Building Energy Management Internet of Things

Reece Bachman Robert O'Malley Jordan Ingram Advisor: Dr. Suruz Miah

Saturday, May 4, 2019

BRADLEY University

Outline

Introduction

- Building Energy Management Open Source Software (BEMOSS) is an open source Internet of Thing (IoT) software
- BEMOSS was developed at Virginia Tech under Department of Energy funding

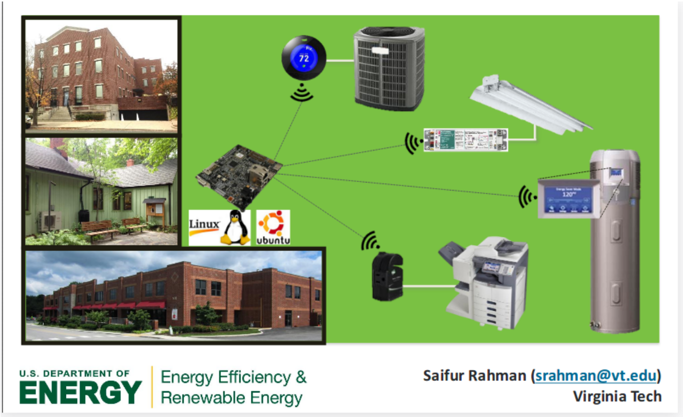


Figure 1: Internet of things

BEMOSS demonstration

- Light demonstration

Building Automation

Contributions

- Created a new device that can be controlled through BEMOSS over the internet
- Created an interface to control a DC motor through BEMOSS
  - Control curtains/ blinds to regulate interior temperatures
  - Control curtains/ blinds to regulate light
  - Close/ open barriers
- Created and simulated PID control algorithm for DC motor control
- Created and simulated LQR control algorithm for Heating Ventilation Air Conditioning (HVAC) system

Outline

Division of Labor

- Jordan
  - Develop accurate models using Simscape
  - Algorithm for BEMOSS
  - HVAC optimal control algorithm
- Reece
  - Develop new DC motor and interface
  - Design motor control circuit
  - Develop network Search and Control algorithm
- Robert
  - Transmit instructions from BEMOSS server to Raspberry Pi
  - Implement new functionality to BEMOSS

Building Automation

Motivation

- Internet of things is a rapidly expanding market
- Universal control of IoT devices
- Increased opportunity for efficiency

BEMOSS Structure

Software Architecture

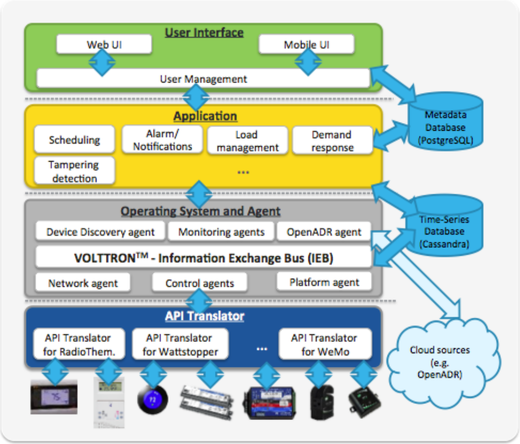


Figure 2: BEMOSS layers

Demonstration

- Motor demonstration

Developed IoT Device

Proposed BEMOSS Structure

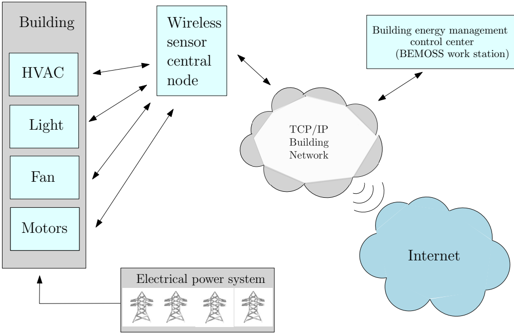


Figure 3: Proposed BEMOSS structure

- XBee S2C radio module
  - Receive motor direction commands
  - Relay encoder positional data to central node
- L298 H-bridge
  - control motor rotation direction
- Pittman 24v DC motor
  - Close/ open curtains
- Optical encoder
  - Measure and relays positional data
- Buck/Boost Converter
  - Supplies board logic power

