure 3.7: ROI for solar panel compared to electricity prices at selected locations.
analysis. In this figure, the ROI and the location-specific electricity prices are shown. The solar radiation levels for the selected locations are illustrated in Figure 3.3.

From Figure 3.7, we observe that the ROI across the selected locations varies from 4.7 years (in San Francisco) to 7.2 years (in Chicago). In addition, we observe the effect that high electricity prices can have on the feasibility of building-level solar energy generation: although Newark has significantly lower solar radiation levels (4.42 kWh/m²/day, the lowest among the selected cities) compared to Phoenix (6.57 kWh/m²/day), the high electricity prices ($0.1623/kWh in Newark, compared to $0.1108/kWh in Phoenix) results in similar ROI values for solar panels at both locations (slightly more than five years). Currently, the solar radiation level is the primary consideration when assessing the feasibility of solar energy generation at a given location. Our analysis indicates that electricity prices should also be factored into these assessments.

In our final analysis on solar panels, we seek to determine the area of solar panels required to satisfy a given proportion of electricity consumption in residential buildings at a given location. In this analysis, we use the statistics provided for the typical American household (monthly electricity consumption of 943 kWh). Figure 3.8 illustrates the results of this analysis.

From the figure, we observe that, at a location with a solar radiation of 5 kWh/m²/day, a solar installation of 40.5 m² is required to completely satisfy the household’s electricity requirements. This is important in a zero net energy building, where the building must generate sufficient energy to satisfy the occupants’ energy requirements. Although this requirement (a roof area of 40.5 m²) is achievable in general, it would be more reasonable if the household electricity requirements are first reduced, e.g., through energy conservation measures. To satisfy 75% of the same household’s electricity requirements, a solar installation of 30.4 m² is required, whereas a solar installation of 20.3 m² is needed to cover 50% of the
Figure 3.8: Solar installation area required to cover selected proportions of monthly electricity consumption.

electricity requirements. In terms of cost, an initial investment of $8,625 (for 25 Sharp ND-250QCS solar panels) is required to cover all of the household’s electricity requirements, whereas $4,485 (for 13 solar panels) is required to cover 50% of the electricity requirements. Thus, if suitable energy conservation measures are implemented along with renewable energy generation, with zero net energy as the goal, the initial investment in renewable energy generation can be significantly reduced; this reduced investment can be used to fund the energy conservation measures.

In summary, solar energy generation at the building level is a promising approach for reducing energy expenditures. In general, the initial investment in solar panels can be recouped in under nine years. Another insight from our analysis is that electricity prices should also be considered, along with solar radiation, when evaluating renewable energy generation technologies. In addition, the combined effects of reduced manufacturing and equipment costs, along with improved energy conversion
efficiency, can help improve the feasibility of small-scale solar energy generation in the future.

### 3.4.2 Wind Energy

We refer to Table 3.2 for the baseline parameters in our evaluation of wind energy generation. The baseline wind turbine reflects the characteristics of the Honeywell Earthtronics WT6500: a constant factor, $k_w$, of 0.365 kg/m$^3$, a turbine area, $A_w$, of 2.63 m$^2$, and an expected unit cost of $6,000.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electricity price</td>
<td>$0.1188/kWh</td>
</tr>
<tr>
<td>Constant factor, $k_w$</td>
<td>0.365 kg/m$^3$</td>
</tr>
<tr>
<td>Turbine area, $A_w$</td>
<td>2.63 m$^2$</td>
</tr>
<tr>
<td>Turbine unit cost</td>
<td>$6,000</td>
</tr>
</tbody>
</table>

From the wind energy generation equation in Section 3.3.2, we see that the annual mean wind speed is the largest contributing factor towards wind energy generation. In our first analysis, we determine the impact of annual mean wind speed on ROI. Figure 3.9 illustrates the results of this analysis, using a fixed electricity price of $0.1188/kWh (the average residential retail electricity price in the United States in 2012 [71]).

From Figure 3.9, we observe that the ROI is significantly higher for areas with an annual mean wind speed below 5.6 m/s, *e.g.*, with a mean wind speed of 5 m/s, the ROI is 48 years, compared to 28 years at 6 m/s. Some examples of locations with annual mean wind speeds above 6 m/s are Cold Bay (Alaska), Dodge City (Kansas), Mount Washington (New Hampshire), and Amarillo (Texas). The annual mean wind speeds at the selected locations, in our solar energy analysis, are all below 5 m/s. Thus, wind energy generation is not feasible for these selected locations. In addition,
Figure 3.9: ROI for wind turbine compared to annual mean wind speed.

we observe that the ROI of the wind turbine decreases significantly in locations with high annual mean wind speeds; above 8 m/s, the ROI is less than 12 years.

To compare the sensitivity of ROI to electricity prices, we performed an analysis similar to that for solar panels. Figure 3.10 illustrates the results of this analysis, using a fixed annual mean wind speed of 6 m/s. From the figure, we observe that, even if electricity prices were to double (to around $0.25/kWh), the ROI remains high (13.2 years).

In our final analysis, we determine the potential for wind energy generation to completely satisfy the household electricity requirements. Figure 3.11 illustrates the results of this analysis. From the figure, we observe that the monthly energy generation increases significantly with the annual mean wind speed. At a location with an annual mean wind speed above 7 m/s, less than four wind turbines are needed to fully satisfy the household’s electricity needs (recall that the average American household consumes 943 kWh of electricity each month). The significantly reduced footprint
of wind turbines, in terms of required installation area, is an important advantage.
of small-scale wind energy generation; in contrast, a solar panel installation requires more than 40 m$^2$ of roof area to fully satisfy the household’s electricity requirements.

In summary, although small-scale wind energy generation is at a disadvantage in terms of investment returns, the small installation footprint of wind turbines makes them attractive for zero net energy buildings. In addition, for windy areas without much sunlight, wind energy generation is a suitable alternative to solar energy generation. As manufacturing costs decrease in the near future, we expect small-scale wind energy generation to become a better investment. However, the noise levels associated with small-scale wind turbines make them unsuitable for deployment in small homes and buildings. Currently, solar energy generation remains the better option from an investment perspective, and does not lead to much noise during operation.

3.5 Issues in Renewable Energy Generation

Both solar and wind energy generation are affected by fluctuations in power output due to changes in weather conditions. Previously, we have discussed how solar radiation levels vary throughout the course of the year. Apart from these seasonal variations, solar radiation levels are also affected by daily weather variations. Figure 3.12 illustrates the global horizontal solar irradiance (in W/m$^2$) at a weather station in Kailua-Kona (Hawaii) for two different days. The global horizontal irradiance is a measure of the solar intensity incident on a horizontal surface; solar radiation levels can be derived from the global horizontal irradiance.

From Figure 3.12(a), we observe relatively stable irradiance levels throughout the course of the day, with only minor fluctuations towards the end of the day. Figure 3.12(b) illustrates the irradiance levels just two days prior. From the figure, we observe significant fluctuations in solar irradiance levels; at certain times, the actual irradiance levels were only 30% of the expected levels. Such fluctuations are also seen
(a) Relatively stable solar irradiance throughout the day.

(b) Significant fluctuations in solar irradiance throughout the day.

Figure 3.12: Global horizontal solar irradiance (in W/m²) for two different days, measured in Kailua-Kona (Hawaii).
in wind energy generation systems. In [84], the effects of wind speed fluctuations on energy generation is studied, with the conclusion that improved forecasting tools can be a valuable asset to renewable energy generation.

Apart from these unexpected fluctuations, there is another issue for solar energy generation: the energy generation drops to zero at night. This has a significant impact on the utility of solar energy generation in residential buildings, since more energy is used at night (for lighting, entertainment systems and other purposes) than during the day. The energy generated in the day would be wasted if systems are not in place to utilize or store this energy. Several approaches can be taken to mitigate the effects of under-generation, when the generated energy is not enough to satisfy demand, and over-generation, when the generated energy exceeds demands and might possibly be wasted. In large-scale solar and wind energy generation systems, non-renewable energy generation facilities (such as coal and natural gas power plants) are used to provide the extra electricity required during times of under-generation. This can also be used in building-level renewable energy generation, where the building would simply tap into the existing power grid in times of under-generation.

Solutions to address over-generation are much more varied. To store the extra energy generated during times of over-generation, several approaches have been proposed, some already implemented in existing renewable energy generation systems. A summary of these approaches is provided in [85]. The most obvious solution would be to store the excess electricity generated in batteries that can be used later when required. However, the high cost of such energy storage systems could provide a barrier to their adoption.

Recently, an approach that has gained traction is vehicles-to-grid (V2G) energy storage [85] [86]. V2G leverages the expected future increase in the number of electric vehicles (EVs). All EVs are equipped with a suitable high-capacity battery [87]. For example, the 2013 Chevrolet Volt is powered by a 16 kWh lithium-ion battery pack.
In a V2G system, these EVs will be used to store the excess energy generated during over-generation and can also be called upon to provide extra energy to the building or power grid during times of under-generation. V2G makes use of battery storage to smooth out fluctuations in energy generation without incurring the high costs of dedicated battery storage systems. In a commercial building, the excess energy generation can be used to charge EVs for a fee. This allows the building owner to sell the electricity generated from his renewable energy sources, without the added expense of battery storage.

A more elegant solution, on a macroscopic scale, would be to allow buildings to transfer the excess energy generated into the power grid in exchange for energy credits or cash. This has been a crucial element in most smart grid concepts \[88\]. In this system, the building owner can directly “sell” the excess energy generated to the utility company, through the smart grid. A more comprehensive smart grid would also make use of V2G systems, on a local scale, to smooth out the fluctuations in energy generation. With these systems in place, it would be easier to convince building owners of the feasibility of renewable energy generation at the building level.

### 3.6 Chapter Summary

In this chapter, building-level renewable energy generation, as a solution to reducing building energy expenditures, is discussed. An overview of the computation of the renewable energy potential for a given location is provided. The steps provided in this chapter can be used by building owners in the evaluation of renewable energy generation for their individual scenarios. Our analysis indicates that building-level renewable energy generation, particularly solar energy generation, is a feasible option for most building owners, depending on the location of the building and the corresponding electricity prices. However, the reduction of overall building energy
consumption, through energy conservation measures, is equally important. After building energy consumption is reduced to a lower level, renewable energy generation can more easily satisfy the electricity needs of the building occupants.
Chapter 4

Localized Heating

Much of the energy consumed in buildings goes toward systems for occupant comfort, such as heating, ventilation, and air conditioning (HVAC). These systems currently function without information on the occupants’ locations and surrounding environment, leading to considerable energy wastage, e.g., in the conditioning of vacant spaces. In this chapter, a localized heating system, utilizing radiant heaters to provide directed heat to occupants, is proposed. Under localized heating, the temperature setting in the central heating system can be lowered considerably, resulting in significant heating energy savings.

To appropriately direct radiant heating, localized heating requires a suitable occupant positioning system. In this chapter, we describe a minimally-intrusive indoor occupant positioning system that can be used to support localized heating. Localized heating is analyzed using a representative set of building models in the EnergyPlus simulation framework. The results of this analysis indicates that the proposed localized heating system can provide average savings of 52% in heating energy (maximum of 76%) and 46% in heating cost (maximum of 59%) when compared to central heating alone. These results highlight the potential of localized heating, and other similar
techniques that utilize fine-grain occupant information, in enhancing building energy efficiency.

Initial work on which this chapter is based was published in \[89\].

### 4.1 Introduction

In the United States, building HVAC represents 52% of total energy consumption in commercial buildings and 53% of total energy consumption in residential buildings [6, 8]. Space heating, alone, represents 38% of the energy consumed in commercial buildings and 43% of residential energy consumption, the largest proportion among all building energy end-uses. Currently, the large majority of deployed building HVAC systems are central systems relying on feedback from a single sensor (see Section 1.2.3). These central systems suffer from limited sensor information (a single thermostat provides information for a large space) and reduced granularity of control (the entire space is conditioned at the same time). Energy is wasted in the conditioning of unoccupied spaces and the over-conditioning of occupied spaces. In the first case, empty rooms or spaces remain heated or cooled, leading to energy wastage. In the second case, the space is conditioned at a higher level than is required, primarily due to the limited sensor information available. Existing solutions, such as the Nest thermostat [14] and smart thermostat [15, 16, 17] target these sources of energy wastage by providing better sensor information, e.g., using occupancy sensors, the system is shut off when the space is determined to be unoccupied. In addition to these sources of energy wastage, the reduced granularity of control leads to energy wastage in the conditioning of large spaces when only a small area within is occupied, e.g., when a large room is occupied by one occupant, it would be more efficient to directly heat the occupant than the entire room. This source of energy wastage is addressed by zonal heating, where the target space is divided into zones for condi-
tioning. However, due to the large granularity of these zones (entire floors or rooms in the building), the problem remains.

In this chapter, a novel heating system, called localized heating, is described. Localized heating directly addresses energy wastage caused by reduced granularity of control in building heating systems, and fulfills the requirements of a multi-sensor, occupant-aware, distributed system, discussed in Section 1.2.3. Localized heating affords a degree of localization or specificity that goes well beyond central and zonal heating systems. Using infrared radiant heaters with controllable motors, heat is directed toward the occupants in the building, further reducing the energy spent on heating unoccupied spaces (even within occupied zones). Figure 4.1 provides a comparison between central, zonal, and localized heating, when implemented in a small residential apartment. From Figure 4.1 we observe that the heated space, under localized heating, is significantly smaller than under both central and zonal heating. Localized heating augments existing central or zonal heating systems by enabling them to be set to lower temperatures; the existing heating systems are used to maintain a minimum indoor temperature (e.g., 10°C).

Localized heating goes beyond smart thermostats [15, 16, 17]; although such approaches have shown good promise, localized heating complements them by addressing wasted energy within the occupied regions. We describe an implementation of the proposed localized heating system and evaluate it on a large set of representative building models using the EnergyPlus simulation framework [22]. Our results indicate that localized heating can reduce heating energy by 35% to 76% (average of 52%) and heating cost by 22% to 59% (average of 46%). We also perform a return on investment (ROI) analysis for the proposed localized heating system and compare it with other solutions, such as programmable thermostats. Our results suggest that localized heating is a promising approach to improve building energy efficiency.
Figure 4.1: Heating systems with different degrees of specificity, as implemented on a model 111 m² apartment.

To enable localized heating, an indoor positioning system that determines the positions of building occupants is required. We propose a positioning system that utilizes either ultrasonic distance sensors or motion sensors placed in a ceiling-mounted array throughout the building. The proposed system is minimally intrusive (does not require users to carry tags) and does not identify users, alleviating privacy concerns associated with other occupant detection technologies such as cameras. We evaluate the accuracy of both versions of the occupant positioning system and provide comparisons between the two versions.

Our proposed systems are modular, i.e., localized heating can be implemented alongside other occupant positioning and sensing systems, and our proposed positioning system can be used with other building applications.
4.2 Background

Our proposal of localized heating requires information about occupant locations within the building. This is a well-studied area, with many solutions that provide a range of tradeoffs among accuracy, reliability, and cost. Many commercial occupant location systems rely on cameras and image/video processing. In addition to the significant cost and computational requirements, cameras are often not desirable due to privacy reasons. Low resolution infrared cameras that cannot identify occupants individually are currently available; these systems can be used for localized heating. The choice of positioning system depends on the preferences of the building occupants. A good overview of research on indoor localization systems is provided in [90, 91]. Among several noteworthy efforts, we would like to mention Cricket [92] and Smart Floor [93] as two representative approaches. The Cricket system consists of beacons at fixed locations and tags that are carried by users; the difference between RF and ultrasound signal propagation is used by tags to calculate their locations. Our positioning system does not require tags, and in that sense is closer to Smart Floor, which uses pressure sensors embedded in the flooring to detect occupant locations. In contrast to floor-based sensors, we utilize either ultrasonic distance sensors or motion sensors, mounted on the ceiling, to detect objects moving into their range. We believe that this makes it easier to retrofit existing buildings (compared to floor-based systems).

4.3 Localized Heating System

In this section, we describe the proposed localized heating system that reduces heating energy consumption by directing heat towards the building occupants, based on inputs from an occupant positioning system. The key insight behind our system is that
directing heat towards the small area around the occupant is a more efficient way of providing heat.

4.3.1 System Outline

Figure 4.2 provides an overview of how localized heating works. First, the locations of all the occupants in the building are obtained by a positioning system, such as the one described later in Section 4.4. Next, the radiant heaters, mounted at fixed positions on the ceiling, are directed towards the occupants, heating the occupants and the immediate space around them. Motors attached to the radiant heaters enable them to be directed in this manner.

![Localized heating concept:](image)

1. Tracking system provides coordinates for infrared heater controller
2. Motors mounted on the base of heaters enable 2-axis rotation
3. Infrared heater is directed towards the position of the occupant

Figure 4.2: Localized heating system overview.

The proposed localized heating system does not replace the central heating system. Rather, the use of localized heating enables the temperature setting of the central heating system to be lowered significantly (e.g., to around 10°C), while maintaining occupant comfort (see Section 4.3.3). Therefore, the net energy savings from this approach can be computed as the difference between the energy saved in the central heating system and the energy consumed by the location sensors and radiant heaters.
Note that localized heating can be used with a central heating system that is divided into zones, and combined with coarse-grain duty cycling techniques that have been previously proposed. Localized heating provides more fine-grain control over heating within occupied regions of a building, and hence offers avenues for further energy conservation. It can even benefit from zonal heating systems, since the decision of whether to use localized heating or not can be performed within each zone separately (for densely occupied zones, localized heating may be less efficient due to the large number of radiant heaters that must be turned on).

