ESE 534 – CYBER PHYSICAL SYSTEMS COURSE PROJECT
Smart Building – Control Of HVAC Systems Based On Occupancy Using OpenBuild

Shashank Rao, Kewal Raul and Pranith Kumar
Stony Brook University

Abstract— **This report elaborates on the work done for course project of ESE 534 - Cyber Physical Systems. We have incorporated occupancy schedule into OpenBuild toolkit of MATLAB and run the simulations for HVAC and lighting systems for the building using EnergyPlus.**

I. INTRODUCTION

A growing awareness of the planet's limited natural resources has heightened the demand for renewable energy, sustainable infrastructures, and waste reduction. As per recent studies, the commercial and residential building sector is one of the largest consumers of energy, which has led to massive research and advances being made to improve efficiency of buildings as well as to minimize wastage of energy. Making buildings more energy efficient is a major step to reduce our energy consumption and carbon footprint to combat global climate change. One of the major steps in this direction is the introduction and widespread adoption of automated building management systems or 'smart buildings'.

A smart building is any structure that uses automated processes to automatically control the building's operations including heating, ventilation, air conditioning, security, fire and flood safety, lighting (especially emergency lighting), humidity control and other systems. A smart building uses sensors, actuators and microchips to collect data and manage itself according to its requirements. The objectives of building automation are improved occupant comfort, efficient operation of building systems, reduction in energy consumption and operating costs, and improved life cycle of utilities.

A smart building uses many different sensors – temperature, humidity and pressure sensors (such as thermistor, platinum resistance thermometer or wireless sensors) to make analog/variable measurements as well as DC/AC signal switches, current switch, air flow switch or a volta-free relay contact to make digital measurements. Analog outputs control the speed or position of a device (an example is a hot water valve opening up a specific required value to maintain a setpoint) while digital outputs are used to open and close relays and switches as well as drive a load upon command (an example would be to open a valve by allowing 24V DC/AC to pass through the output powering the valve).

The main infrastructure of a smart building depends upon a controller, the lighting, the air handlers, a central plant, alarms and security, as well as the occupancy of the building. Controllers are essentially small, purpose-built computers with input and output capabilities. These controllers come in a range of sizes and capabilities to control devices commonly found in buildings, and to control sub-networks of controllers. Lighting can be turned on, off, or dimmed with a building automation or lighting control system based on time of day, or on occupancy sensor, photo sensors and timers. The air handlers mix and return outside air so less temperature/humidity conditioning is needed. A central plant is needed to supply the air-handling units with water. It may supply a chilled water system, hot water system and a condenser water system, as well as transformers and auxiliary power unit for emergency power. Alarms and security detects a potentially hazardous situation if no one who can solve the problem is notified. Notification can be through a computer (email or text message), cellular phone voice call, audible alarm, or all of these. Occupancy is usually based on time of day schedules. Depending on the occupancy, the smart building aims to provides a comfortable climate and adequate lighting, often with zone-based control so that users on one side of a building have a different thermostat (or a different system, or sub system) than users on the opposite side. A temperature sensor in the zone provides feedback to the controller, so it can deliver heating or cooling as needed.

II. OVERVIEW OF THE PROJECT

In our project, we have upgraded an existing software system perspective to improve energy efficiency in buildings. We have incorporated occupancy data along with OpenBuild. Once the number of occupants in the zones and building had been detected, the electrical systems in the building were monitored and controlled as per requirement. We currently controlled mainly two high energy consuming systems – HVAC and lighting systems - in the building, with a scope to increase to other systems if required.