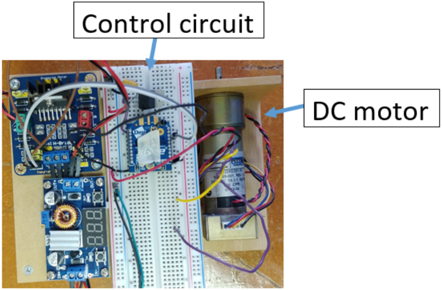
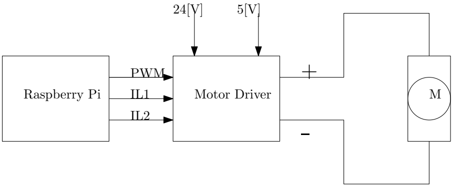


Figure 4: Remote DC motor

Developed IoT Device

DC Motor Connections



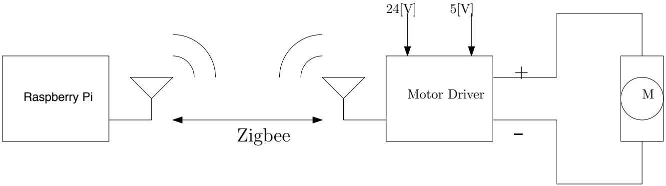


Figure 5: Raspberry pi motor driver interface

Developed IoT Device

Central Node

- Raspberry Pi Model 3B
  - Connect to BEMOSS
  - TX pin to transmit commands
  - RX pin to recieve encoder data
- Python Script:
  - Transmit remote AT commands to XBee Module
  - Process encoder data and send stop start command
- XBee S2C radio module
  - Coordinator XBee

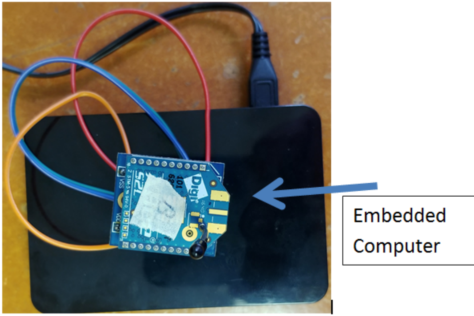


Figure 6: Central node

- Radio Frequency (RF) communication modules
- Operate at 2.4 GHz
- Allow single data processing station
- Control of numerous motors through one WiFi connection
- Initialized and controlled via XCTU software and serial commands

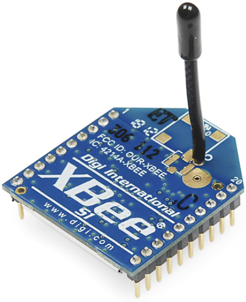
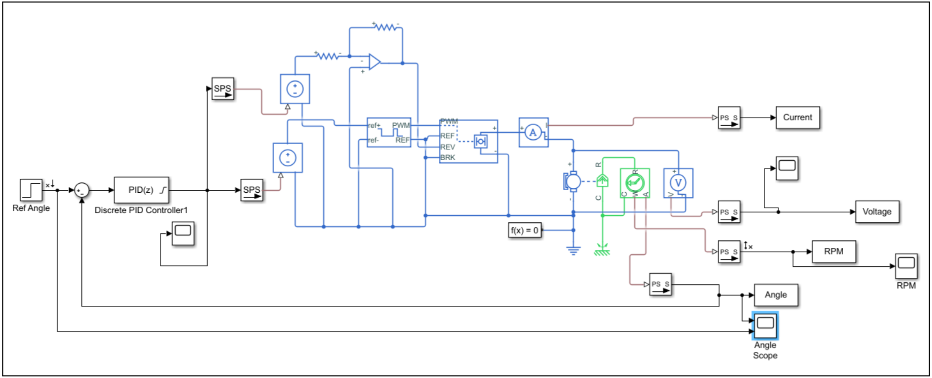


Figure 7: XBEE radio frequency module

Outline

BRADLEY University

Outline



Ref Angle
PID(z)
Discrete PID Controller1
SPS
SPS
ref+
ref-
PWM
REF
PWM
REF
REV
BRK
A
V
f(x) = 0
PS S
Current
Voltage
RPM
RPM
Angle
Angle Scope