Finally, since occupants may move within the building, our localized heating system is designed to continuously track the occupants throughout the building and follow them with radiant heaters, similar to a spotlight tracking a performer on stage.

In Figure 4.3, we present some alternative localized heating setups. A dual radiant heater setup is presented in Figure 4.3(a). In this setup, radiant heating is directed at the occupant from two opposing directions; the power output of each radiant heater is adjusted accordingly. This setup provides uniform heating for the occupant, as opposed to heating from a single direction. The impact of this uniform heating, along with other thermal comfort considerations, is discussed in Section 4.3.3. In Figure 4.3(b), we show how localized heating can be deployed alongside a radiant floor heating solution. Radiant floor heating ensures that the floor of the building will be at a comfortable temperature; this is important for bare-foot occupants.

In the following sections, we discuss the control algorithms associated with our localized heating system and the positioning system that we developed to support localized heating. In addition, we discuss the thermal comfort considerations when using localized heating.
4.3.2 Control Algorithm

The localized heating system first determines if there are enough radiant heaters to heat the occupants in each room of the building. If there are more occupants than radiant heaters, e.g., during a house party, the system falls back upon the original central or zonal heating. Note that the occupants can preset the heating system to central or zonal heating in anticipation of such events.

The system assigns heaters to occupants by minimizing the distance between them, such that the nearest heater would be assigned to the occupant, as far as possible. The control algorithm for localized heating comprises two parts: determining the direction to point the assigned radiant heaters towards, and determining the output levels of these heaters. Since radiant heaters are located at known, fixed positions on the ceiling, computing the direction to point the heaters is possible. Given a radiant heater at coordinates $\left(x_h, y_h, z_h\right)$ and an occupant at coordinates $\left(x_o, y_o, z_o\right)$,
the direction computation is as follows:

\[(x_p, y_p, z_p) = (x_o - x_h, y_o - y_h, z_o - z_h)\]

\[d = \sqrt{x_p^2 + y_p^2 + z_p^2}\]

\[\phi = \text{atan2}(y_p, x_p)\]

\[\theta = \arcsin\left|\frac{z_p}{d}\right|\]

Note: \(-\pi < \phi \leq \pi\) and \(0 \leq \theta \leq \frac{\pi}{2}\)

where \(\phi\) denotes the heater angle in the horizontal plane (the ceiling), and \(\theta\) denotes the heater angle in the vertical plane, as illustrated in Figure 4.4. The atan2 function is defined as follows:

\[
\text{atan2}(y, x) = \begin{cases} 
\arctan\left(\frac{y}{x}\right) & x > 0 \\
\arctan\left(\frac{y}{x}\right) + \pi & y \geq 0, x < 0 \\
\arctan\left(\frac{y}{x}\right) - \pi & y < 0, x < 0 \\
\frac{\pi}{2} & y > 0, x = 0 \\
-\frac{\pi}{2} & y < 0, x = 0 \\
0 & y = 0, x = 0
\end{cases}
\]

The atan2 function returns values between \(-\pi\) and \(\pi\). Based on the computed angles, \(\phi\) and \(\theta\), the control algorithm directs the corresponding radiant heater towards the occupant. Some occupant positioning systems, such as most camera-based systems and the ultrasonic distance sensor-based system detailed in Section 4.4.1, are able to detect if the occupant is standing or sitting. In this case, the localized heating system will account for the posture of the occupant. When this information is not available, the system will assume that the occupant is standing. This ensures adequate radiant heater coverage for the occupants.
The required radiant heater output depends primarily on the temperature differential, $\Delta T$, in °C, between the occupant’s environment and his desired (comfort) temperature. The existing central and zonal heating systems are used to maintain a minimum temperature (e.g., 10°C) in the building. This is to ensure that the indoor temperature does not drop to a level lower than what the radiant heaters can cover. The output, $P$, in Watts (W), for the radiant heater is computed as follows:

$$ P = f_T \times \Delta T $$

where $f_T$ is the radiant heater factor, in W/°C, computed based on the specifications of the radiant heater. The temperature differential is based on sensor readings from temperature sensors deployed in the different building zones, and is computed by subtracting the air temperature in the zone that the occupant is in from the occupant’s desired (comfort) temperature. In our analysis, we assume a desired (comfort) heating temperature of 22°C.

From the manufacturer’s specifications in [94], we determine the value of $f_T$ to be 13.6 W/°C, for a 1 m² heating area. We use these values in our savings analysis for localized heating in Section 4.6. Note that the $f_T$ factor can be adjusted by the occu-
pant according to his requirements, and varies depending on the specifications of the deployed radiant heater. In addition, the power output can be set to vary according to the distance of the occupant from the radiant heater, *e.g.*, if radiant heaters are limited in number and far apart. Although we have shown a linear output function with respect to the temperature differential, the control function for radiant heater output need not be linear, *i.e.*, the function can be adjusted to suit the occupants’ requirements.

Using the computed directions and radiant heater outputs, the localized heating system can direct the respective radiant heaters toward the occupants and provide the required level of radiant heat.

### 4.3.3 Thermal Comfort

Occupant thermal comfort is of utmost importance in any heating system. The overall thermal comfort under localized heating depends on two key requirements: quick radiant heater start-up and sustained thermal comfort under radiant heating. The first requirement can be satisfied by using heating elements that achieve 75% of rated radiant energy in approximately five seconds [94]. This allows the quick switching of radiant heaters as the occupant moves around the building, without any gaps in coverage. This quick switching is another advantage of localized heating over zonal heating, apart from the greatly reduced heated space. Under zonal heating, it might take up to an hour or more for a room to be heated to the desired (comfort) temperature [95], depending on the setback temperature. This results in a period of discomfort while the occupant waits for the room to be heated. Although occupancy prediction used in smart thermostats [15][16] alleviates this problem, the occasional mispredictions will still lead to occupant discomfort. In addition, these mispredictions lead to energy wastage when the building is conditioned unnecessarily. Localized heating is a more reliable solution in this regard.
Much work has been done in the evaluation of thermal comfort under radiant heating, with unanimously positive results [96, 97, 98]. These results were based on survey data collected from subjects exposed to various thermal conditions. Subjects indicated their level of thermal comfort using the thermal sensation scale shown in Table 4.1. This scale was developed under the ANSI/ASHRAE Standard 55-2004 [99], a widely followed standard in the design and implementation of heating systems. The acceptable range for thermal comfort, specified under the standard, is -0.5 to +0.5.

Table 4.1: The ASHRAE thermal sensation scale

<table>
<thead>
<tr>
<th>Value</th>
<th>Thermal sensation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+3</td>
<td>Hot</td>
</tr>
<tr>
<td>+2</td>
<td>Warm</td>
</tr>
<tr>
<td>+1</td>
<td>Slightly warm</td>
</tr>
<tr>
<td>0</td>
<td>Neutral</td>
</tr>
<tr>
<td>-1</td>
<td>Slightly cool</td>
</tr>
<tr>
<td>-2</td>
<td>Cool</td>
</tr>
<tr>
<td>-3</td>
<td>Cold</td>
</tr>
</tbody>
</table>

From [96], when heated by one radiant heater (at an outdoor temperature of 0°C), subjects, on an average, indicated a comfort level of “-0.20” on the thermal sensation scale after two hours compared to a comfort level of “+0.25” for convective heating based on central heating. This fluctuation is well within the acceptable range for thermal comfort. Note that negative comfort level values do not indicate discomfort; comfort levels indicate the deviation of thermal comfort from the ideal value, “0”. We cannot determine if a “-0.5” or a “+0.5” thermal sensation is more desirable, since some users might prefer a warmer environment and some a cooler one. In [97], Imanari et al. reported an average comfort level of “-0.40” under radiant heating, again within the acceptable range. In [98], only 5.4% of participants indicated thermal comfort outside the acceptable range. Note that all of the survey results presented were based on experiments with preset output levels for radiant heaters and no user feedback. Under localized heating, these output levels can be adjusted according to the user’s preferences and/or settings.
Another result from [96] is that, when heated by two radiant heaters from different directions, occupants experienced better thermal comfort in the short run. Upon entry into the test space, subjects indicated a comfort level of “-1.25” when heated by one radiant heater, “0” when heated by two radiant heaters, and “-0.75” under convective heating. Subjects entering a heated space from a much colder environment generally welcome a short period of increased heating intensity [100]. After the entry period, both the single and dual radiant heater setups provide acceptable levels of thermal comfort (“-0.25” for one radiant heater and “+0.25” for two). This result can be used to the advantage of localized heating. In order to provide enhanced thermal comfort, our system can be programmed to direct the heat from two radiant heaters (see the dual heater setup in Figure 4.3), and/or at slightly elevated levels, at an occupant entering the building. After a period of time, the heaters can then be adjusted back to their steady-state levels. This issue is not addressed automatically by central heating.

In addition to the flexibility in radiant heater setups and output levels, localized heating can also accommodate different central heating settings. In our discussion, we considered a minimum central heating temperature setting of 10°C. This is the lowest recommended indoor ambient temperature (to prevent the freezing of water pipes). This central heating temperature setting (of 10°C) also represents the highest possible level of energy savings that can be derived from localized heating. To accommodate occupants who are initially wary of low ambient temperatures, a higher central heating setting can be used. The central heating setting can then be reduced as the occupants get accustomed to localized heating. Our simulation results (discussed in Section 4.6) indicate that, even at a central heating setting of 20°C (with radiant heating covering the remaining 2°C), considerable heating energy savings, around 18%, can still be achieved. In summary, the flexibility offered by localized heating allows us to provide comparable, and sometimes better, thermal comfort to the occupants, at a significantly reduced energy consumption level.
4.4 Occupant Positioning

In this section, we describe the positioning system that we designed and implemented to support the proposed localized heating system. The key objectives in the design of our positioning system were to avoid intrusiveness (users should not have to carry tags), minimize cost (easily deployable using low-cost, off-the-shelf components), and preserve privacy (avoid uniquely identifying users).

4.4.1 System Outline

There are two versions of our proposed positioning system. Both systems rely on arrays of sensors placed at fixed positions (in a grid pattern) on the ceiling throughout the building. The first version utilizes ultrasonic distance sensors, whereas the second version uses passive infrared (PIR) motion sensors. We provide a comparison between these two versions at the end of this section.

Ultrasonic Distance Sensor Positioning

The first proposed positioning system is an ultrasonic distance sensor-based positioning system, summarized in Figure 4.5. An array of ultrasonic distance sensors, arranged in a grid, is attached to the ceiling of the building. Each sensor outputs the distance between itself and the closest object within its range. Clusters of sensors are managed by microcontrollers that communicate wirelessly with a central node, which processes the sensor inputs to obtain the occupant positioning information within the building. We discuss implementation details for our prototype in Section 4.5.

The positioning system is first calibrated with a baseline reading of all sensors to take into account all inanimate objects (such as furniture) in the indoor environment. When a human steps into the range of a distance sensor, its distance reading changes (lower than the baseline reading), indicating the presence of an occupant under the
sensor. The positioning system utilizes a preset threshold for the change in sensor output. When the reading from a sensor changes by more than this threshold, the system detects this event as an occupant moving into the range of the sensor. This threshold may be calibrated to account for a range of occupants, including adults, children, and pets. The underlying positioning algorithm is summarized in Figure 4.6.

The positioning system collects all current sensor readings, compares the readings to the respective baselines, and computes the positions of all occupants in the building based on these comparisons. To cater to changes in the indoor environment (e.g., adding, removing or shifting furniture), the positioning system can be set to recalibrate the baseline readings either by an explicit command from the user or automatically at specified times.

### Passive Infrared Motion Sensor Positioning

The second proposed positioning system foregoes ultrasonic distance sensors in favor of PIR motion sensors. The motion sensors are attached to the ceiling and detect occupant motion (as opposed to distance to the nearest object) within their respective

---

**Figure 4.5**: Overview of the proposed ultrasonic sensor-based positioning system.
Figure 4.6: Overview of the positioning algorithm for the proposed ultrasonic distance sensor-based positioning system.

coverage areas. Figure 4.7 illustrates how the positioning system works when an occupant moves from one part of the building to another.

When the occupant starts to move, the motion sensors nearest him will be activated. As he moves within the building, the motion sensors along his path will be activated in the sequence shown in Figure 4.7. After the occupant reaches his destination, none of the motion sensors will be activated, since there is no detectable motion. Thus, it is up to the control system to trace the occupant’s path to determine his position.
Figure 4.7: Overview of the proposed motion sensor-based tracking system. Darker area denotes more recent sensor activation (at the end of movement).

We developed an algorithm, illustrated in Figure 4.8, to monitor the activation of the motion sensors and derive the resulting occupant positions. In our algorithm, a trace is a sequential set of coordinates that represents the path of a moving occupant, whereas a stop is the coordinate of a stationary occupant. The current positions of the occupants can be derived from the most recent coordinates of all traces and the coordinates of all stops. The input to the algorithm consists of Boolean maps, i.e., two-dimensional arrays of Boolean values corresponding to the array of motion sensors on the ceiling, where a true value denotes the activation of the motion sensor at the current time. In the pre-processing phase, the algorithm derives the list of coordinates for sensors that were activated in the current timestep. This is achieved using an exclusive-OR operation on the Boolean maps for the current timestep and the previous timestep.

In the classification phase, the algorithm attempts to classify the list of newly activated sensors. For each newly activated sensor position, the algorithm first attempts
Figure 4.8: Overview of the positioning algorithm for the proposed motion sensor-based positioning system.
to extend current traces, i.e., the sensor activation is due to an occupant continuing on his path. If there are no suitable traces to extend, the algorithm then tries to create new traces from existing stops, i.e., the sensor activation is due to a previously stationary occupant starting to move. Finally, if the above two steps are not possible, the algorithm creates a new stop at the sensor position, i.e., there is a new occupant entering the building.

In the post-processing phase, the algorithm determines if any current traces can be converted to stops, i.e., a previously moving occupant has reached his destination and is now stationary. This is achieved by comparing the time between the last recorded movement for each trace with the current time. If a fixed threshold of time, \( \tau \), has passed, the trace is converted to a stop. The final occupant positions are then derived from the list of traces and stops at the end of the algorithm.

### 4.4.2 System Comparison

Both versions of our positioning system are equally capable of providing accurate occupant positioning information. However, there are advantages unique to each version, and these differences might affect the choice of the positioning system in a given scenario.

Ultrasonic sensor-based positioning requires minimal post-processing effort, apart from comparing the sensor readings with the calibrated baseline readings. This is evident in the much reduced positioning algorithm for the ultrasonic sensor-based system (see Figure 4.6) compared to that for the motion sensor-based system (see Figure 4.8). However, since ultrasonic sensors are active sensors (they function by sending ultrasonic signals, then waiting to receive the reflected signal), much effort must be taken, in a large-scale deployment environment, to reduce the interference between the sensors. This is usually accomplished by staggering the operation of each sensor, so that no two sensors in close proximity will operate at the same time.
The result of this is a reduced sensing frequency, to around one complete reading (of all the sensors in the building) per second. In addition, ultrasonic sensor-based positioning is not ideal for spaces with many tall objects (such as a library with many tall shelves); since the system depends on distance readings (compared to a baseline) to determine the presence of an occupant, the distance readings will be permanently low due to the presence of these tall objects.

Motion sensor-based positioning resolves some of the issues related to ultrasonic sensor-based systems. PIR motion sensors are passive sensors and do not interfere with one another. In addition, since these sensors detect motion, they can function even in the presence of tall objects, as long as the objects do not completely obscure the sensors. Motion sensors are also cheaper than ultrasonic distance sensors, typically around half the price of an ultrasonic sensor with comparable specifications. The complexity of the positioning algorithm for motion sensor-based positioning is its key disadvantage. However, this allows for more creativity in designing positioning algorithms for different implementation scenarios. We believe that the advantages of motion sensor-based positioning outweigh its disadvantages. Thus, we focus our ROI analysis on motion sensor-based occupant positioning. The decision about which positioning system to deploy is ultimately up to the user.

4.5 Prototype Implementation

To evaluate the feasibility of our localized heating system, we implemented a small-scale proof of concept for a single room, which we describe in this section (larger-scale evaluations using the EnergyPlus building simulator [22] are presented in Section 4.6.1). In addition, we present some observations based on our prototype implementation.
4.5.1 Occupant Positioning

We implemented both versions of our proposed occupant positioning system (see Section 4.4). We chose a larger-scale implementation for the motion sensor-based positioning system due to its lower cost and potential for improvement based on the applied tracking algorithm.

Ultrasonic Distance Sensor Positioning

The components used in our ultrasonic sensor-based positioning system implementation are shown in Figure 4.9(a). We use the Maxbotix LV-EZ0 ultrasonic distance sensor \[101\], controlled by an Arduino Mega microcontroller \[102\]. We outfit the microcontroller with an XBee \[103\] for communications using the 802.15.4 wireless communications protocol. Another XBee chip is connected to a control node (personal computer), allowing us to control the positioning system wirelessly and view readings taken by the system.