EnergyPlus is a whole building energy simulation engine that can be used to model both energy consumption – for heating, cooling, ventilation lighting, plug and process loads – as well as water consumption in buildings. It provides an integrated solution of thermal zone conditions and HVAC system response within the building. A user-defined time step can be provided in order to simulate the interaction between thermal zones and the environment along with automatically varied time steps for interactions between thermal zones and HVAC systems. Combined heat and mass transfer models, advanced fenestration models, illuminance and glare calculations, component-based HVAC, a number of built-in HVAC and lighting control strategies, functional mock-up interface, standard summary and detailed output reports are all distinctive features of EnergyPlus. We have used EnergyPlus in our course project to simulate the functioning of the building, specifically, modelling the building and simulating functioning of HVAC and lighting by incorporating occupancy data.

OpenBuild is a MATLAB toolbox for advanced controller design for buildings heating ventilation and air conditioning systems, with an emphasis on Model Predictive Control (MPC) methods. It provides researchers in the control community the capability of testing control algorithms on a wide range of realistic simulation scenarios, by providing most of the data needed to easily perform simulation and optimization. It combines the convenience of controller design in MATLAB with the processing power and simulation capabilities of the building simulation software EnergyPlus. It includes a building modeling tool to construct linear state-space models of building thermodynamics based on building description data, making it particularly useful for design of optimal controllers requiring a good prediction model of the building, as well as providing the input data necessary for simulation such as weather, occupancy and internal gains data. The ability to co-simulate the building between MATLAB and EnergyPlus enables fast prototyping and validation of the models and controllers. We have modified the working of OpenBuild in our course project to include people occupancy as a parameter to be sent to EnergyPlus as well as extract the same data for graphical results.

III. EXPERIMENTAL SETUP

A. INITIAL SETUP

1) Download and install MATLAB from its official site. This will require a license for download and install.
2) Download and install Java JDK.
3) Download and install Python libraries.
4) Download and install EnergyPlus v8.5 (Do not use versions higher than this since OpenBuild parameter processing will fail for higher versions)
5) Download and extract OpenBuild toolkit into a work directory. Set the requisite paths in "installOpenBuild.m" file and run it in MATLAB. This will install the OpenBuild toolkit into MATLAB.
6) Download and install the Gurobi Optimizer; this is required by OpenBuild to perform HVAC processing. This will also require a license for download and install.
7) Run the "tutorial.m" file; the processing occurs and the output is displayed.

B. FLOW OF THE SIMULATION

The main input file for OpenBuild is the idf description file which contains the building model parameters as well as the scheduling periods and values during these scheduled periods. The idf file contains different classes which are used by EnergyPlus to run its simulations. The main classes of interest to us in the idf file are:

1) Timestep - Specifies the "basic" timestep for the simulation. The value entered here is also known as the Zone Timestep. This is used in the Zone Heat Balance Model calculation as the driving timestep for heat transfer and load calculations.
2) Site:Location - Specifies the building's location. Only one location is allowed. Weather data file location, if it exists, will override this object.
3) SizingPeriod:DesignDay - The design day object creates the parameters for the program to create the 24 hour weather profile that can be used for sizing as well as running to test the other simulation parameters. Parameters in this include a date (month and day), a day type (which uses the appropriate schedules for either sizing or simple tests), min/max temperatures, wind speeds, and solar radiation values.
4) RunPeriod - Specified a range of dates and other parameters for a weather file simulation. Multiple run periods may be input, but they may not overlap.
5) ScheduleTypeLimits - ScheduleTypeLimits specifies the data types and limits for the values contained in schedules.
6) Schedule:Compact - This is the schedule type for an irregular object. Fields which can be included are :- "Through", "For", "Until" followed by the value to be defined until "Until" time.
7) Zone - Defines a thermal zone of the building.
8) People - Sets internal gains and contaminant rates for occupants in the zone. If you use a ZoneList in the Zone or ZoneList name field then this definition applies to all the zones in the ZoneList.
9) Lights - Sets internal gains for lights in the zone. If you use a ZoneList in the Zone or ZoneList name field then this definition applies to all the zones in the ZoneList.
10) ElectricEquipment - Sets internal gains for electric equipment in the zone. If you use a ZoneList in the Zone or ZoneList name field then this definition applies to all the zones in the ZoneList.
11) ExternalInterface - Activates the external interface of EnergyPlus. If the object ExternalInterface is present, then all ExtnernalInterface:* objects will receive their values from the BCVTB interface or from FMUs at each zone time step. If this object is not present, then the values of these objects will be fixed at the value declared in the "initial value" field of the corresponding object, and a warning will be written to the EnergyPlus error file.
12) ExternalInterface:Schedule - Schedule contains only one value, which is used during the warm-up period and the system sizing.
13) ExternalInterface:Variable - At the beginning of each zone time step, its value is set to the value received from the external interface. During the warm-up period and the system sizing, its value is set to the value