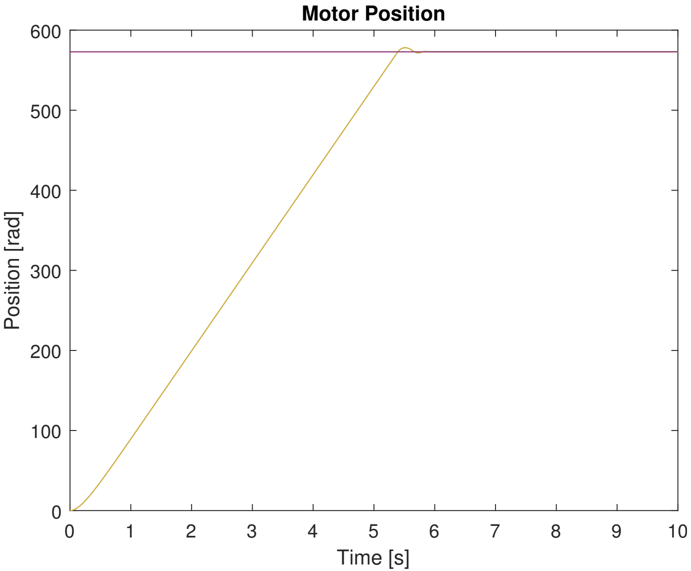
Outline

BRADLEY University

Motor Requirements

- The model is modeled based on the Pittman GM8024 24V DC Motor
- The model is run by a PWM signal and H-bridge
- A PD controller is used to control the position of the motor

Motor Model Results



HVAC Modeling
Simscape Model

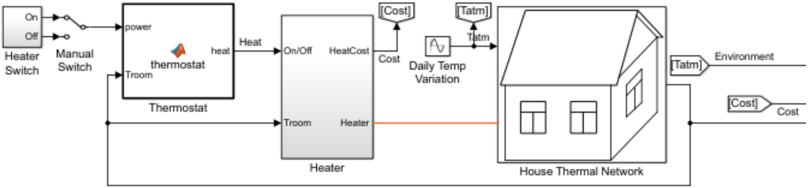


Figure 9: Simscape house model

HVAC Modeling

House Simscape Model

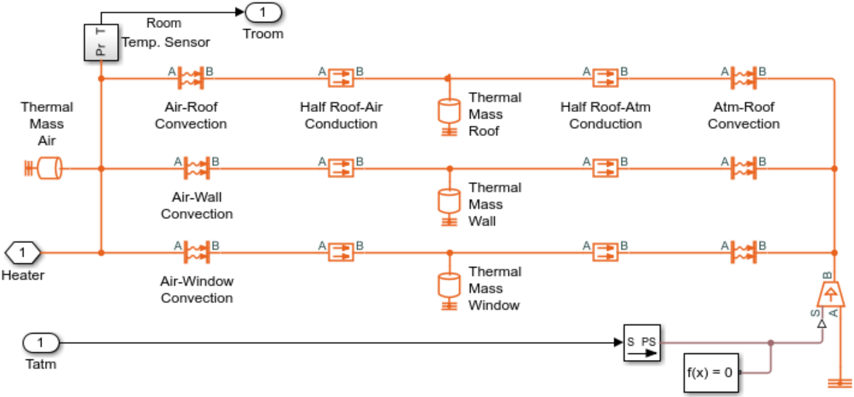


Figure 10: Simscape internal thermal house model

HVAC Modeling

Model Specifications

- On-Off control based on temperature threshold
- External Temperature changes with time
- Generate a cost associated with the energy required to heat the system