![Figure 4.9: Occupant positioning system components.](image)

Based on our implementation, we determined that the distance sensor we used was able to detect objects within a 20 cm radius cylinder from the sensor to the
floor. Thus, sensors were arranged in a regular matrix array, 40 cm apart. Note that this detection radius is specific to the ultrasonic distance sensor that we have selected (Maxbotix LV-EZ0); other sensor models will have different detection radii. Since there were no objects within the test deployment area, the baseline distance for all sensors was 2.5 m. When an occupant is present within a sensor’s range, the sensor value drops, depending on the occupant’s height (e.g., to 0.8 m). Experimental evaluations of the positioning system indicated that it was able to detect a human within its sensing area with consistent accuracy.

**Passive Infrared Motion Sensor Positioning**

The components used in our motion sensor-based positioning system implementation are shown in Figure 4.9(b). We use a Parallax PIR motion sensor [104], controlled by an Arduino Fio [105] microcontroller equipped with an XBee chip [103] for wireless communication. An XBee chip connected to a personal computer (PC) was used to receive sensor readings from the positioning system. In our prototype, the tracking algorithm was implemented on the PC for convenience. However, the tracking algorithm can be deployed in a low-power processing unit, such as the BeagleBoard [106], without affecting its function.

The rated detection radius of the motion sensor used in our implementation is 4.6 m [104]. Since we do not require such a large detection radius, we manually reduced it to 0.3 m by covering a portion of the fresnel lens of each sensor, as shown in Figure 4.9(b). Note that this detection radius can be modified depending on the actual implementation requirements.

We implemented the motion sensor-based positioning system in a 5.4 m × 3.6 m section of a living room. Since each sensor covers a radius of 0.3 m (a diameter of 0.6 m), a 9×6 sensor array was required to cover the entire space. We use one
Figure 4.10: Motion sensor-based positioning system implemented in a 5.4 m × 3.6 m section of a living room, using a 9×6 sensor array.

Arduino Fio microcontroller and XBee chip for each 3×3 array of motion sensors. Our prototype implementation is illustrated in Figure 4.10.

We performed three sets of experiments to test the accuracy of the positioning system. Figure 4.11 illustrates the paths taken in our tests. The first test is a straight-line test that involves an occupant moving in a straight line to a fixed location in the room, stopping for ten seconds, then exiting the room. The second test involves an occupant moving to a fixed location in the room (A), stopping for ten seconds, then exiting the room. This is similar to the first test except that, in this case, the occupant is not just moving in a straight line (see Figure 4.11). The third test involves an occupant moving between two fixed locations (A and B) in the room, stopping for ten seconds at each location. The two points, A and B, were the locations most frequented by the occupants in our testing environment: a PC is located at A, and B represents the best spot on the couch.

To evaluate the accuracy of the positioning system, we introduce two metrics: the trace deviation metric, α_t, measures the deviation of the detected path from the expected path (at fixed intervals), whereas the stop deviation metric, α_s, measures the
Figure 4.11: Paths taken in positioning system tests.

distance between the detected stationary positions of the occupant and the expected positions (essentially, the error in the positioning algorithm). We first determined the expected paths for the three tests described above. The detected paths are then compared to their respective expected paths. Figure 4.12 illustrates a sample path, where $d_k$ represents the distance between the detected path and the expected path at the $k$-th interval. The distance is computed as the Euclidean distance between the two points. When computing this distance, we take into account the width of the occupant, approximated as a 0.3 m radius circle. Thus, any detected point within this circle is considered ideal (in Figure 4.12, $d_2$ is zero). For each path, we define the stops as follows: end of the straight line for the first test, point A for the second test, and both points A and B for the third test. The Euclidean distance between the detected stop and the defined stop, for the $k$-th stop, is denoted as $d_{s,k}$. Given a path with $N$ intervals and $M$ stops, the metrics are defined as follows:

$$\alpha_t = \frac{1}{N} \sum_{i=0}^{N} d_i \quad \text{and} \quad \alpha_s = \frac{1}{M} \sum_{i=0}^{M} d_{s,i}$$
where $\alpha_t$ and $\alpha_s$ are expressed in meters (m), since they are averages of distance computations. Note that all $d_i$ and $d_{s,i}$ values are positive, since they represent Euclidean distances. Thus, all computed $\alpha_t$ and $\alpha_s$ values are positive. $\alpha_t$ and $\alpha_s$ represent the average error in the tracking algorithm for traces (or paths) and stops, respectively. We believe that $\alpha_s$ is the more important metric of the two, since the average occupant spends a significantly greater amount of time stationary as opposed to constantly moving around the building.

![Figure 4.12: Metrics used in the evaluation of the positioning system.](image)

**Table 4.2: Results of occupant positioning system evaluation**

<table>
<thead>
<tr>
<th>Test</th>
<th>$\alpha_t$ (m)</th>
<th>$\alpha_s$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Max</td>
</tr>
<tr>
<td>Path 1</td>
<td>0.066</td>
<td>0.117</td>
</tr>
<tr>
<td>Path 2</td>
<td>0.332</td>
<td>0.555</td>
</tr>
<tr>
<td>Path 3</td>
<td>0.281</td>
<td>0.528</td>
</tr>
</tbody>
</table>

For each of the three tests, we performed ten trials. The measured metrics were computed as the average across these trials. The results from our experiments are presented in Table 4.2. We observe that the positioning system has no problem accurately detecting an occupant moving in a straight line (path 1). Further, there is little variation in both metrics for the latter tests (paths 2 and 3), indicating that the system is equally capable of locating occupants moving within a space as well as those entering and exiting the target space. There is a consistent error (of around 0.3 m) in the detected path, but this is alleviated by the accurate stop detection, indicated
by the low $\alpha_s$ values. We note that this error can be reduced further through the optimization of the tracking algorithm.

In summary, the experimental results are positive, and the system is capable of accurately locating and positioning occupants for our applications.

### 4.5.2 Implementation Considerations

The prototype implementation described in the previous section suggests that motion sensors for the positioning system need to be placed 60 cm apart. A naive solution would be to place the sensors in a regular grid throughout the building. Figure 4.13 illustrates the floorplan for a model 111 m$^2$ two-bedroom, two-bath apartment. We use the same model home for our simulations in the next section. Under the aforementioned scheme, this apartment would require 270 motion sensors. However, we can significantly reduce the number of sensors needed for our positioning system through several considerations. The optimized placement, illustrated in Figure 4.13, is discussed next.

![Figure 4.13: Sensor and heater placement for a 111 m$^2$ apartment.](image)

First, we remove the sensors covering areas that occupants cannot get to, *e.g.*, above a dresser or refrigerator. Next, we remove the extra sensors covering narrow
corridors, along the walls of the building, and the area within beds. This is because the width of the occupant (typically from 40 to 60 cm) is such that he can be easily located by the next available sensor. Through the operations discussed above, we reduce the number of sensors needed for the 111 m$^2$ home from 270 to 190.

Note that, in Figure 4.13, the sensors placed over the dining table are preserved. This is to accommodate changes in furniture placement that occur around the dining table, e.g., moving chairs and shifting the table. In general, when such changes in the landscape are expected, the sensors should be retained.

Figure 4.13 also shows possible placement positions for the radiant heaters used in localized heating. In our implementation, we take into consideration the expected occupancy levels of each room. A total of 14 directed radiant heaters are sufficient to cover the apartment. Additional heaters can be installed to support more occupants in each room or to cover more heavily populated areas.

4.6 Simulation Results

In this section, we analyze the savings attributable to localized heating, discuss its implementation cost, and provide a comparison with other related technologies in the field of building heating energy conservation.

4.6.1 EnergyPlus Simulation and Savings

In order to analyze the potential savings from implementing a localized heating system, we used EnergyPlus, a commonly used building energy simulator distributed by the United States Department of Energy. We performed our analysis on a publicly-available set of building models and climate zones that represent 70% of the building stock in the United States. From this set, we determined that two apartment models (high-rise and mid-rise), along with three office models (large, medium, and
small), were the most suitable for localized heating. We selected climate zones that required significant building space heating, omitting climate zones, such as Riyadh in Saudi Arabia, that do not require heating. The climate zones we selected are as follows: Baltimore (Maryland), Boise (Idaho), Burlington (Vermont), Chicago (Illinois), Duluth (Minnesota), Fairbanks (Alaska), Helena (Montana), Salem (Oregon), and Vancouver (Canada). We sought to determine the energy savings, in Joules (J), cost savings, in U.S. dollars ($), and carbon emissions reductions, in pounds carbon dioxide equivalent (lb CO₂), that can be attributed to localized heating when compared to purely central heating. Table 4.3 lists the input parameters that were used in our savings analysis. In our analysis, we assume that the central heating system consumes natural gas, whereas the localized heating system consumes both electricity (by the radiant heaters) and natural gas (by the central heating system to maintain the minimum temperature).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electricity price</td>
<td>$0.1188/kWh [71]</td>
</tr>
<tr>
<td>Gas price</td>
<td>$1.265/therm [71]</td>
</tr>
<tr>
<td>Emission factor (Elec.)</td>
<td>1.329 lb CO₂/kWh [107]</td>
</tr>
<tr>
<td>Emission factor (Gas)</td>
<td>11 lb CO₂/therm [107]</td>
</tr>
<tr>
<td>Radiant heater factor</td>
<td>13.6 W/°C [94]</td>
</tr>
</tbody>
</table>

Costs and emission factors were derived from the U.S. national averages [71, 107]. The radiant heater factor defines the additional power required by the radiant heater per °C of temperature difference (see Section 4.3.2).

We used the coarse-grain occupancy schedules embedded in the building models to estimate the number of occupants and, thus, the number of radiant heaters that need to be turned on for each simulation timestep. These schedules differ according to building type, e.g., the occupancy schedules for the simulated apartments are different

*Recall that our localized heating approach requires central heating to maintain a minimum temperature, e.g., 10°C.
from those for the offices. In addition, there are more occupants in larger buildings compared to smaller ones (further information on the simulation models can be found in [56]). The additional electricity consumed by localized heating is determined by the following equation:

\[ E_e = \sum_{t=0}^{N} [O_t \times (T_d - T_z) \times f_T \times \Delta s] \]

where \( E_e \) is the additional electricity consumed by localized heating (in Joules), \( N \) represents the total number of simulation timesteps, \( O_t \) is the number of occupants in the building at timestep \( t \), \( T_d \) and \( T_z \) represent the occupant’s desired (comfort) temperature and the current zone temperature (in °C), respectively, \( f_T \) is the radiant heater factor discussed previously, and \( \Delta s \) is the simulation timestep interval (in seconds). We used a desired (comfort) heating temperature of 22°C in our simulations.

All the information needed to perform the above computation was derived from the EnergyPlus standard output files. The simulation workflow is as follows. First, we simulate the building model with central heating in EnergyPlus to derive the base case. Second, we simulate the building model for localized heating, with a reduced central heating setting of 10°C, in EnergyPlus (simulations using higher central heating settings are discussed later in this section). Third, we post-process the output file from EnergyPlus for the second simulation and determine the additional localized heating energy. This additional energy is then combined with the heating energy values we obtained from the EnergyPlus simulation. Fourth, the total energy consumption is used to determine the cost and carbon emissions attributed to both central and localized heating, using the parameters from Table 4.3. Finally, we compare the energy consumption, cost, and carbon emissions of localized heating with those of central heating.
The simulation steps described above are repeated across the different building categories and climate zones. We present our simulation results in Figure 4.14, arranged according to building category. We observe that localized heating is most efficient in the apartment models, potentially achieving up to 76% energy savings and 59% cost savings. This is largely due to the fact that these apartments are sparsely populated, with only two to three occupants per 88 m² space. Conversely, in densely populated buildings, such as a small office, the savings that can be achieved by localized heating, although reasonable, are significantly reduced to around 39% energy savings and 22% cost savings. In general, greater energy savings can be achieved in sparsely populated buildings due to the significantly reduced heating area under localized heating. However, in all scenarios, there are positive energy savings, cost savings, and carbon emissions reductions. The medium-sized office building model includes heating systems that depend on both electricity and natural gas. Note that electricity costs 2.5 times as much as natural gas per unit energy based on our sim-
ulation parameters. Using localized heating, both electricity and natural gas can be saved, resulting in higher energy savings than cost savings. Taking the average across all simulated building categories, the potential energy and cost savings, under localized heating, are 52% and 46%, respectively. Note that the actual cost savings and emissions reductions will vary according to the energy costs and emissions factors of the target location (our results are based on U.S. national averages [71, 107]). In addition, the real-world savings from localized heating will vary from our simulation results according to the actual building characteristics. We provide simulation results as an indicator of the potential savings that can be derived by using localized heating.

In order to analyze the savings impact of different central heating temperature settings, we performed further simulations on the model 111 m² apartment, presented in Figure 4.13 across the same set of climate zones. This is to accommodate users who are initially apprehensive about the low indoor ambient temperature. For this analysis, we chose the following central heating settings: 10°C (standard localized heating setting), 12.5°C, 15°C, 17.5°C, and 20°C. Figure 4.15 illustrates the results of our simulations, averaged across the set of climate zones. From Figure 4.15, we observe that heating energy savings of 18% can still be achieved at a central heating setting of 20°C, with radiant heating covering the remaining 2°C. Thus, new users can set the central heating system to a higher temperature setting while still achieving considerable energy savings. As the users get accustomed to localized heating over time, this central heating setting can be reduced to enable additional energy savings.

Also illustrated in Figure 4.15 is the savings that can be achieved from a dual radiant heater setup (see Figure 4.3). In this setup, two radiant heaters are used to heat each occupant, providing uniform heating. Although the total power output from these heaters would be similar to that from a single radiant heater, we assume a doubling of the power output in our analysis. This allows us to analyze the savings enabled when a higher level of radiant heating is required. From Figure 4.15, we
Figure 4.15: Savings from localized heating for different central heating settings. Savings from a dual radiant heater setup is also shown.

observe that energy savings from a dual radiant heater setup are still substantial (65% compared to 74%). Cost savings is reduced (39% compared to 61%) due to the higher electricity costs (2.5 times more than gas). Overall, the dual radiant heater setup is promising as it affords better thermal comfort (see Section 4.3.3), along with significant energy and cost savings.

4.6.2 Implementation Cost and ROI

The ROI is typically one of the primary concerns for building owners when considering new energy conservation technologies. ROI analysis involves estimating the time needed for the investment to pay for itself, considering implementation costs and expected savings. In the case of localized heating, the implementation cost comprises two parts: the positioning system and the radiant heaters. We performed ROI analysis on our model 111 m$^2$ home using the Chicago climate zone, based on the implementation scheme shown in Figure 4.13. We use the motion sensor-based posi-
tioning system in our ROI analysis. Note that the ROI will vary depending on actual building characteristics, occupancy, and climate.

Table 4.4: Return on investment (ROI) computations

<table>
<thead>
<tr>
<th>Item</th>
<th>Unit cost</th>
<th>Qty.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIR motion sensors</td>
<td>$2.50</td>
<td>190</td>
<td>$475</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>$30</td>
<td>22</td>
<td>$660</td>
</tr>
<tr>
<td>Radiant heaters</td>
<td>$100</td>
<td>14</td>
<td>$1,400</td>
</tr>
<tr>
<td>Installation cost</td>
<td>$20/hour</td>
<td></td>
<td>$480</td>
</tr>
<tr>
<td>Total cost</td>
<td></td>
<td></td>
<td>$3,015</td>
</tr>
<tr>
<td>Annual savings</td>
<td></td>
<td></td>
<td>$647</td>
</tr>
<tr>
<td>Annual maintenance</td>
<td></td>
<td></td>
<td>$42</td>
</tr>
<tr>
<td>ROI</td>
<td></td>
<td></td>
<td>5.0 yrs</td>
</tr>
</tbody>
</table>

We break down the costs of localized heating into its components in Table 4.4. We assume that the wholesale costs of PIR motion sensors are half of that of their retail cost, i.e., each motion sensor costs $2.50. In addition, we assume microcontrollers with wireless functionality to cost $30 each, and radiant heaters, with attached motors and controllers, to cost $100 each. These cost estimates are based on our prototype implementation costs (see Section 4.5). Since each microcontroller handles nine motion sensors, our implementation requires 22 microcontrollers. Additionally, we assume an installation cost of $20 per hour, for a total of 24 hours. Based on our implementation experience, this is a reasonable estimate.

From our simulations, the estimated annual cost savings from localized heating is $647. We include an annual maintenance cost to factor in the cost of replacing infrared bulbs. These bulbs have a rated life of approximately four years, when used only during the winter months, and cost around $12 each. Amortized over four years, the annual maintenance cost for 14 radiant heaters is $42. Based on the above information, we expect the localized heating system to pay for itself in five years. The ROI for localized heating is reasonable, reflecting the feasibility of localized heating from a cost savings perspective. Note that the homeowner stands to benefit from significant annual cost savings beyond the ROI years. The ROI for localized heating
is comparable to that for solar panels (see Chapter 3). However, localized heating reduces natural gas consumption whereas renewable energy generation indirectly reduces electricity consumption.

In our ROI analysis, we assumed that the positioning system will be used solely for the purpose of occupant heating. However, the same positioning system can also be used for other purposes. For example, it can be used as part of an appliance management system that automatically shuts appliances off when the occupant leaves the area, resulting in reduced energy wastage. Another possible use of the positioning system is to track elderly people living alone and alert caregivers when anomalies are detected. These alert systems typically cost around $30 per month [108]. Thus, the actual utility of the positioning system is underestimated in our evaluation. In Chapter 5, we describe two additional applications of the positioning system: occupant-level indoor air quality management and smart lighting.