specified by the field "initial value." This object can be used to move data into the EnergyPlus subroutines.

14) Output:VariableDictionary - Produces a list summarizing the output variables and meters that are available for reporting for the model being simulated (output file). The list varies depending on the types of objects present in the idf file.
15) Output:Table:SummaryReports - This object allows the user to call report types that are predefined and will appear with the other tabular reports. The entries for this object is a list of the predefined reports that should appear in the tabular report output file.

Other than these fields, there are still many other fields which define the building structure, materials used, wind and sunlight effects, etc. which are present in the idf file; we have not changed or updated these classes and have used a copy of the same idf file used by the tutorial(OfficeBuilding_V850.idf) after modifying only the above required classes.

The following are the values used by us for the important objects in EnergyPlus:

1) Timestep Object - Value used is 60; this means that iterations will be performed by EnergyPlus every one minute, leading to a total of 10080 ((60 min)*(24 hrs)*(7 days)) iteartions.
2) SiteLocation Object - We have used "Islip Long Isl Macarthur Ap_NY_USA Design_Conditions" weather file, which is the weather file for Long Island, New York; the other parameters for this object are based on this Long Island file parameters.
3) RunPeriod Object - We have run the simulation from 2018-1-15 to 2018-1-21 (1 week) starting from Monday to the next Sunday.
4) Zone Object - We have used three zones - ZN_1 (co-ordinates(0,0,0)), ZN_2 (co-ordinates(15,0,0)) & ZN_3 (co-ordinates(25,0,0)) each with a ceiling height of 3.8m. The floor area and volume are set to auto-calculate.
5) ExternalInterface Object - This object is present as commented in the initial idf description and will be read during parsing in OpenBuild to create the intermediate idf description file. We have set this value as PtolemyServer.
6) ExternalInterface:Schedule - This object is present as commented in the initial idf description and will be read during parsing in OpenBuild to create the intermediate idf description file. We have added HVAC and People object's scheduling declaration in this object.
7) ExternalInterface:Variable - This object is present as commented in the initial idf description and will be read during parsing in OpenBuild to create the intermediate idf description file. We have added HVAC and People object's scheduling definition in this object.

The following steps elaborate on the flow of the entire simulation:

1) The simulation begins at the MATLAB file "project.m"; a building object as well as idfFile and epwFileobjects are created initially.
2) The first function called is loadData() which attaches the idf description file to the building object created at the beginning of the function.
3) The next function called is collectData() which reads the idf description file and collects all data present in that file.
4) The next function called is createModel() which creates the model of the building in MATLAB based on the building data previously collected from the idf description file.
5) The functions called next are addCustomInputs(); the function adds the external interface objects (HVAC objects) to the created model.
6) After this getStateSpaceModel(), getSimulationInputs() & createIDFFile() functions are called; the first function gets the continuous-time state space model of the entire building including the added custom inputs; the second function gathers inputs to the linear model from last simulation; and the third function creates another intermediate idf simulation file which includes the external interface objects added from custom inputs from the previous steps.
7) Next, the simulationEngine(), energyPlusSimulator(), and addSimulatorEnergyPlus() functions are called; the simulationEngine() class supervises co-simulation in MATLAB, it includes all the components of the control structure (controller, estimator, building simulator) and takes care of the data passing between them and it runs the main simulation loop; energyPlusSimulator() is the initialization for a simulator of a building in Energy+, it takes an idf file, an epw file and the outputTableData necessary for the cosimulation between MATLAB and EnergyPlus; addSimulatorEnergyPlus() function attaches an EnergyPlus simulator object to the simulation engine.
8) After this, as EnergyPlus simulation runs in parallel with the MATLAB simulation, HVAC processing (of the values from EnergyPlus) is done in the rest of the "project.m" by using the Gurobi optimizer.
9) The final section of "project.m" file reads the output obtained from the co-simulation of MATLAB and EnergyPlus and plots graphs of the readings.