HVAC Modeling

System Response

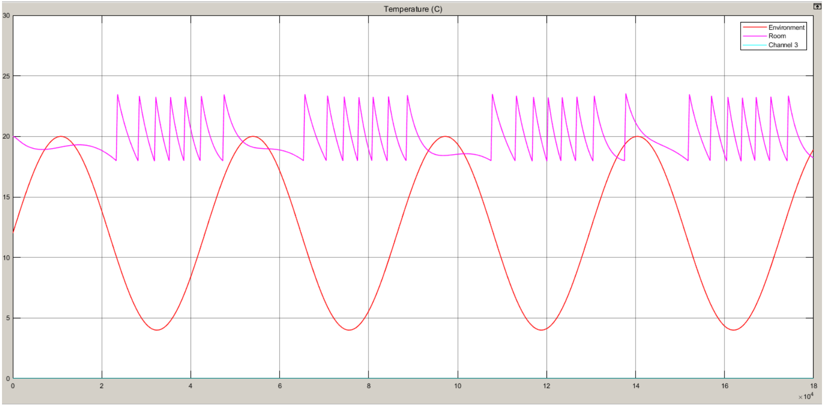


Figure 11: House model data

- More robust control designed preferred
- Linear Quadratic Regulator (LQR) is an optimal control design
- More easily allows MIMO control

HVAC Modeling

LQR Specifications

- Based on the work from research paper from Dr. Kang
- Single room system that consists of: a heat exchanger, a chiller, an air fan, ductwork, and dampers

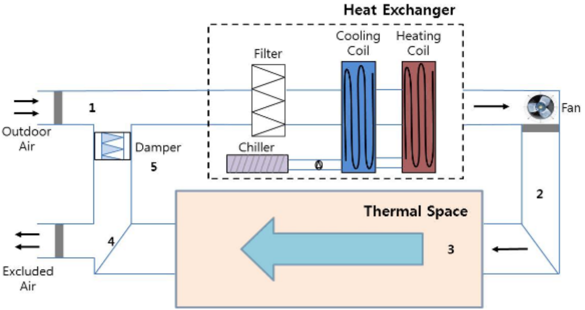


Figure 12: Conventional HVAC model

LQR Specifications

- Three State MIMO system that controls, Temperature, Humidity, and CO_2 levels
- Create a linearized model of the system to create controller and apply linearized control to the nonlinear system
- Model created using MATLAB/Simulink

$$\boldsymbol{A} = diag\left[\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\0 & 0 & 0\end{bmatrix} \begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\0 & 0 & 0\end{bmatrix} \begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\0 & 0 & 0\end{bmatrix}\right];$$

$$\boldsymbol{B} = diag\left[\begin{bmatrix}0\\0\\1\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix}\right];$$

and $\mathbf{C} = \mathrm{diag}[[1 \quad 0 \quad 0] \quad [1 \quad 0 \quad 0] \quad [1 \quad 0 \quad 0]]$.

BRADLEY University

$$\bar{A} = \begin{bmatrix} A & 0_{nxp} \\ -C & 0_{pxp} \end{bmatrix} \bar{B} = \begin{bmatrix} B \\ 0_{pxm} \end{bmatrix} \quad (1)$$

- Define the Q matrix based on the weights of the state variables
- Define the R matrix based on the weights of the control variables
- Create the K gain values for the optimal PI control through the LQR MATLAB function

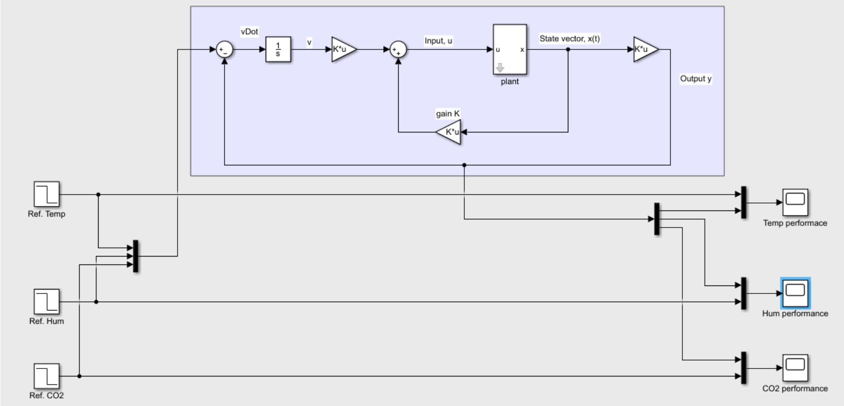
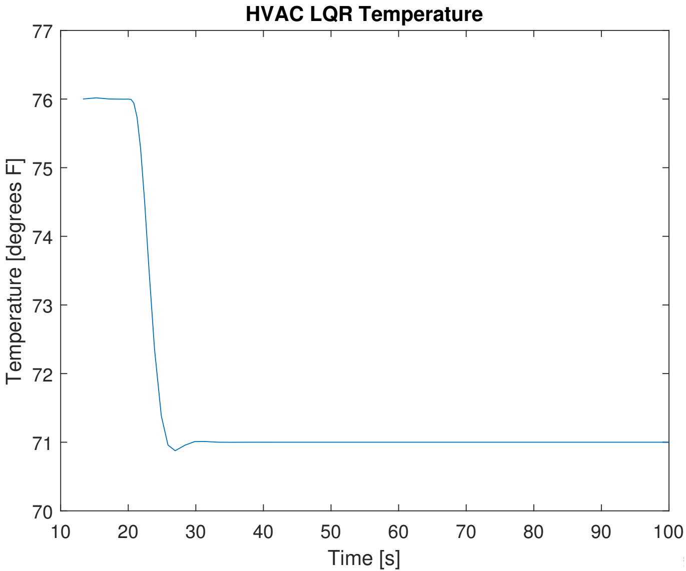