4.6.3 Comparison with Other Solutions

Using the same model home and climate zone (Chicago) that are subjected to our ROI analysis in Section 4.6.2, we performed savings analysis on programmable thermostats and compared the results with localized heating. Table 4.5 summarizes these results. Since programmable thermostats enable savings from building cooling in addition to heating, we combine the energy, cost, and carbon emissions for both heating and cooling in our comparison. The savings values are presented as potential savings versus a conventional central heating and cooling solution. For our simulations, we assumed that the programmable thermostat was set according to the Energy Star guidelines [109].

Based on the results from Table 4.5, we see that even though programmable thermostats enable savings with building cooling in addition to heating, the savings lag that of localized heating. This is primarily due to the much lower cooling energy,
Table 4.5: Potential savings for localized heating vs. programmable thermostat

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Localized heating</th>
<th>Programmable thermostat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy savings</td>
<td>86.6%</td>
<td>38.9%</td>
</tr>
<tr>
<td>Cost savings</td>
<td>72.6%</td>
<td>40.2%</td>
</tr>
<tr>
<td>Emissions reductions</td>
<td>65.1%</td>
<td>40.9%</td>
</tr>
</tbody>
</table>

compared to heating, in the Chicago climate zone. For our simulations, the total cooling energy consumption is only 8% of heating energy consumption. Note that programmable thermostats have a much lower ROI due to their low cost. In addition, for warmer climates, programmable thermostats might be more useful than localized heating. However, as we previously discussed in Section 2.1.1, reliance on human involvement in their operation may significantly reduce their potential savings.

Without actual occupant profiles and positioning data, it is difficult to estimate the potential savings from zonal heating and smart thermostats. However, smart thermostats claim an additional 15% in energy savings over conventional programmable thermostats [15], due to automation of the process of temperature setting. From Table 4.5, we can see that the energy/cost savings and emission reductions from the smart thermostats would still be much less than those of localized heating.

### 4.7 Chapter Summary

In this chapter, we proposed a localized heating system that significantly reduces the energy consumed in heating buildings. In addition, we described two versions of an indoor positioning system, capable of accurately locating building occupants, that can be used to support localized heating. Our analysis confirms the feasibility and effectiveness of localized heating, as well as other occupant-based solutions.
Chapter 5

Occupant-level Sensing

Existing building systems, such as heating, ventilation, and air conditioning (HVAC), operate using rudimentary sensing and control mechanisms. Currently, indoor environmental sensors are placed at the building or zone level, resulting in very coarse-grain data on which control decisions are based. Since the ultimate goal of these building systems is to provide occupant comfort, an important next step in the evolution of intelligent building systems is the shifting of these sensors to the occupants themselves.

In this chapter, we describe an occupant-level sensing (OLS) system. The OLS system comprises a user-carried device including a number of environmental sensors, along with the appropriate infrastructure support. The information provided by the OLS system reflects the exact conditions that each occupant experiences. OLS enables novel applications that reduce or eliminate energy wastage in building subsystems by allowing them to expend just the right level of effort, while simultaneously improving the comfort level of occupants by bridging the gap between sensor readings and user experience.

We then explore some applications enabled by OLS and/or occupant positioning: improved localized heating, occupant-level indoor air quality management, and
occupant-level smart lighting. Improved localized heating, combining OLS with the localized heating system described in Chapter 4 provides guarantees for occupant comfort, along with significant heating energy savings. Occupant-level indoor air quality management detects the air quality around each occupant and regulates the ventilation system accordingly. Occupant-level smart lighting controls indoor lighting based on the occupants’ positions and preferences. We present a prototype implementation of the OLS system, and discuss some results obtained from our implementation. We then evaluate the energy savings potential of occupant-level smart lighting.

5.1 Introduction

Traditional HVAC systems rely on a small number of sensors installed at specific locations around the building. These sensors are limited to detecting the conditions in their immediate environment, leading to disparities between the sensor readings and the actual conditions faced by the building occupants, e.g., a thermostat placed at the far end of the room will not accurately reflect the exact conditions in the entire room. We performed an experiment to determine the temperature variations in a given building. Temperature sensors (built into the Texas Instruments EZ430-RF2500 development platform [110]) were placed at different locations in a 111 m² apartment to track the temperature over time. Figure 5.1 illustrates this sensor placement, along with the position of the thermostat in the apartment.

Sensor data were collected every thirty seconds, over a fifteen hour time period. Figure 5.2 illustrates the results of this experiment. Throughout the experiment, the external temperature varied between 0°C and 18°C; the building heating system was set to 22°C. From the figure, we observe that the temperature in the building is generally well regulated. However, in some areas, such as bathroom 1 and bedroom 2, the temperature fell to a level that would be uncomfortable to the building
occupants for extended periods of time, e.g., in bedroom 2, the temperature fell to below 20°C for approximately six hours. In addition, some areas, especially bedroom 1, are frequently over-conditioned (the temperature is consistently higher than the 22°C setting), resulting in energy wastage if these are the only occupied areas in the building.

Figure 5.2: Temperature variations in the building, each timestep represents thirty seconds. External temperature between 0°C and 18°C.
We believe that the next frontier in building energy conservation requires sensing at the level of the eventual target: the occupant. Occupant-level sensors refer to sensors that are carried by all occupants in the building. These sensors provide specific information about the occupant’s location and his immediate environment. This is made possible by miniature sensors and low-power wireless communication technology. As these technologies become commonplace, it is now possible to integrate these sensors into wearable systems (e.g., a wristwatch or badge) or mobile devices that are carried by users (e.g., mobile phones).

In this chapter, we present an OLS system that is capable of providing information on the surrounding environmental conditions of all occupants in the building. OLS can be combined with occupant positioning (see Section 4.4) to provide specific information on the occupants and their locations. Previous efforts have focused on tracking occupancy of different regions within a building or, in some cases, tracking occupant positions. Our work is the first effort that considers the precise environmental conditions experienced by each occupant, along with occupant positioning information, to improve building energy efficiency. We discuss three novel occupant-aware building applications enabled by OLS and/or occupant positioning: improved localized heating, occupant-level indoor air quality management, and occupant-level smart lighting. In addition, we present a prototype implementation of the OLS system, and provide an evaluation of the system. We evaluate the energy savings potential of occupant-level smart lighting using a simulation tool, incorporating fine-grain occupant positioning information, that we developed.

The key contributions of this chapter are as follows:

- Proposal and in-depth analysis of an OLS system.
- Exploration of applications enabled by OLS.
- Prototype implementation and evaluation of OLS system.
Proposal and evaluation of occupant-level smart lighting system.

5.2 Background

Although much work has been done in the field of wearable sensors, the large majority of this work focuses on monitoring the user’s health [111, 112, 113, 114]. Our work is the first of its kind to monitor the direct surrounding environment of the occupants for building energy management purposes, using metrics such as surrounding temperature, humidity, and indoor air quality.

The proposed OLS system is meant to augment other building systems, such as localized heating, by providing better feedback to these systems. As far as possible, these building subsystems should function normally even if the occupants are not wearing their devices. Thus, although incorporating a tag-based occupant positioning system (e.g., the Cricket system [92]) alongside OLS would be ideal in reducing implementation effort, since the occupants will already be carrying devices for sensing, occupant positioning information would be unavailable if the occupants neglect to carry their devices, i.e., building systems that depend on occupant positioning, such as localized heating, would not be able to function. We consider the tradeoffs between different implementation schemes in Section 5.3.2.

5.3 Occupant-level Sensing

In this section, we describe an OLS system for reducing energy wastage in buildings. This novel sensing and control system augments other building systems, providing sensor information obtained from the direct surrounding environment of each occupant.
5.3.1 System Outline

The OLS system comprises two parts: a user device carried (or worn) by all occupants in the building and an infrastructure support system to communicate with the user devices and aggregate and act on the received sensor data.

User Device

The user device is a lightweight, low-power device carried by occupants in a building at all times. Sensor readings from the user device provide the control system with a clear picture of the environmental conditions that each occupant is experiencing. This information is sent wirelessly through a low-power wireless communications channel, such as the 802.15.4 wireless communications protocol. Figure 5.3 outlines the components of the user device and their corresponding functions.

![User device components and functions.](image)

The primary function of the user device is to measure the environmental conditions surrounding the occupant. Specifically, sensor data from the user device are used to monitor the occupant’s thermal comfort, using temperature and humidity sensors, and surrounding air quality, using gas sensors. An improved localized heating system that uses OLS information to ensure the occupants’ thermal comfort is described in Section 5.4.1. To monitor indoor air quality, the user device is outfitted
with air quality sensors to measure various parameters such as carbon dioxide (CO₂) concentration. Additional gas sensors, such as carbon monoxide (CO), radon, and particle pollution sensors, can be added as required. The application of OLS to indoor air quality management is further discussed in Section 5.4.2. Apart from providing information about the occupant’s location and surroundings, the user device can also be used to relay feedback from the occupant to the control system through a simple user interface.

The user device can be used for occupant location and tracking. The Cricket positioning system [92], a lightweight, low-power occupant location system, is particularly suited to this application, since the occupant would already be carrying an OLS device. We discuss the advantages and disadvantages of such an approach later in this section.

The transmission rate of information from the user device to the processing node depends on the information that is being transmitted. Since the system requires fast response to changes in the occupant’s location, if a positioning system is deployed in OLS, positioning information should be transmitted at a rate of at least 1 Hz. Environmental parameters and air quality information do not need to be updated every second. One transmission every 1 to 5 minutes would be sufficient for these categories of information (for evaluation purposes, we use a transmission period of 15 seconds in our prototype implementation in Section 5.5). The low sampling rates can be exploited to reduce the energy consumption of the user device.

**Infrastructure Support**

Infrastructure support is needed to receive the sensor data sent by the user devices and act on these data to control the building subsystems. Figure 5.4 outlines the required components for infrastructure support and their respective functions.
The processing node is the central node to which all sensor information is sent. For each user device, the processing node keeps track of the measured occupant location, the various sensor readings, and the time of the last received signal. Using this information, the processing node computes and sends the required outputs to the building subsystems. The user devices rely on low-power, limited-range wireless channels, such as Zigbee, to communicate sensor information to the infrastructure support. For large buildings, a network of relay nodes (such as a mesh network) is needed to receive and relay the sensor information to the processing node.

The infrastructure support must also be able to communicate with and control existing building subsystems. This can be achieved using existing building automation solutions, such as the Insteon suite of products [115] or an alternative proprietary solution. The components required will depend on the applications needed.
5.3.2 Implementation Considerations

In Figure 5.5, we present a possible implementation of the OLS system in a model 111 m$^2$ apartment. Also illustrated in the figure are the paths taken for sensor information to travel from the user devices to the processing node. Relay nodes are used when the processing node is out of range. The processing node is placed in a central location to reduce the number of hops needed.

![Figure 5.5: OLS system implementation in a 111 m$^2$ apartment.](image)

Based on our prototype implementation (discussed later in Section 5.5), we determined that each relay and processing node can reliably cover an area the size of a large room (approximately 40 m$^2$), with light furnishings typical of a residential building. We observe that, in a small apartment, such as the model home, only one processing node and two relay nodes are needed to cover the apartment. The number of user devices needed depends on the number of occupants in the apartment; in the example illustrated in Figure 5.5, there are four occupants in the apartment.

As discussed previously, OLS can be implemented with or without occupant positioning capabilities. We initially designed OLS to be a supplementary system, i.e., OLS serves to improve existing building systems, and these building systems can continue to function normally without OLS. This is primarily due to our belief that
critical building systems should not be susceptible to user error (e.g., when the user forgets to carry his device). However, we also recognize the convenience in implementing an occupant positioning system, such as Cricket, alongside OLS. Ultimately, we leave it to the building owner to decide which system (combined or separate occupant positioning and OLS) is more suitable for his purposes.

5.4 Occupant-aware Building Applications

In this section, we first present two applications of OLS: improved localized heating and occupant-level indoor air quality management. Next, we describe a novel lighting system, called occupant-level smart lighting, that considers the occupants’ positions in building lighting control. This system relies on the infrastructure support present in OLS for lighting control. All three applications are enabled by OLS and occupant positioning, and are examples of occupant-aware building systems.

5.4.1 Improved Localized Heating

Occupant-level sensing improves upon localized heating by incorporating guarantees for thermal comfort. Figure 5.6 shows how this expanded localized heating system works. With real-time information on the surrounding temperature and humidity of the occupants, the output level of localized heating can be adjusted to stay within the occupant’s comfort range [99]. This allows the system to actively track and manage the thermal comfort of the occupants, resulting in both energy savings and improved occupant comfort.

The control algorithm used for improved localized heating is similar to that for localized heating (see Section 4.3.2). The key difference is that, instead of relying on fixed radiant heater output levels (using a factor of 13.6 W/°C) or user-defined...
output levels, improved localized heating can automatically determine the required output levels, using the sensor information obtained from OLS.

### 5.4.2 Occupant-level Indoor Air Quality Management

Another application of occupant positioning and OLS, apart from energy-saving applications, is the fine-grain monitoring of indoor air quality, using air quality sensors installed on the user devices. This is an equally important application because poor air quality exposes occupants to health and safety risks [116]. Further, studies have shown that improvements in indoor air quality lead to improved occupant productivity [117]. Shifting the air quality sensors onto the occupant, as opposed to placing them on different locations within the building, provides a clearer picture of the quality of air that each occupant is exposed to. In this way, we can ensure acceptable air quality for every occupant in the building.

Indoor air quality can be measured using several metrics and corresponding sensors. Of these, CO₂ concentration is the most commonly used indicator of indoor air quality. ASHRAE Standard 62-2001 recommends an indoor CO₂ concentration below...
1000 parts per million (ppm) \[118\]. Thus, in an air quality management system, the ventilation system corresponding to an occupant’s location can be switched on or set to a higher output level when the measured CO\(_2\) concentration rises above this value for that occupant. We discuss the collected CO\(_2\) concentration data for our prototype implementation for OLS in Section 5.5.

### 5.4.3 Occupant-level Smart Lighting

The illuminance of the immediate area around the occupant depends on his location. At the same output level, occupants nearer a lamp will be better illuminated than occupants farther from the lamp. Thus, we identify an opportunity for improvement. In this section, we present a novel indoor lighting system, called occupant-level smart lighting, that takes into account occupant positions in building lighting control. The system computes the optimum output levels, based on the occupants’ locations, that satisfy their requirements for lighting. This computation is performed using a genetic algorithm detailed ahead. The lamps in the building are then automatically set to these computed levels. To incorporate natural lighting into these computations, light sensors are placed near windows to detect the amount of sunlight entering the room. In addition, the feedback mechanism on the user device can be used by the occupant to fine-tune the amount of lighting that he receives. This provides an avenue of control for the occupant over the indoor environment.

The concept of smart lighting has been extensively covered \[38, 39, 40, 41, 42, 119\]. In general, smart lighting systems combine coarse-grain occupancy sensing and dynamic lighting tuning, using a network of sensors, along with the corresponding control algorithms, to determine the lowest output levels for the lighting fixtures in the building that would satisfy the requirements of the occupants. In \[119\], an intelligent lighting system, called Illuminator, is proposed, primarily for use in stage lighting and media applications. Illuminator considers the lighting requirements at specific
locations on a stage and controls the output levels of the corresponding spotlights to satisfy these requirements. In [40] and [41], lighting systems that consider both user comfort and energy efficiency are proposed. In both systems, the lighting levels in the building are controlled based on the occupants’ preferences and their predetermined (fixed) positions in the building. This first group of solutions controls building lighting based on the occupants’ requirements. The second group of solutions relies on coarse-grain occupancy sensors (e.g., motion sensors) and indoor light sensors to determine if a given space in the building is occupied and compute the corresponding lighting level required for that space [38, 39, 42]. Both groups of solutions target buildings with relatively fixed occupant positions, such as offices, since they do not rely on occupant positioning systems to locate building occupants.

Occupant-level smart lighting extends previous work in the area by incorporating requirements-based lighting, natural lighting detection, and fine-grain occupant positioning information in the control of building lighting. The system receives occupant positioning information (from a suitable occupant positioning system) and computes the required lighting output levels in real-time, as opposed to precomputing these output levels based on fixed occupant positions. In this way, occupant-level smart lighting is more flexible, and takes into account the movements of building occupants within the building.

Occupant-level smart lighting depends on control algorithms to determine the optimal lighting settings. Finding the set of lighting outputs that maximizes occupant comfort while minimizing energy consumption is a nontrivial problem. For example, considering a building zone with 10 lamps and 11 possible settings per lamp (from 0 to 10, where 0 corresponds to the lamp being off whereas 10 corresponds to the lamp being fully on), we have $11^{10}$ possible solutions. It is not practical to consider all these possible solutions; especially in an actual implementation, where decision-making has to occur in real-time. Genetic algorithms have been identified as a possible solution to
Figure 5.7: Outline of occupant-level smart lighting control, showing the gene used in the genetic algorithm.

This problem. In [40] and [119], two fairly similar genetic algorithms for constraints-based lighting systems are proposed. These algorithms provide near-optimal solutions with a significantly reduced search space. We have developed a variation of these algorithms for use in our occupant-level smart lighting system. The use of occupant positioning information in our system allows improvements to the original algorithms. An outline of the decision-making process is presented in Figure 5.7.