The main modification we made was to the input idf description file. OpenBuild reads the initial presented idf file, parses it and adds the commented ExternalInterface objects as custom inputs to the intermediate idf file it creates and passes to EnergyPlus. Hence, we added our occupancy count as a people object along with the requisite schedule in the idf description file. As a result, occupancy data for the week entered by us was factored into HVAC calculations as well as final power output.

The commented section of the idf description file will include three new People objects for the three zones, three new ExternalInterface:Schedule objects declarations which

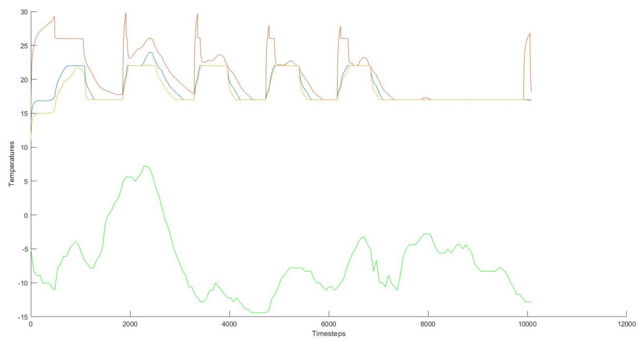

Fig. 1. Temperature vs Time

will be mapped to the above mentioned People objects, as well as definitions of the three new Schedule objects declared.

In order to parse these commented section and add them to the intermediate idf description file, changes also had to be made in certain MATLAB files. In ”building.m” file, the function addCustomInputs() was changed to create a new People object as per the definition required by EnergyPlus and then this newly created People class was passed to addObject function present in ”addObject.m” file. Changes were made in this function also.

In order to extract the occupancy and lighting data from the output entries, changes also had to be made in ”add-Output.m” file. This allowed us to extract the required data (already calculated from EnergyPlus and Gurobi optimizer) and plot graphs for the same.

IV. EXPERIMENTAL RESULTS

We were able to successfully integrate occupancy data with OpenBuild framework. The HVAC and lighting systems were able to factor in the occupancy data into their power calculations; and this output result was captured in the graphs plotted.

Figure 1 shows the graph between temperature and time. The green curve shown is the temperature outside the building, which is pretty low due to the season being considered is winter. The remaining three curves shows the temperature within the three zones, and this temperature varies between 15 and 30 degree Celsius, which have been considered as optimal indoor temperatures for maximum comfort of the occupants.

Figure 2 shows the graph between HVAC power and time. The operation of the HVAC has been observed and calculated according to the simulation parameters passed from EnergyPlus to the Gurobi optimizer and captured in the graph. There are three curves in the graph which correspond to the HVAC power inputs in the three zones respectively. As can be observed in the graph, the HVAC operation moves from on to off when number of people(occupancy) reduces and moves from off to on when number of people(occupancy) increases. This helps us in understanding the behavior of the HVAC with respect to occupancy.