The set of output settings for lamps in occupied areas forms the gene used in our genetic algorithm. In the first stage, the control algorithm filters out the lamps in unoccupied areas and switches them off (set to 0). This reduces the search space, preventing the generation of suboptimal solutions that involve switching on lights in unoccupied areas. The set of output settings is expressed as \( s = \{s_1, s_2, ..., s_L\} \), where \( s_i \) denotes the setting (between 0 and 10) for the \( i \)th lamp, with a total of \( L \) lamps.
The goal of the genetic algorithm is to find a set of output settings $s$ that satisfies the following objectives:

$$\max\left[\sum_{i=1}^{N} C_i(s, w, p_i)\right], \min[E(s)], \min[W(\hat{s}, s)]$$

$C_i$ represents the comfort metric for the $i$th occupant, based on the lighting outputs, $s$, the set of natural lighting illuminance levels for each window, $w$, and the coordinates of the occupant, $p_i$. $N$ denotes the number of occupants in the building. $E$ is the energy consumption, based on the lamp settings, $s$.

The comfort and energy metrics are typically the only metrics considered in smart lighting systems [40, 119]. We introduce a new metric, $W$, to minimize lamp switching, an equally important metric. Consider a simple example, in which there are two lamps directly above the occupant, each providing sufficient lighting such that only one lamp (fully on) is needed at any point in time. Assume that there is no daylight at this time. Consider two solutions, $s = \{10, 0\}$ and $s = \{0, 10\}$. Both solutions satisfy the required objectives of maximizing comfort while minimizing energy consumption. If at time $t = 0$, we pick the first solution and then at time $t = 1$, we pick the second solution, and continue alternating between these solutions, the problem is evident. First, continuously switching on and off the lamps might damage the light bulbs, shortening their lifespan. Second, this continuous switching might annoy the occupants, going against our objective of ensuring occupant comfort. Thus, our algorithm introduces a new objective to minimize this switching behavior. The switching metric is expressed as $W(\hat{s}, s)$, where $\hat{s}$ represents the set of current output settings for the lamps in occupied areas. An example of computing the switching metric is to take the sum of the differences in settings between $s$ and $\hat{s}$.

By default, the control algorithm seeks, first, to satisfy all occupants’ lighting requirements. The algorithm then chooses the lowest energy solution that does not
result in excessive switching behavior. The resulting output will then be used to control the lamps in the building.

The occupant-level smart lighting control system can be extended to appliances. In such an application, to reduce energy wastage when appliances are left on, the smart appliance system can automatically shut off certain appliances when the occupant moves out of range. Individual profiles have to be created for each category of appliances. Some appliances, like stereos, have longer usage ranges than others, such as fans and computer monitors. Other appliances need to be left on and must be excluded from this automation. Examples include computers, wireless routers, hot water pots, and bedside alarm clocks.

5.5 OLS System Prototype

We implemented a prototype OLS system to test and evaluate the proposed OLS system. In this section, we describe a prototype OLS implementation, providing details of the components used and battery requirements. In addition, we deploy and evaluate the prototype OLS system.

5.5.1 Prototype Implementation

To implement the OLS system, several choices of components are available. We used the following off-the-shelf components for the user device: Arduino Fio microcontroller board [105] (with Xbee chip [103] for wireless communications), Sensirion SHT15 humidity and temperature sensor [120], and Cozir Ambient CO2 sensor [121]. The implemented OLS user device is illustrated in Figure 5.8. We used an XBee connected to a PC as infrastructure support.

Since the user device needs to be portable, its battery life is an important concern. Fortunately, the aforementioned components are low-power: considering a transmis-
Figure 5.8: Components used in the implemented OLS system.

sion period of 15 seconds, the Arduino Fio is rated at around 100 milliWatts (mW) during normal operation (this can be reduced even further through aggressive use of the available low-power sleep feature), the attached XBee chip is rated at around 5 mW, the Sensirion SHT15 is rated at 6 microWatts, and the Cozir CO\textsubscript{2} sensor is rated at 3.5 mW. In all, each user device has a rated power consumption of approximately 110 mW. The components required for infrastructure support are connected to the mains, hence negating any battery concerns. However, to minimize the energy footprint of OLS, we recommend the use of low-power processing devices, such as the BeagleBoard [106]. With all components considered, we expect an additional power load of less than 5 Watts for the implementation of OLS in a small residential building.
5.5.2 Prototype Evaluation

To test and evaluate our OLS system, we deployed the system in a real-world environment. Figure 5.9 illustrates the sensor readings captured over a 24-hour time period (starting at 5pm) for an individual, where each timestep represents 15 seconds. The first set of readings (ending at timestep 3800) was obtained when the individual was at home, followed by a period with no data (all readings set to zero) as he was commuting to work. This is then followed by another set of readings (starting at timestep 4000) from the office.

From Figure 5.9(a), we observe that the CO$_2$ concentration reading is much higher in the home than in the office. This is primarily due to the fact that the home is heated via a convective heating system with an indoor furnace, whereas the office is heated via hot water radiators. In addition, the large swings in CO$_2$ concentration readings at home can be attributed to the various activities that the occupant is undertaking, e.g., the spike in CO$_2$ concentration levels between timestep 1000 and 1500 was caused by cooking on a gas stove. Since the home ventilation system is linked to the heating system, the ventilation system is only switched on when heating is required, regardless of the indoor air quality. The office ventilation system runs separately from the heating system, constantly drawing fresh air into the office. This allows the system to maintain a constant CO$_2$ concentration under 800 ppm.

From Figure 5.9(b), we observe that there is little variation in temperature between the home and office. The heating systems are capable of maintaining the desired (comfort) temperature range (between 22°C and 25°C). Through the data logging period, the external temperatures varied between 8.9°C and 12.8°C. We observe from Figure 5.9(b) that the temperature falls to around 22°C at home during the night. It is observed that the bedroom is typically slightly colder than the living room (where the thermostat is located) for the target home. This is not unusual for buildings with central heating systems controlled by a single thermostat. The individual uses
(a) CO$_2$ concentration over the monitoring period.

(b) Temperature and humidity over the monitoring period.

Figure 5.9: OLS data logging results. Gap in graphs (where readings are zero) occurred when the user was commuting to work.
a humidifier in the bedroom, explaining the higher relative humidity values for the home (especially at night) compared to the office.

From our deployment results, we observe that the individual is exposed to very different indoor environmental conditions through the course of the day. This is primarily due to the fact that existing building systems do not take into account environmental metrics other than temperature. We believe, however, that these other metrics (CO₂ concentration and indoor humidity) are equally important for the comfort of the occupant and should be considered in indoor building conditioning. OLS is an important first step towards this consideration.

5.6 Simulation

In this section, we analyze the potential energy savings that can be realized through occupant-level smart lighting. To assist in this analysis, we developed a building lighting simulation tool that incorporates fine-grain occupant positions, since existing building energy simulators do not support fine-grain occupant positioning-based control systems. The work outlined in this section serves as a precursor to the retrofit-oriented building energy simulator described in Chapter 6.

5.6.1 Simulator Outline

Although it does not support fine-grain occupant information, EnergyPlus provides a strong base for our simulator to build on. Thus, as far as possible, our simulator uses standard EnergyPlus input files, such as EnergyPlus building models and weather data. This allows us to utilize freely available weather data files and the EnergyPlus building model that we created for the model apartment in our analysis of localized heating (see Chapter 4). To incorporate fine-grain occupant positioning information,
we include, as inputs, occupant profiles that define the occupant’s preferences and specific coordinates over the simulated time period (typically a year).

The inputs are passed to the lighting module that we developed to compute the energy consumption under occupant-level smart lighting. This is based on the computed output levels for the lamps in the building, using the control algorithm for occupant-level smart lighting that we previously discussed. The computed set of output levels over time is then passed to a post-processor that summarizes the energy, cost, and carbon emissions metrics required.

The output levels computed from the control algorithms can be used to directly control the actual lamps in a real-world implementation. This can be achieved using the same code base with one key difference: instead of simulator inputs, the real-world implementation would receive actual sensor data. All the relevant computations will be performed in the processing node of the OLS system.

5.6.2 Occupant Profiles

Ideally, to provide the most accurate estimates of savings potential, the occupants in a target building must be tracked over an extended period of time to generate a full occupant profile. This can be achieved using tracking systems such as Cricket [92]. However, this method requires an upfront investment in the tracking system before its benefits can be fully explored.

An easier method for estimating savings potential is to use automatically-generated occupant profiles, based on certain user-defined heuristics. We developed an occupant profile generator for this purpose. The heuristics used in our generator, for a residential building, are as follows:

- On weekdays, the occupants leave home between 9am and 11am and arrive home between 5pm and 7pm.
• On weekends, the occupants leave home between 12pm and 2pm and arrive home between 4pm and 7pm.

• The occupants sleep from 12am to 8am daily. Lighting is not required when the occupant is sleeping.

• Occupants in the building have an 80% chance of remaining in their previous position. This takes into account the fact that occupants do not move much within a building.

These heuristics can be modified by the user, depending on the actual building type and occupant behavior.

5.6.3 Methodology and Results

We ran our simulator on a model 111 m$^2$ apartment (see Figure 4.13 in Chapter 4). The occupant profile used was generated based on the heuristics defined previously. The occupant lighting preferences used in our simulation are presented in Table 5.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lighting preferences (ambient)</td>
<td>50 lux</td>
</tr>
<tr>
<td>Lighting preferences (task)</td>
<td>100 lux</td>
</tr>
</tbody>
</table>

We performed simulations using a selected set of regional weather profiles, namely: Anchorage (Alaska), San Francisco (California), Denver (Colorado), Atlanta (Georgia), Chicago (Illinois), Boston (Massachusetts), Detroit (Michigan), Minneapolis (Minnesota), Newark (New Jersey), New York (New York), Houston (Texas), and Seattle (Washington). We selected these regions to cover a wide range of geographical locations in North America. In our analysis, we seek to determine the lighting energy savings that are enabled by occupant-level smart lighting. The results of our analysis are illustrated in Figure 5.10.
Figure 5.10: Savings from occupant-level smart lighting in different regions.

From Figure 5.10 we observe that lighting energy savings from occupant-level smart lighting do not fluctuate much as there is little variation in the external illuminance values for the selected regions. Taking the average of our simulation results across all regions, we expect around 47% lighting energy savings for the model apartment. The results of our analysis indicate the potential of occupant-aware building systems in reducing building energy consumption while preserving (or improving) occupant comfort.

5.7 Chapter Summary

In this chapter, we presented a novel OLS system capable of providing specific information about occupants and their environment. We also discussed the various applications of the proposed system. Making building systems occupant-aware is an important next step in the evolution of wireless sensor network applications in build-
ing monitoring and control. Occupant-aware building systems allow better occupant comfort and energy efficiency.

Although we have primarily focused on energy-efficient building applications, there are other applications of occupant positioning and occupant-level sensing, *e.g.*, in the areas of personal health monitoring and elderly fall detection. These additional applications can be easily implemented in the modular OLS system framework that we have described. The OLS system framework presented in this chapter serves as a stepping stone for future applications in building automation and occupant monitoring.
Chapter 6

ROBESim: A Retrofit-oriented Building Energy Simulator

Reducing building energy consumption can result in an immediate impact on global energy expenditures and the environment. As we discussed in Chapter [1], although improvements in building architecture and design can help in the long run, retrofitting existing buildings is a more sensible means toward reducing the building energy footprint. Currently, building owners have several retrofit choices, from renewable energy generation to energy-efficient building systems and other energy conservation measures. In order to assist in the retrofit selection process, building energy simulations can be used. These simulations provide energy savings estimates from different retrofits, allowing more informed retrofit choices. Currently available simulators, however, are not retrofit-oriented and are not built to enable comparisons among retrofits. In order to perform these comparisons, users are required to manually create building models for each retrofit, which is obviously a cumbersome process.

In this chapter, we introduce a retrofit-oriented building energy simulator, ROBESim [122], that we developed as part of this dissertation. ROBESim is based on the popular EnergyPlus framework, and relies on EnergyPlus for most of the
supported computations. By using the retrofit modules in ROBESim, the user can quickly and easily generate building models to perform retrofit comparison simulations. We have designed ROBESim to be modular and extensible. We outline these modules and show how developers can build upon our work in ROBESim. In addition, we describe the retrofit module development process and discuss additional ease-of-use enhancements that make building energy simulation possible for users of all experience levels.

6.1 Introduction

Building energy simulators, such as EnergyPlus [22, 23], DOE-2 [24], TRNSYS [25], and eQuest [26], can be used to help building owners decide which retrofits to install in their buildings. There are two main goals of building energy simulation. First, the simulation provides the building owner with the end-use splits that are relevant to his specific building, i.e., information similar to that shown in Figure 1.2 (in Section 1.1), but tailored to the specific building. This allows the building owner to narrow down his retrofit choices to those that are more pertinent to his needs, e.g., in buildings that consume little energy in space heating (i.e., in a warm climate), the building owner should consider retrofits in other end-uses. Second, building energy simulation allows for comparisons between retrofits before implementing them on the building. This allows the building owner to weigh the costs and benefits of each retrofit choice. Building energy simulation, in general, provides sufficient information for the building owner to decide which retrofits to install. As we discussed in Section 2.2, other options are available for more in-depth building energy analysis.

Although most of the aforementioned building energy simulators are freely available (namely, EnergyPlus, eQuest, and DOE-2), they require a certain level of technical expertise to use. Thus, the typical building owner has to rely on trained pro-
fessionals to perform this building energy analysis, resulting in additional costs. The analysis performed by trained auditors is, in general, more customized and, thus, more accurate. However, the additional costs and time investment involved could serve as a significant barrier for many building owners. This is especially true for residential home owners with a lower retrofit budget.

In addition to their complexity, most building energy simulators are not retrofit-oriented, i.e., they do not contain built-in functionality to automatically apply retrofits and compare among different retrofit choices. Thus, users have to manually create building models to simulate across different retrofit choices. This is to be expected, since the primary purpose of these building energy simulators is to provide accurate simulation frameworks that can be built upon by additional software. Nonetheless, the lack of built-in retrofit support results in additional effort on the part of the building owner, providing another barrier to entry.

In this chapter, we introduce ROBESim, a retrofit-oriented building energy simulator based on the EnergyPlus framework. The initial release version of ROBESim is compatible with EnergyPlus v8.0. We chose EnergyPlus because it is freely available, widely used in both academic and commercial applications, and has a comprehensive feature set. In addition, the core design philosophy of EnergyPlus is such that wrappers and add-ons can be easily developed around it; since EnergyPlus relies on text-based input and output files, this allows for easy external invocation by wrapper programs.

ROBESim comprises two parts: the core simulation framework and ease-of-use enhancements (in the form of user interfaces and input file generators). The core simulation framework extends EnergyPlus by allowing built-in definitions and separate computations for retrofits, i.e., a retrofit developer can easily write a module that can be processed in ROBESim for energy analysis. For building owners, along with the release version of ROBESim, we provide a database of popular existing
retrofits and some interesting retrofits that are under development, e.g., better insulating windows, programmable thermostats, and smart thermostats [16, 17, 30, 31]. This ensures that the user will have an immediately available retrofits database for comparison. We discuss the included retrofits in detail in Section 6.5.

Since a number of novel retrofits depend on room occupancy data and/or detailed occupant positioning information, the ROBESim core simulation framework also provides built-in support for occupant positioning information, in the form of occupant profiles. The occupant profiles contain information on the positions of the occupants throughout the simulation period. This information can be used to better gauge the energy savings that can be derived from occupant-aware retrofits, such as localized heating and smart lighting.

To support the ROBESim simulation framework, we include a number of ease-of-use additions. We target all experience levels in our approach, i.e., we have developed interfaces for building owners, retrofit developers, and simulation software developers. Specifically, for casual building owners, we have developed a user interface for quick and easy input, simulation, and comparison across our retrofits database. For retrofit developers, we provide software and libraries that expose all the functionality provided by the ROBESim core. This enables the retrofit developer to quickly develop retrofit modules for deployment in ROBESim. Apart from the development of modules for currently available retrofits, ROBESim can also be used in the preliminary development stages for novel retrofits. We detail examples of some retrofit modules we developed in Section 6.5. For software developers looking to extend our work, we provide ROBESim libraries, along with source code for the interfaces that enable batch application of retrofits and one-click simulation across multiple building models. These simulator usage scenarios are further explored in Section 6.4.

In addition to user interfaces, ROBESim includes a number of input file generators that, based on the provided information, can quickly and easily generate the required
input files (namely, the building model and occupant profiles). The implementation
details for these generators are further discussed in Section 6.7.

The ROBESim core simulation framework and ease-of-use enhancements were
developed using the Java programming language (in comparison, the core EnergyPlus
framework is written in Fortran). We chose Java since it is one of the most commonly
taught programming languages. In addition, for users looking to extend our work,
there is minimal effort required to include and invoke the available ROBESim external
libraries.

In summary, we seek to achieve two goals in our ROBESim design. The first
is to provide quick and easy retrofit development and retrofit-oriented simulation
tools. The second goal is to support retrofits that depend on occupant positioning
information. The key features of ROBESim are as follows:

- **Built-in support for building retrofits.** ROBESim automatically applies the chosen retrofits, and provides comparisons with the original building models.

- **Support for occupant profiles.** For applications that depend on occupant-positioning or occupancy status, ROBESim provides support for generating and utilizing occupant profiles.

- **User-friendly interfaces.** ROBESim includes ease-of-use additions in the form of one-click batch simulators, along with easy input file generation.

- **Modular and extensible.** ROBESim follows the modular coding style of EnergyPlus. All components in ROBESim are modular, and can be easily extended. In addition, ROBESim is programmed in Java, allowing for easier extension for those familiar with the language.

- **EnergyPlus compatibility.** As far as possible, ROBESim retains the EnergyPlus input and output formats, with the exception of additions, such as retrofit
and occupant profile definitions. Further, ROBESim relies on the EnergyPlus simulation framework as much as possible, only performing computations that are not supported under EnergyPlus.

6.2 Background

In the previous section, we covered most of the reasons for our choice of EnergyPlus as the underlying simulation framework for ROBESim. Due to its flexibility, EnergyPlus has been used in the evaluation of unconventional building systems, e.g., underfloor air distribution systems [123], dual airflow windows [124], thermal chimneys [125], and radiant cooling systems [126]. Standard EnergyPlus benchmark models have also been used to quickly estimate building energy consumption without the need for customized building models [127], highlighting the accuracy of the EnergyPlus simulation framework.

Recent efforts have sought to simplify building energy simulation. Due to its extensible nature, several add-ons have been developed for the EnergyPlus simulation framework [22] to assist users in performing building simulations [128] [129] [130] [131] [132] [133]. NewFacades [129] extends EnergyPlus to help analyze intelligent building facades in the early stages of the building design process. Using NewFacades, different intelligent facade combinations can be quickly generated and evaluated. OpenStudio [130] extends EnergyPlus by providing a visual platform for building model generation and results viewing. Using OpenStudio, the process of building model generation is greatly accelerated. In addition, OpenStudio supports Radiance [134], a simulation tool commonly used to simulate indoor lighting conditions. The modularity of EnergyPlus is highlighted in OpenStudio, since the OpenStudio platform is able to effectively combine two different simulation tools, resulting in greater overall utility.
MLEPlus [131] provides a co-simulation platform between MATLAB [135] and EnergyPlus. This allows experienced MATLAB users to easily perform building energy simulations by invoking EnergyPlus from MATLAB. Such cross-platform tools are very useful as they enable users experienced in a given platform to easily invoke EnergyPlus, reducing the training period required for familiarization with EnergyPlus.

The use of text-based input and output files in EnergyPlus enables a suite of add-ons that provides graphical interfaces for the input and output files. xEsoView [132] provides a graphical user interface for the EnergyPlus standard output files, converting the text files into more readable graphs. These tools facilitate the use of EnergyPlus by simplifying the user interface. ROBESim includes a built-in building model generator for easy building file creation. Our building model generator is further discussed in Section 6.7.

ROBESim can be most closely compared to BEopt [133], a building energy optimization software used to analyze the performance of retrofits and building design choices. BEopt provides a simplified user interface for building modeling and retrofit selection. The user can then simulate across a number of retrofit choices. BEopt provides visual output of the energy savings and implementation cost that can be derived from the respective retrofit choices. Originally developed to consider the construction and material choices during building design (e.g., to compare between double and triple-pane windows), BEopt has recently been updated to consider retrofits. However, the set of retrofits that are hard-coded in BEopt is limited to material and construction retrofits, basic thermostat schedules, and basic equipment efficiency improvements. Although the retrofits included are comprehensive, BEopt does not support novel retrofits. This is primarily due to the focus on building owners and designers, as opposed to retrofit developers. In contrast, we provide application programming interfaces (APIs) to access the basic modules included in ROBESim. Using
these APIs, retrofit developers can quickly develop retrofit modules for their novel retrofits and perform batch simulations in ROBESim. This also allows building owners using ROBESim to simulate across a more complete list of retrofits, such as the Nest learning thermostat [14]. In addition, retrofit modules in ROBESim are not limited to functionality available in EnergyPlus, since the ROBESim co-simulator can perform computations that are not supported under EnergyPlus. This functionality is not available under BEopt. Nonetheless, we believe that the simplicity of the user interface presented in BEopt is unique in the field of building energy simulators, and serves as a model for future simulator development. In ROBESim, we aim to match this simplicity in our user interfaces.

Apart from supporting the automated exploration of retrofit choices, ROBESim also supports fine-grain occupant positioning information (in the form of occupant profiles). Once again, this differentiates ROBESim from BEopt, as occupant profile support is beyond the scope of BEopt. This is important for the analysis of novel, occupant positioning-based retrofits. In ROBESim, we provide built-in support for the following retrofits that depend on occupant profiles: the Nest learning thermostat [14], smart thermostats [16, 17, 30, 31], smart lighting [40, 41, 42], and localized heating. The Nest thermostat, smart thermostat, and smart lighting have been previously discussed in Chapter 2. Localized heating has been discussed in Chapter 4.

These retrofits all rely, in varying degrees, on the occupant profiles supported in ROBESim. For the Nest and smart thermostats, coarse-grain occupancy status is sufficient, whereas smart lighting and localized heating rely on fine-grain occupant positioning information to fully analyze their performance. Although simulation of these retrofits is possible in EnergyPlus (requiring some careful schedule setting), these retrofits are natively supported in ROBESim, and can be easily deployed within the framework.
Previous work has incorporated building occupancy models and occupant preferences in building energy simulation [136, 137, 138]. The methods described in [136] and [137] can be used to generate coarse-grain occupancy information for simulation. In addition to this coarse-grain occupancy information, ROBESim supports fine-grain occupant positioning information. This is important for accurate simulation of retrofits that depend on such information, e.g., smart lighting and localized heating. User preferences can also affect the operation of building systems [138]. Existing simulation tools do not take into account variations in user preferences during simulation. In the simulation of building HVAC, for example, fixed thermostat settings are used, regardless of the occupants that are in the building. Using the occupant profiles in ROBESim, the occupant preferences can be captured and simulated. For example, the thermostat setting can be made dependent on the preferences of the occupants that are in the building or zone. This better reflects real-world scenarios, where occupants frequently adjust the thermostat settings based on their personal preferences [138].

In summary, building energy simulation represents a quick and cost-effective means of estimating building energy consumption. However, existing simulators lack comprehensive support for building retrofits. Since one of the key uses of building energy simulation is to provide before-and-after information on the impact of retrofit choices, this is an important deficiency that we remedy with our ROBESim simulation framework. In addition, we provide native support for occupant profiles, an important component for occupant-aware retrofits.
6.3 Simulator Overview

In this section, we discuss some of the key ROBESim components and their corresponding interactions. We provide an overview of the core simulation framework that ROBESim is based upon.

6.3.1 Simulator Workflow

In order to perform comparisons between different building configurations, several simulation runs have to be performed (one for each building configuration). The workflow for one ROBESim simulation run, for a single building configuration, is illustrated in Figure 6.1. This simulation workflow is managed by the SimulationManager class in ROBESim. The simulation run is broken up into three phases: preprocessing, simulation, and postprocessing. In the preprocessing phase, the Preprocessor takes in the original inputs and produces two different sets of inputs. The first input set is EnergyPlus-compatible, and is used for all computations that are supported under EnergyPlus. The second input set is sent to the ROBESim co-simulator, for additional computations not supported by EnergyPlus. In the simulation phase, EnergyPlus is invoked externally (using the EnergyPlusInvoker class), based on the first set of EnergyPlus-compatible inputs. The CoSimulator is run either simultaneously with or after the completion of EnergyPlus, using the second set of inputs. The outputs from both EnergyPlus and the ROBESim co-simulator are then sent to the postprocessor. The Postprocessor summarizes these outputs in user-friendly terms, such as total energy consumption, end-use breakdown, and energy savings.

In general, depending on the applied retrofit modules, a ROBESim simulation run takes a slightly longer time than a single EnergyPlus simulation. Our analysis (explored further in Section 6.8) indicates a simulation overhead of less than one minute. Of course, the simulation runtime is extended for retrofit modules that involve
more complex computations. Retrofit modules are software modules developed by retrofit developers to be used in ROBESim. In general, these modules reflect the functionality of the underlying retrofit. Retrofit modules are described in greater detail in Section 6.5.

The workflow shown in Figure 6.1 represents a single ROBESim simulation run. Additional functions in ROBESim build upon this workflow. For example, it provides the means to run multiple instances of this simulation across different building models, climates, and retrofit choices; the outputs are then automatically combined and presented in comparison with one another. Another class, StatsGenerator, is used for this comparison step. We discuss these additional usage scenarios in greater detail in Section 6.4.
6.3.2 Inputs

The file-based input/output system of EnergyPlus ensures its modularity and ease of extension. To ensure compatibility and future extensibility, ROBESim follows the same file-based input/output system. In most cases, ROBESim inputs have identical formats as EnergyPlus inputs. For functionality beyond EnergyPlus, ROBESim relies on additional functions that are not supported in EnergyPlus. These functions and their definitions are clearly defined in the ROBESim documentation.

The main inputs of each ROBESim simulation run are as follows:

- **Weather data.** ROBESim accepts EnergyPlus weather files, since they are widely available. A comprehensive list of weather files can be found in [139]. The weather data are exposed to users of the ROBESim libraries through built-in functions.

- **Building model.** The building model comprises two parts: an EnergyPlus-compatible building file and an extended building file used in ROBESim. The extended building file contains information related to the specific appliances in the building, along with their positions. This information can be used to develop smart retrofit modules in ROBESim.

- **Occupant profiles.** For each occupant in the building, there is a corresponding occupant profile containing the positions of the occupant across the simulation period. This information is used in retrofit modules for occupant-aware retrofits.

- **Energy cost and emissions.** This information is primarily used to compute total energy cost and carbon emissions in the postprocessing stage, and varies according to the location of the simulated building. Energy cost and emissions information can be provided as part of the EnergyPlus building model or as a separate input file to ROBESim.
• **Retrofit information.** Retrofits are defined in the retrofits database. The input retrofit information is used primarily to determine which retrofits are to be applied. By changing this retrofit information, multiple ROBESim simulations can be performed without manually modifying the main building file. Note that although the retrofit modules are hard-coded into ROBESim (in the retrofits database), we support the use of input parameters, *e.g.*, different sets of triple-pane window panels may vary in cost and insulation values, but can still use the same retrofit modules with different input parameters, significantly reducing the time needed to define these retrofits for simulation. We describe retrofit modules in greater detail in Section 6.5.

The selected retrofits are automatically applied in the preprocessor, based on the input building model and retrofit information. For example, if the selected retrofit is a triple-pane window panel, the input building model can be the original building with single-pane windows. The preprocessor clones the original building model and applies the selected retrofit, yielding a new building model with triple-pane windows. The parameters for the new triple-pane window are obtained from the respective retrofit information file. In this case, simulation across multiple retrofit combinations is simplified, requiring only a single input building model.

The inputs sent to EnergyPlus are the EnergyPlus-compatible weather and building files, along with a set of schedule files. The schedule files are used in EnergyPlus to simulate the schedules of building systems, *e.g.*, thermostat settings across time for the HVAC system and the lighting schedules for lamps in the building. The ROBESim co-simulator inputs include the extended building file, occupant profiles, the relevant retrofit information, and the same EnergyPlus weather file. Based on the retrofit information, the co-simulator can then determine which computations to perform. This information is specified in the corresponding retrofit module and is further described in Section 6.5.
6.3.3 Outputs

Interim outputs from both EnergyPlus and the ROBESim co-simulator are files in their respective formats. For ease of processing (in the postprocessor), the ROBESim co-simulator output uses the EnergyPlus standard output format (full documentation for the EnergyPlus standard output format is provided on the EnergyPlus website [22]). Thus, the same output file parser can be used to process both sets of outputs.

Listing 6.1 shows a snapshot of the interim output from the ROBESim co-simulator for localized heating (see Chapter 4). Under localized heating, heating energy is split into two parts: the central gas furnace and the electric radiant heaters. The thermostat controlling the gas furnace is set to a minimum temperature of 10°C. The additional heating requirement is covered by radiant heaters directed at the occupants’ positions. This additional energy is simulated in the ROBESim co-simulator, based on the occupant profiles provided. We further detail the localized heating retrofit module in Section 6.5.

Listing 6.1: Co-simulator interim output.

ROBESim 0.1beta, Building-4Z-Light-Res-1;
2,1,TimeStep
   TimeStep
End of Data Dictionary
2,0
8000,146880.0
2,1
8000,146880.0

In Listing 6.1, the first line describes the simulator version, along with the building name. The data dictionary is then defined, using the following format: [Unique identifier], [Number of output parameters], [Description for each output parameter].
This format is identical to the EnergyPlus standard output format. The output data follow this data dictionary. In the example above, for each of the first two timesteps, the localized heating radiant heaters consumed 146,880 Joules (J) of electricity. We describe the computations involved within the co-simulator in Section 6.6.

The ROBESim postprocessor collates the interim outputs into a combined output and derives additional sets of outputs from this combined output. The postprocessor is invoked with a file containing the required processing parameters, primarily involved in the categorization of energy consumption and generation. The default file allows the categorization of the end-uses. By modifying this parameters file, the user can choose to combine the individual HVAC components into a combined HVAC category, for example. This is especially useful for retrofits that impact multiple end-uses at the same time (such as the smart and Nest thermostats). Listing 6.2 shows a sample postprocessor combined output, using modified parameters files, for one timestep. In this case, based on the data dictionary, the following information is captured: energy consumption, energy production, energy derived from natural gas, energy from electricity, heating energy, cooling energy, ventilation energy, lighting energy, and combined HVAC energy consumption. This combined output follows the same format as the co-simulator and EnergyPlus interim outputs.

Listing 6.2: Postprocessor combined output.

ROBESim 0.1 beta, Building-4Z-Light-Res-1-localized_heating\Building
-4Z-Light-Res-1-localized_heating.fso;
2,1,TimeStep
1000,1,Consumption [J] !TimeStep
1001,1,Production [J] !TimeStep
1002,1,Electricity [J] !TimeStep
1003,1,Electricity [J] !TimeStep
1004,1,Heating [J] !TimeStep
1005,1,Cooling [J] !TimeStep
Using this combined output, along with the cost and emissions information from the inputs, the postprocessor computes the desired final output. In ROBESim, this final output comprises the following information: total energy consumption, breakdown of energy consumption by end-use, total cost and emissions information, retrofit-specific comfort metrics, and return-on-investment (ROI) analysis. To incorporate additional metrics, alternative postprocessors can be developed. The output file parser is made available to users of the ROBESim libraries, allowing for quick development and incorporation of alternative postprocessors.

6.3.4 Summary

We have provided an overview of a single ROBESim simulation run in this section, along with the corresponding inputs and outputs. In Section 6.4, we describe the wrapper functions we have developed for different simulator usage scenarios, based on this single simulation run. These wrappers are built into the release version of ROBESim, and represent the core functionality of the simulation tool. In Sec-
tion 6.6, we discuss, in greater detail, the core components of the ROBESim simulation framework, i.e., the SimulationManager, Preprocessor, CoSimulator, and Postprocessor classes. We provide implementation details and discuss the functions contained within each component.

6.4 Simulator Usage Scenarios

In this section, we discuss two usage scenarios for ROBESim: from the building or home owner’s perspective, and from the retrofit developer’s perspective. In addition, we describe the wrapper functions that we have developed for each of these usage scenarios.

6.4.1 Building Owner

As we discussed in Chapters 1 and 2, building owners have several choices when retrofitting their buildings. One of the primary goals of ROBESim is to help these owners in their decision-making process. For these simulations, a single building model and weather file is used to simulate across different retrofit combinations, i.e., one building and climate model to many retrofits. This can be achieved with the addition of a wrapper function.

Figure 6.2 illustrates how this wrapper function works. In this case, the building model, weather data file, occupant profiles, and cost and emissions information remain the same for all simulation runs. The user provides a list of retrofits that are of interest; alternatively, he can use the default retrofits database provided in ROBESim. The wrapper function then applies these retrofits accordingly, invoking the simulation manager for each retrofit combination. When all the retrofit combinations have been explored, the wrapper function includes a summary report comparing the performance
of the retrofit combinations across the output categories, \( e.g. \), the total energy savings derived from each retrofit combination.

By using the wrapper function developed for building owners, the user can perform simulations across multiple retrofit combinations with minimal effort. To further simplify this process, we provide input file generators for the building file and occupant profiles (covered in Section 6.7). Using this suite of tools, the user can quickly and easily obtain an overview of the building’s performance across different retrofit combinations. We provide simulation examples and results in Section 6.8.

### 6.4.2 Retrofit Developer

Whereas the building owner seeks to explore the performance of different retrofit combinations, the retrofit developer needs to evaluate the performance of his retrofit across different building models and climate zones, \( i.e. \), one retrofit to many building models and climate zones. We developed a wrapper function that serves this purpose.

Figure 6.2: Wrapper program for a building owner usage scenario (one building and climate zone to many retrofits).
Figure 6.3: Wrapper program for a retrofit developer usage scenario (one retrofit to many buildings and climate zones).

Figure 6.3 illustrates the functionality of the wrapper developed for retrofit developers. This wrapper function is similar to that for building owners, with the inputs reversed. The retrofit information is sent straight to the simulation manager, whereas the building models and climate zones are retrieved accordingly. The occupant profile is linked to the respective building model, whereas the cost and emissions information is tied to the climate zone. Using this wrapper, a representative set of building models can be simulated across a given set of climate zones. The output from this wrapper highlights the performance of the retrofit in the given building-climate combinations.