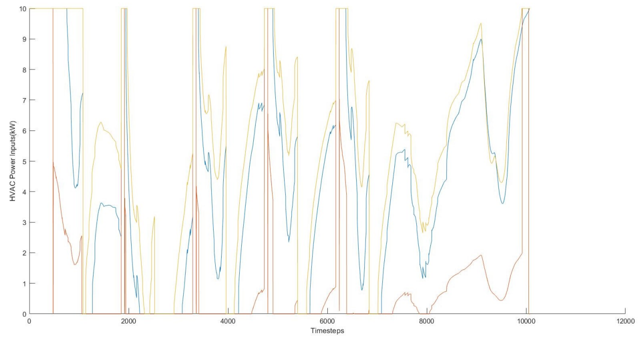

Fig. 2. HVAC Power vs Time

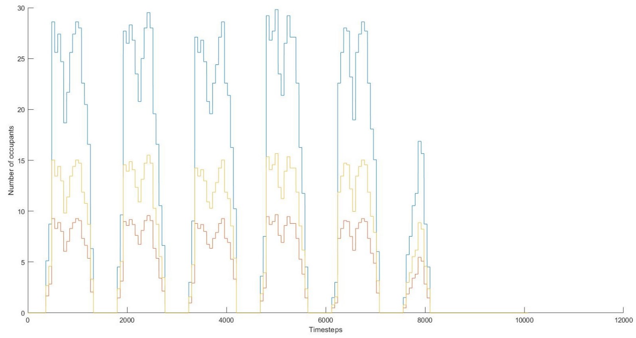

Fig. 3. Occupancy vs Time

Figure 3 shows the occupancy data for a week. The three curves correspond to the per zone occupancy. As can be seen from the graphs, occupancy increases during the day and decreases during the night on weekdays; increases to a lower number on Saturdays and remains zero on Sundays.

Figure 4 shows the lighting power used during the week. The three curves correspond to the zone wise distribution of power due to lighting. Depending on the occupancy, power on lighting increases or decreases per zone on every day of the week.

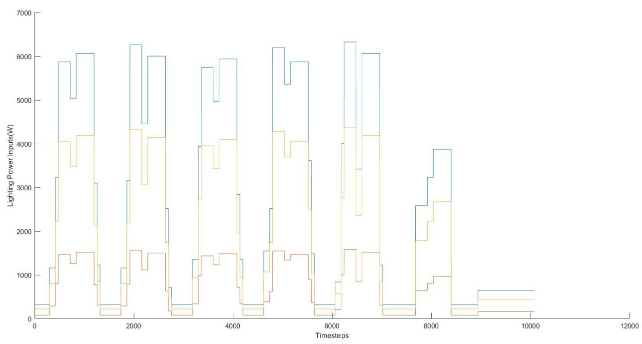

Fig. 4. Lighting Power vs Time

V. CONCLUSIONS

We have been able to successfully incorporate occupancy scheduling into OpenBuild framework. We had planned initially to use occupancy dataset from FORK, but were unable to do so, due to the complexity of the dataset involved and the limited time available to us after we got OpenBuild up and running. However, our initial tests were successful in yielding a positive result and this project has enabled us to learn MATLAB, EnergyPlus and co-simulation between the two to a great extent. We hope to further our knowledge and experience in the same fields by conducting more experiments and addition of more functionalities.

REFERENCES

[1] "OpenBuild: An Integrated Simulation Environment for Building Control", By Tomasz T. Gorecki, Faran A. Qureshi and Colin N. Jones, April 16, 2015

[2] https://www.rcrwireless.com/20160725/business/smart-building-tag31-tag99

[3] https://wiki.uiowa.edu/display/greenergy/Smart+Buildings

[4] https://en.wikipedia.org/wiki/Building_automation

[5] https://www.energyplus.net/

[6] http://www.gurobi.com/

[7] http://la.epfl.ch/files/content/sites/la/files/shared/common/openBuid/Manual.pdf

[8] http://la.epfl.ch/page-106260-en.html

[9] EnergyPlus Documentation : InputOutputReference.pdf, AuxiliaryPrograms.pdf, GettingStarted.pdf, ExternalInterfacesApplicationGuide.pdf