To assist in building model and occupant profile generation, the input file generators that we have developed for ROBESim can be plugged into the wrapper function. In this case, the user only needs to input the relevant building characteristics and
occupant profile heuristics, and the wrapper handles the generation and simulation process. We provide simulation examples and results for this wrapper in Section 6.8.

6.4.3 Additional Usage Scenarios

Apart from the two primary usage scenarios above, we have developed ROBESim to be extensible to accommodate additional usage scenarios. Both wrapper functions build upon the single simulation workflow (presented in Section 6.3), with the simulation manager being run multiple times for each scenario. The general model for the wrapper functions is as follows:

- **Input file collection.** Different input file combinations are presented; the wrapper selects appropriate input files for each single simulation run.

- **Multiple simulations.** The simulation manager is invoked multiple times, with different sets of inputs.

- **Output processing.** The final set of outputs is collated and processed. The target information is then extracted from these outputs to provide the final output presented to the user.

This model can be used to develop additional wrappers that consider different input combinations and scenarios. For example, a multi-building, multi-retrofit wrapper can be developed to simulate across a diverse range of conditions. Beyond wrappers, we also provide external libraries for the core ROBESim simulation framework that can be used to incorporate additional usage scenarios, e.g., combining ROBESim with other EnergyPlus add-ons, allowing ROBESim to be extended by simulation software developers for additional functionality.
6.5 Retrofit Description and Management

In this section, we define the retrofit modules used in ROBESim, and highlight the tools and methods available to retrofit developers who develop these retrofit modules. We first outline the basic modules present in ROBESim to provide an overview of the information available to the developer. Second, we present the retrofits database that is included in ROBESim. We then describe how retrofits are processed in the ROBESim preprocessor and co-simulator. To serve as a case study for retrofit developers, we describe the development of the localized heating retrofit module.

6.5.1 Basic Modules

The set of basic modules available in the ROBESim libraries is presented in Figure 6.4 along with some of the information that can be accessed from the modules. Each of these modules is a class within ROBESim, and the information is accessed through built-in methods for these classes.

![Figure 6.4: Basic ROBESim modules.](image)

File parsers are available from the ROBESim libraries for the input weather data, building model, and occupant profiles. Upon running the corresponding parser on
the input file, a `WeatherDataset`, `BuildingModel`, or `OccupantProfile` object is created; these objects allow access to the data contained in the input files through their respective methods. For example, the `WeatherDataset` class comprises an array of `WeatherData` objects. Each `WeatherData` object stores the corresponding weather information for a single timestep. The set of methods used for accessing this information is presented in Listing 6.3. The data stored in the `OccupantProfile` class can be similarly accessed, allowing the retrofit developer to write retrofit modules that depend on occupant preferences and positions.

Listing 6.3: Methods for accessing information in `WeatherData`.

```java
WeatherData:
    public double getTemperature();
    public double getHumidity();
    public double getGlobalHorizontalRadiation();
    public double getDirectNormalRadiation();
    public double getDiffuseHorizontalRadiation();
    public double getIlluminance();
    public double getWindDirectionDeg();
    public double getWindSpeed();
```

The `BuildingModel` class reflects the building models used in EnergyPlus. Most of the core components in EnergyPlus, such as `BuildingSurface`, `Construction`, `Material`, and `FenestrationSurface` have a corresponding ROBESim object. This allows for manipulation of the building model within ROBESim, as well as a built-in means to access the required building model information. For example, to apply a triple-pane window retrofit module using ROBESim, the building model can first be accessed to determine the number and size of windows in the building. The retrofit module can then target the triple-pane window by replacing all the windows (fenestration surfaces) in the building model. We further detail retrofit processing under ROBESim later in this section.
As an added benefit to the mirroring of core EnergyPlus components, a developer can generate a building model within ROBESim, and use the built-in methods to convert this building model into an EnergyPlus-compatible file. This allows for automated building model generation. We discuss our version of this building model generator in Section 6.7.

The final component in the basic modules set involves information related to building systems such as HVAC, lighting, and appliances. EnergyPlus provides full support for HVAC definitions but limited support for both lighting and appliance definitions. In ROBESim, we incorporate most of the commonly used HVAC templates. For lighting and appliances, we include additional information, such as the positions of the appliances, that is not available in EnergyPlus. Since this extended lighting and appliance information is not supported in the EnergyPlus building file, it is exported to and read from a separate file. This file forms the extended building file used in ROBESim simulations (see Section 6.3).

6.5.2 Retrofits Database

Before discussing the process involved in applying retrofits, we first describe the retrofits database that is provided in ROBESim. In addition, we categorize these retrofits and describe how the retrofit categories affect the simulation process. Figure 6.5 illustrates a categorized set of currently available building retrofits, along with the retrofits (in bold) that are included in the retrofits database, i.e., retrofit modules that are available along with the release of ROBESim. This list is non-exhaustive; for more information, a comprehensive survey of currently available building retrofits is provided in [140].

From Figure 6.5 we identify retrofits in the following building categories: building surface, lighting, HVAC, appliances, and renewable energy generation. Building surface involves retrofits that affect the building envelope, typically based on the
Figure 6.5: Categorized set of currently available building retrofits. Retrofits in bold are included in the ROBESim retrofits database.
use of better insulating materials. Lighting, HVAC, and appliance retrofits involve improvements to the respective building systems. Renewable energy generation involves the small-scale generation of electricity through solar panels and wind turbines. ROBESim handles retrofits from different building categories separately. For example, for building surface retrofits, the input handler expects a Material and/or Construction object, along with the required information.

We classify retrofits into the **passive** and **active** types. In ROBESim, passive retrofits are retrofits that do not rely on the additional information in ROBESim, *i.e.*, they are natively supported under EnergyPlus. Conversely, active retrofits rely on additional information, such as occupant profiles and other relevant information (see Figure 6.4). Active and passive retrofits are processed separately in ROBESim (discussed in the next subsection).

Passive retrofits are further classified into replacement and additive retrofits. Replacement retrofits replace existing building systems or constructions; only one replacement retrofit can be applied to each building system or construction, *e.g.*, a triple-pane window retrofit replaces existing windows and cannot be applied along with a double-pane window retrofit. Additive retrofits are additions to existing systems or constructions, and multiple additive retrofits can be applied to the same system or construction, *e.g.*, additional insulation added to the roofs of the building.

Each retrofit module comprises two parts: the code for the module in ROBESim and a parameters file. We use the parameters file to capture variations among similar building retrofits while preserving the core module code, *e.g.*, to support window retrofits of different materials and constructions, only one module needs to be written; the additional material and construction information is read from the parameters file. The parameters file for a triple-pane window retrofit is presented in Listing 6.4.

### Listing 6.4: Parameters file for a triple-pane window.

```
Window, ! Category
```

132
Replacement, ! Type
Triple Glazed Window, ! Name
Window, ! Surface Type
100.0, ! Cost per window
0, ! Additional cost by area
15.0, ! Installation duration per window
0; ! Additional duration by area

! Materials used in current retrofit (EnergyPlus format)
WindowMaterial: Glazing,
  CLEAR 3MM,
  SpectralAverage,
  ,
  0.003,
  0.837,
  0.075,
  0.075,
  0.898,
  0.081,
  0.081,
  0,
  0.84,
  0.84,
  0.9;

WindowMaterial: Gas,
  AIR 6MM,
  Air,
  0.006;

! Construction that defines the retrofit (EnergyPlus format)
Construction,
  Trpl Clr 3mm/6mm Air, ! Name of Construction
In Listing 6.4, the respective window materials and constructions are included in the parameters file. In this way, all replacement window retrofits can share the same code module in ROBESim, e.g., a double-pane window can be simulated by just modifying the Construction object in a new parameters file.

6.5.3 Retrofit Processing

Retrofits are processed in two stages in ROBESim: the preprocessor and the co-simulator. Figure 6.6 summarizes the operations that are performed in each of these two stages. The actual operations are defined in the respective retrofit modules.

![Figure 6.6: Outline of retrofit input processing under ROBESim.](image)

In the preprocessing phase, the preprocessor takes, as input, the building model and retrofit parameters. Based on these parameters, the corresponding retrofit module in ROBESim is invoked to determine the expected implementation cost and dura-
tion. Listing 6.5 shows the required methods in the Retrofit interface in ROBESim. All retrofit modules will implement this interface, ensuring that the same functionality is available across different retrofit modules.

Listing 6.5: Functions in the Retrofit interface.

```java
public interface Retrofit {
    // Retrofit parameters
    public String getName();
    public RetrofitType getType();
    public RetrofitCategory getCategory();

    // Implementation cost and duration
    public double getImplementationCost(BuildingModel building);
    public double getImplementationDuration(BuildingModel building);

    // Apply this retrofit to the given building model
    public void applyRetrofit(BuildingModel building);
}
```

In addition to determining the retrofit implementation cost and duration, the preprocessor generates the retrofitted building model using the applyRetrofit function of the retrofit module on a cloned building model. Listing 6.6 shows the pseudo-code for the applyRetrofit function in a window retrofit module.

Listing 6.6: Pseudo-code for applyRetrofit function in triple-pane window retrofit module.

```java
public void applyRetrofit(BuildingModel building) {
    // Apply construction to all windows in building
    for all windows in building {
        set window construction to construction described in parameters file
    }
}
```
The function retrieves a list of all the windows in the building and sets their construction to match the new window construction. In this example, the retrofitted building model is EnergyPlus-compatible, and there are no additional computations required in the co-simulator.

For active retrofits, such as localized heating, that require co-simulator computations, these computations are performed in the simulation phase. As shown in Figure 6.6 the co-simulator invokes the retrofit module with the relevant information to determine the energy consumption and other metrics. Listing 6.7 shows the required functions in the ActiveRetrofit class.

Listing 6.7: Functions in the ActiveRetrofit class.

```java
global abstract class ActiveRetrofit implements Retrofit {
    // For retrofits with stored local information
    public abstract void init();

    // Get energy consumed or produced, based on occupant profile and weather data
    public abstract double getEnergy(OccupantData occupantData, WeatherData weatherData);

    // For generating reports
    public String getReportKey();

    // Format: [Number of report items], [Name]: [RetrofitCategory]: [FuelType]: [Consumption/Production]: Energy [J]
    public abstract String getReportHeader();
    public abstract String getReport(OccupantData occupantData, WeatherData weatherData);
}
```
In addition to energy consumption information, the ActiveRetrofit class supports easy generation of reports. A separate ReportManager class is updated at each timestep with the reports from all active retrofits. These reports form the co-simulator interim outputs in Listing 6.1 (see Section 6.3).

### 6.5.4 Case Study: The Localized Heating Retrofit Module

As a guided process through retrofit module development, we present a case study on the development of the localized heating retrofit module. Localized heating belongs to the HVAC building category, since it functions as an add-on to central heating systems. Localized heating depends on the occupant positioning information from the occupant profiles, and is considered as an active retrofit. The co-simulator is tasked with determining the additional energy requirement for the electric radiant heaters at each simulation timestep.

In the preprocessing phase, the localized heating retrofit module changes the heating setpoint for all thermostats in the building to the preset minimum temperature, e.g., 10°C. The pseudo-code for this operation is presented in Listing 6.8.

Listing 6.8: Pseudo-code for applyRetrofit function in localized heating retrofit module.

```java
public void applyRetrofit(BuildingModel building) {
    // Change heating setpoint for all thermostats in the building
    for all thermostats in building {
        set heating setpoint to minimum temperature (10 degrees celsius)
    }
}
```

In addition, the pseudo-code used to determine the additional radiant heating energy, in the co-simulator, is shown in Listing 6.9. The retrofit module first determines which occupants are in the building, along with their respective positions. It
then computes the required radiant heating energy for each occupant, based on his preferences. Finally, the total energy consumption is returned to the co-simulator for logging purposes.

Listing 6.9: Pseudo-code for `getEnergy` function in localized heating retrofit module.

```java
public double getEnergy(OccupantData occupantData, WeatherData weatherData) {
    // Determine total energy consumption by radiant heaters
    for all occupants {
        if occupant is in the building {
            compute required radiant heating energy
        }
    }
    return sum of computed radiant heating energy
}
```

From this example, we observe that, by exposing the building model components to the user, ROBESim allows automated retrofit application. In addition, retrofits that are not normally compatible with EnergyPlus can still be used under ROBESim. In the localized heating example, EnergyPlus is still used to compute the baseline heating energy requirement. The co-simulator computes the part of the retrofit module that is not supported in EnergyPlus.

In this case, the minimum temperature and radiant heating energy computation for localized heating are variables input to the retrofit module during initialization. Thus, to consider the performance of localized heating using a different minimum temperature (e.g., 15°C), the same localized heating retrofit module can be initialized with the required minimum temperature. In general, retrofit modules are developed to accept variables during initialization. In this way, only one retrofit module needs to be developed to consider variations to the retrofit. As another example, only one wall
retrofit module was developed to consider different wall thicknesses in our parametric analysis in Section 6.8.2.

6.5.5 Summary

In this section, we discussed the information that the retrofit developer can use when developing retrofit modules in ROBESim. We also described some of the required methods for these modules. The localized heating retrofit module serves as a case study for the future development of retrofit modules.

6.6 ROBESim Components

In Section 6.3, we provided an outline of the ROBESim simulator workflow. In this section, we discuss, in greater detail, the components of the ROBESim core simulation framework: the simulation manager, preprocessor, co-simulator, and postprocessor.

6.6.1 Simulation Manager

The SimulationManager class is used to manage the single-run simulation workflow shown in Figure 6.1. It is initialized with a building file, corresponding set of occupant profiles and weather data, energy cost and emissions, and the retrofits that need to be applied, along with their parameters. As previously discussed, the application of the retrofit is performed automatically in ROBESim; a retrofitted building model is generated in the process. In this way, the retrofit developer has control over the retrofit application process, and the user does not have to concern himself with manual retrofit application.

The tasks performed in the simulation manager are as follows:

1. Invoke preprocessor with given inputs.
2. Invoke EnergyPlus with compatible building and weather files.

3. Invoke co-simulator with extended building file, weather file, and occupant profiles.

4. Upon completion of EnergyPlus and co-simulator simulations, invoke postprocessor with interim simulator outputs.

5. Terminate.

The simulation manager serves as a base that additional usage scenarios build upon, as previously discussed in Section 6.4.

6.6.2 Preprocessor

The preprocessor first performs a preliminary check of the inputs. This is to ensure the existence of input files and their compliance with the requirements of the simulator. Next, the selected retrofits are applied to a clone of the base building model to yield a retrofitted building model. Finally, the preprocessor reads and parses the input files into objects that can be accessed within the ROBESim framework. The methods used to access these objects are presented in Listing 6.10.

Listing 6.10: Methods for accessing information in Preprocessor.

```java
WeatherData:
    public BuildingModel getBuilding();
    public WeatherDataset getWeatherDatabase();
    public OccupantProfile getOccupantProfile();
    public RetrofitDatabase getRetrofitDatabase();
    // Energy cost and emissions information
    public ExtendedInputs getExtendedInputs();
```

In Listing 6.10, the energy cost and emissions information is stored in an `ExtendedInputs` object. This class can be expanded to contain additional informa-
tion, as required. The preprocessor essentially serves as an input file processor and storage unit for the input information.

6.6.3 Co-Simulator

The co-simulator is used to compute additional energy information not supported in EnergyPlus. This primarily affects active retrofits; since active retrofits rely on additional information, they typically require some computation in the co-simulator. However, not all active retrofits require co-simulator computations. For example, the smart thermostat module uses the occupant profile information to determine the correct thermostat schedules for simulation. This computation is performed in the preprocessing phase, when the \texttt{applyRetrofit} method is called. Once the schedules are determined, they can be used in the building model sent to EnergyPlus, since EnergyPlus provides native support for thermostat schedules.

For other active retrofits, such as localized heating and smart lighting, computations are required in the co-simulator. For each timestep, the co-simulator invokes the \texttt{getEnergy} method with the weather and occupant data for that specific timestep. The additional energy computations are then performed as defined in the respective retrofit modules. The co-simulator then invokes a secondary class, \texttt{ReportManager}, to collate and store the output reports for this phase. After all computations are complete, the report manager writes the interim outputs to an interim output file (see Listing 6.1 in Section 6.3).

6.6.4 Postprocessor

The postprocessor reads the interim outputs from both EnergyPlus and the ROBESim co-simulator to produce a combined output file. This is based on an input parameters file that is used to sort the relevant outputs into different categories.
Listing 6.11: Snapshot of input parameters file used in Postprocessor.

Electricity,
   Cooling:Electricity,
   Fans:Electricity,
   InteriorLights:Electricity,
   Electricity:Consumption;

Lighting,
   InteriorLights:Electricity;

HVAC,
   Heating,
   Cooling,
   Fans:Electricity;

Listing 6.11 shows a portion of the input parameters file, used to classify electricity, lighting, and HVAC energy consumption. The postprocessor cross-references the data dictionaries of the interim outputs with the input categories defined in the parameters file to determine the correct categorization. In this example, the HVAC category covers dictionary items with the following terms: Heating, Cooling, and Fans:Electricity. Note that dictionary items can belong to several different categories, e.g., the Fans:Electricity dictionary item also belongs to the main Electricity category. The result of this classification step is a combined output file (see Listing 6.2 in Section 6.3) that tracks the energy transfer in the respective categories.

We have presented some key information on the core ROBESim components. Apart from these components, there are a number of helper modules that we have developed to simplify retrofit development and extension of our simulation tool, e.g., input parsers, output tools, and data storage classes. Additional information on these helper modules can be found in the API documentation provided for ROBESim.
6.7 Input File Generators

One of the key goals for our design of ROBESim was accessibility for users of all experience levels. With this in mind, we have developed a number of ease-of-use modules to help improve user experience. These modules primarily assist with building model and occupant profile generation.

6.7.1 Building Model Generator

We have developed an automated building model generator capable of quickly generating multiple building models offline. This is in contrast to the EnergyPlus example file generator [14] available online. Using the online tool, users generate building models one at a time; the generated files are then sent to the user through email, usually after a period of time (from approximately fifteen minutes to up to an hour). Conversely, the building model generator in ROBESim can be generated directly on the users’ machines; a typical building model can be generated in under one minute (the output building model file is fully compatible with EnergyPlus v8.0). Our building model generator contains several default building system choices, especially for users that are not aware of the systems installed in their building, e.g., the default choice for HVAC system is the HVACTemplate:Zone:Unitary template defined in EnergyPlus, referring to the unitary HVAC systems most commonly used in residential and small commercial buildings.

An overview of the different stages in the building model generator is presented in Figure 6.7. The key inputs are as follows: building dimensions, window fraction (proportion of walls covered by windows), materials, and climate zone. For building materials, we use the preset constructions provided in EnergyPlus, i.e., Light for wooden buildings, Medium for brick buildings, and Heavy for concrete buildings. We find that these preset constructions provide a good approximation for most buildings.
of their type. The use of preset constructions presents a more convenient abstraction for the average user.

Based on the inputs above, the building model generator generates the building envelope. This envelope is then divided into different zones. We approximate this process by defaulting to a four-zone per floor scheme. The next stage involves generating the necessary HVAC templates for the building model. The generator supports two templates: HVACTemplate:Zone:Unitary and HVACTemplate:Zone:IdealLoadsAirSystem. The latter represents an ideal loads air system, serving as a high-level abstraction for the actual HVAC system in the building. The output from the generator comprises the EnergyPlus-compatible building model, and the extended building model containing additional information about the appliances in the building.
We provide the source code for our building model generator in ROBESim. It can be extended to include other additional considerations that are not reflected in our generator.

### 6.7.2 Occupant Profile Generator

The occupant profile generator is used to automatically generate occupant profiles for a given building. Figure 6.8 provides an overview of the occupant profile generator provided in ROBESim.

![Occupant profile generator outline](image)

Figure 6.8: Occupant profile generator outline.

We have developed two versions of occupant profile generators: a random profile generator and a schedule-based generator. The inputs to the generator are as follows: the building model (or file), the number of occupants to generate, heuristics (for the random profile generator), and schedules (for the schedule-based generator).

The random profile generator is the easiest way to generate a working occupant profile. It is based on the following heuristics:

- *Hours unoccupied.* The time interval when the occupant is out of the building.
• **Movement probability.** Occupants in a building generally do not move around too much. This probability approximates the movements in real-world situations.

• **Sleeping hours.** For residential buildings, this represents the time interval when the occupant is sleeping and, thus, not moving.

In general, we find that the random profile generator works well, especially considering that it requires no additional effort on the part of the user. Thus, we expect it to be more commonly used.

For users that require a more deterministic occupant profile, we provide a schedule-based profile generator. Since it is not reasonable to expect the user to input a full year of schedules, we require a week of schedules in our generator. These schedule data are then replicated to cover a full calendar year. Sample weekday schedule data are presented in Listing 6.12.

Listing 6.12: Sample weekday schedule data.

```
DayDay, !Day of the week (1=Monday) 1 2 3 4 5, !Start,End,Activity,Location (Out=not in building)
0000,0900,Sleep,Bed,
0900,0930,Shower,Bath,
0930,1800,Work,Out,
1800,1900,Meal,Dining,
1900,2000,TV,Sofa,
2000,2130,Computer,Desk,
2130,2200,Shower,Bath,
2200,2400,Sleep,Bed;

!Map locations to points
LocationMap,
```

146
The schedule data include the following information: occupant schedule for all seven days of the week and map of given locations to actual coordinates. Based on the given schedule, the corresponding occupant profile is then generated.

As in the case of the building model generator, we provide source code for both versions of our occupant profile generator. Our examples serve as reference for future versions of occupant profile generators.

6.8 Simulation Examples

We have discussed the core ROBESim components, along with some of the ease-of-use extensions that we have developed. In this section, we present some simulation examples for ROBESim.

6.8.1 Building Owner Usage Scenario

In Section 6.4.1 we covered the one building to many retrofits usage scenario that most building owners, and users of ROBESim, will follow. We performed one simulation run using the wrapper program for the building owner usage scenario. The input file to the wrapper program is presented in Listing 6.13.

Listing 6.13: Input to wrapper program for building owner usage scenario.

```
Building,
  Example-4Z-108-1-Res-Light;
```
Weather,
   USA_NJ_Trenton-Mercer.County.AP.724095_TMY3.epw;

Occulant,
   occ_example_basic;

Retrofit,
   localized_heating,
   programmable_thermostat,
   smart_thermostat,
   triple Pane_windows;

The input file is parsed in the wrapper program to determine the input building file, weather file, occupant profiles, and the retrofits to consider. The input building file is a single-story 108 m² residential apartment with light construction, i.e., built of wood. The input weather file, in this case, is for Trenton, New Jersey. The occupant profiles used are for three regular workers, i.e., they leave for work on weekdays. These profiles were generated using the schedule-based generator described previously. In this example, we consider the following retrofits: localized heating, programmable thermostat, smart thermostat, and triple-pane windows. We seek to determine the impact on HVAC energy consumption of implementing each of these three retrofits.

Figure 6.9 illustrates the simulation results. The energy savings results for the smart thermostat and for localized heating are in line with the results presented in [16, 17, 30, 31] and Chapter 4, respectively. The savings results for the programmable thermostat and triple-pane windows are in line with our manual simulation results; in our validation test, we created retrofitted building models manually for both retrofits to obtain these simulation results.
6.8.2 Parametric Analysis Example

In addition to performing comparisons between different retrofits, ROBESim can also be used to perform parametric analysis on a specified retrofit. In this case, the base retrofit remains the same; variants of this base retrofit are generated from user-specified parameters. These retrofit variants are then automatically applied to the building model for simulation.

As an example of parametric analysis using ROBESim, we evaluated the impact of wall insulation thickness on HVAC energy consumption. In this analysis, we used the same building model and climate zone (single-story 108 m² residential apartment with light construction in Trenton, New Jersey). In our simulations, we used the following wall insulation thicknesses: 50 mm, 75 mm, 100 mm, 125 mm, 150 mm. The wall insulation thickness in the original building model was 50 millimeters (mm). This was used as our base case for energy savings comparisons. The results of this analysis are illustrated in Figure 6.10.
Figure 6.10: HVAC energy consumption and savings across different wall insulation thicknesses.

From Figure 6.10, we observe that the energy savings increase as additional insulation is added to the walls. A wall insulation thickness of 150 mm enables HVAC energy savings of around 10%, compared to the base case (50 mm). In this example, ROBESim automatically applies the new wall constructions to the base building model to create new building models with the appropriate wall insulation thicknesses. In addition, the output from ROBESim allows quick and easy comparison between the different parameters. Thus, the effort required from the user is significantly reduced.

### 6.8.3 Retrofit Developer Usage Scenario

In this example, we explore the retrofit developer’s perspective: a one retrofit to many buildings usage scenario. For this example, the wrapper program input is as shown in Listing 6.14.

Listing 6.14: Input to wrapper program for retrofit developer usage scenario.

```
Building,
Example-4Z-108-1-Res-Light;
```
The input instructs the wrapper program to perform several ROBESim simulations with the triple-pane window retrofit, using the same building model, across different climate zones. The climate zones selected are as follows: Trenton (New Jersey), Chicago (Illinois), Boulder (Colorado), Phoenix (Arizona), Seattle (Washington), and Atlanta (Georgia). The simulation results are illustrated in Figure 6.11. From the figure, we observe that the HVAC energy savings, as a percentage, are approximately the same for the six climate zones (around 20%), although the energy savings, when represented in gigajoules (GJ), vary significantly. This is primarily due to the varying HVAC requirement across the different climate zones. Thus, although the proportion of HVAC energy savings is similar in Chicago and Phoenix (22% vs. 19%), the total energy and cost savings will be significantly higher in Chicago than in Phoenix (23.2 GJ vs. 8.6 GJ).

In the two examples above, we described the structure of the wrapper program inputs. Upon receiving these inputs, the rest of the wrapper programs is automated, i.e., the programs do not require any user intervention. The output from the simulations is a comma-separated values (CSV) file that can be easily processed to produce
Figure 6.11: HVAC energy savings for triple-pane window retrofit across different climate zones.

The figures shown in Figures 6.9 and 6.11. Thus, using the wrapper programs included in ROBESim, the user can quickly and easily perform simulations for the different usage scenarios.

6.8.4 Simulation Overhead Analysis

In this section, we discuss the simulation overhead attributed to ROBESim. In our analysis, we performed ten identical ROBESim simulation runs, using the 108 m² apartment building model in Trenton, New Jersey. We used localized heating as an example, since localized heating requires additional computations in the co-simulator to determine the energy required in radiant heating (see Section 6.5.4). Figure 6.12 illustrates the results of this analysis. From the figure, we observe that EnergyPlus represents the bulk of the simulation run time in our example. On average, the additional simulation overhead from ROBESim (preprocessing, co-simulation, and postprocessing) is around 14.5%, compared to running EnergyPlus alone.
In our example, the computations in the co-simulator, using the localized heating retrofit module, do not require much additional time (approximately 0.4 seconds). This is not true for all retrofit modules, since some retrofit modules depend on control systems using complex computations (e.g., smart lighting requires more co-simulation time due to its dependency on genetic algorithms for control), whereas other retrofit modules, such as simple wall or window retrofits, do not require any computations in the co-simulator. Thus, the actual simulation overhead depends on the retrofit modules that are applied.

6.9 Chapter Summary

In this chapter, we have motivated the need for more information when making decisions to retrofit buildings. In addition, we have highlighted building energy simulation as a quick and inexpensive means to obtain this information. We anticipate that, as more retrofits are developed and made commercially available, retrofitting old build-
ings will be an increasingly popular solution to reduce energy consumption. Existing simulation tools, however, are too complex for most building owners to operate. Our solution, ROBESim, is a retrofit-oriented simulation tool that supports built-in functions for retrofit modules. This enables a host of user-friendly add-ons that simplify the building energy simulation process. We are confident that ROBESim will be easy to use and useful to both building owners and retrofit developers of all experience levels.

In ROBESim, we have made a deliberate decision, in our design process, for our simulation framework to be modular and extensible. We have also made the relevant libraries, source code, and API available. We hope that ROBESim can serve as a base for additional modules and interface enhancements to improve the building energy simulation experience.
Chapter 7

Conclusions and Future Research

Energy expenditures represent a significant proportion of the global gross domestic product (GDP). Globally, the rapid pace of development, especially in countries such as Brazil, Russia, India, and China, presages further increases in energy expenditures, leading to considerable socioeconomic and environmental impact. This motivates the need for energy conservation measures to reduce the global energy footprint.

In Chapter 1 of the dissertation, we introduced buildings as a prime target for energy conservation measures, since buildings represent a large proportion of global energy consumption. We motivated the need for building retrofit solutions, in addition to energy-efficient building design and architecture, to enable immediate energy savings. We presented the different energy end-uses splits in commercial and residential buildings in the United States, and highlighted the end-uses that can have the most significant impact. We provided an overview of the different approaches for the analysis and optimization of building energy consumption.

In Chapter 2 we presented some related work in the field of building energy efficiency technologies and building energy analysis. We classified the different building energy efficiency approaches into their target end-uses: heating, ventilation, and air conditioning (HVAC), lighting, water heating, and computing. We also discussed
some of the challenges involved in designing novel energy-efficient building technologies. Building energy analysis allows the building owner to make informed decisions about which energy-efficient building technologies (or retrofits) to implement. We discussed three important tools in building energy analysis: building metering, building energy audits, and building energy simulation. We highlighted building energy simulation as a cost-effective solution toward more informed building retrofit choices.

In Chapter 3, we discussed building-level renewable energy generation as a solution to help reduce energy expenditures. We described the energy generation characteristics for both solar panels and wind turbines. We also provided the methodology that building owners can follow to determine the solar and wind energy potential for their building. We evaluated selected renewable energy technologies (one solar panel and one wind turbine), to determine their feasibility from an investment perspective. We discussed the issues related to building-level renewable energy generation. The results of our evaluation indicate that small-scale solar energy generation is a feasible solution, primarily due to improved technology and reduced costs.

In Chapter 4 we introduced localized heating, a novel building heating system. Under localized heating, radiant heaters are used to heat individual occupants, with the central heating system set to a much lower temperature setting (e.g., 10°C). This results in a significantly reduced heating space and, consequently, considerable energy savings. We outlined the proposed localized heating system, providing details of control algorithms for the system components. We also discussed thermal comfort related to radiant heating. We presented an occupant positioning system that can be used to locate building occupants and described our prototype implementation of such a system. We provided simulation results to estimate the heating energy savings enabled by localized heating. Our results indicate that localized heating is a promising approach, resulting in significant energy and cost savings.
In Chapter 5, we introduced an occupant-level sensing (OLS) system, capable of providing information on the exact environmental conditions that each building occupant is experiencing. We outlined the components in the OLS system and provided details of their function. We discussed some applications enabled by OLS: improved localized heating, occupant-level indoor air quality management, and occupant-level smart lighting. We described our prototype implementation of the OLS system and presented some of the data collected from a real-world deployment of the prototype system. In addition, we evaluated the energy savings potential of occupant-level smart lighting. Our results indicate that OLS can help improve building systems by providing guarantees toward occupant comfort, as well as allowing these systems to be set to an output level that is just right.

In Chapter 6, we presented a retrofit-oriented building energy simulator, ROBESim, that we developed as part of this dissertation. ROBESim was developed to natively support building retrofits. Retrofit modules can be automatically applied to building models, removing the need to manually update building models to simulate different retrofits. In addition, ROBESim supports occupant-aware building applications, such as localized heating, through the use of occupant profiles, i.e., occupant preferences and location information. We discussed the ROBESim simulation workflow and described some built-in wrapper programs for two different usage scenarios: building owners and retrofit developers. We described the retrofit module development process and provided an overview on the modules that are available to retrofit developers. We then described some ease-of-use enhancements that we have developed to facilitate the simulation process for novice users: a building model generator for quick building model creation and an occupant profile generator to quickly generate occupant profiles based on specified heuristics. We then provided some simulation examples and analysis to showcase the features of ROBESim.
We now discuss some new research directions that build upon the work presented in this dissertation.

**Occupant-aware building systems:** Existing building systems are largely occupant-unaware, leading to energy wastage during operation. The occupant positioning and occupant-level sensing systems presented in this dissertation can be utilized in building applications beyond localized heating. We envision two different categories of occupant-aware building solutions:

- **Energy-efficient building applications:** Building systems can be made occupant-aware to help improve energy efficiency. We have described how building heating can be improved under localized heating. Further research can be performed to formulate a comparable solution in building air conditioning. Building lighting is also a strong candidate for occupant awareness. Existing solutions in these areas focus on systems found primarily in commercial buildings. A more general solution that can be applied to all buildings will be useful.

- **Other occupant-aware building systems:** Occupant awareness goes beyond energy efficiency applications. Future work can be done to utilize occupant positioning in fall detection for the elderly. Existing solutions depend on user-carried devices that are often misplaced. Using the tagless occupant positioning system that we have described, algorithms can be formulated to determine anomalies and report them to the respective caregivers. In addition, occupant positioning can be used in building security purposes, where the building owners can be alerted to the presence of intruders in their buildings.

**Building system interoperability:** Modern buildings depend on several heterogeneous building systems to provide comfort to building occupants. These systems currently operate independently of one another, e.g., building lighting systems do not communicate with building HVAC systems during operation. However, these sys-
tems are all reliant on the occupant for feedback. In building lighting, occupants flip switches to control lighting systems; building HVAC relies on occupant feedback through the thermostat. To improve building energy efficiency, a solution worth exploring is to link these controls such that the occupant need only convey his current state or action (e.g., “going out,” “going to bed,” and “studying”) and the building systems will respond accordingly. When the occupant indicates the “going out” state, for example, the building HVAC and lighting systems should be automatically shut off or set to a lower output level. In this way, building systems can be elevated from simple on/off systems to intelligent, cooperative systems.

**Ease-of-use extensions to building energy simulation:** Building energy simulators are valuable tools in building energy analysis. Unfortunately, these simulation tools were not designed with ease-of-use in mind. As a result, a large majority of building owners is unable to take advantage of these tools. An important next step in the field of building energy simulation is to simplify these tools. The target areas for simplification are the building modeling phase and the results analysis phase. Building modeling is the most demanding phase in building energy simulation. Users often have to grapple with complex modeling tools. Although previous efforts have sought to simplify building modeling, it still remains as the primary barrier to entry for building energy simulation. The results output from the simulation tools should also be presented in easy to understand terms for the novice user. Future research can focus on these two aspects of building energy simulation to enable a smoother experience for all users.
Bibliography


160