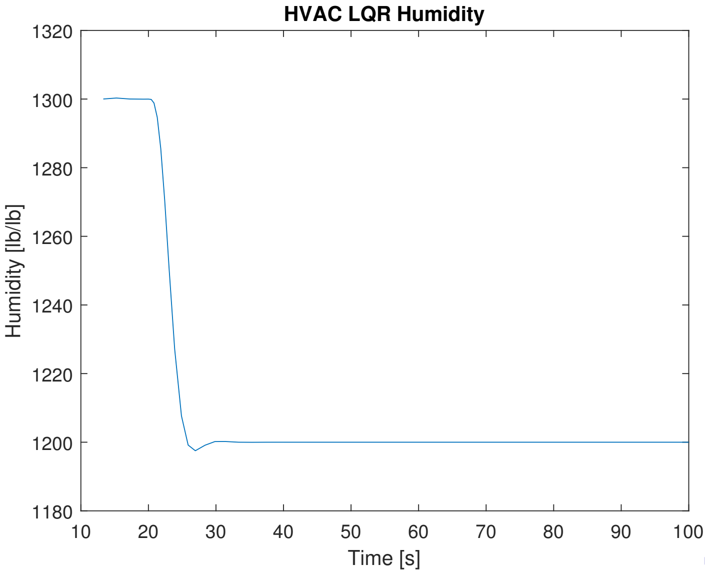
Figure 13: HVAC LQR Simulink model

HVAC Modeling

LQR Results

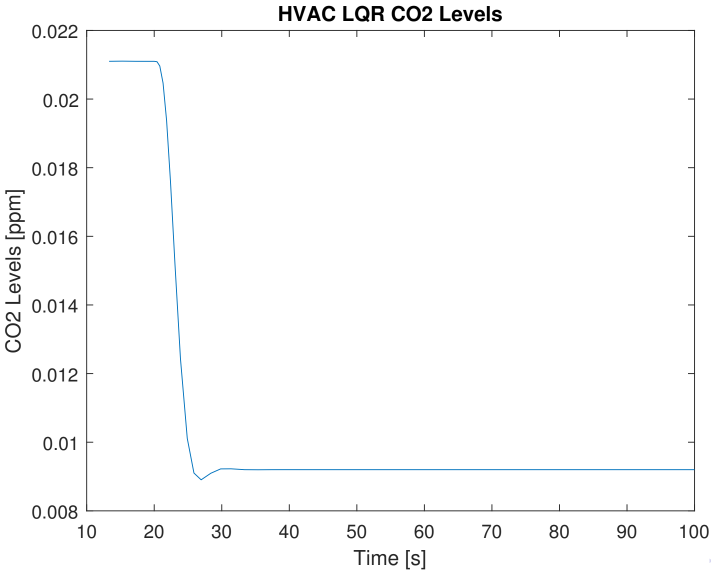


HVAC Modeling
LQR Results



HVAC Modeling

LQR Results



BRADLEY University

Outline

- Acquire all connected WiFi credentials
- Discover desired device via MAC address
- Shell into associated IP

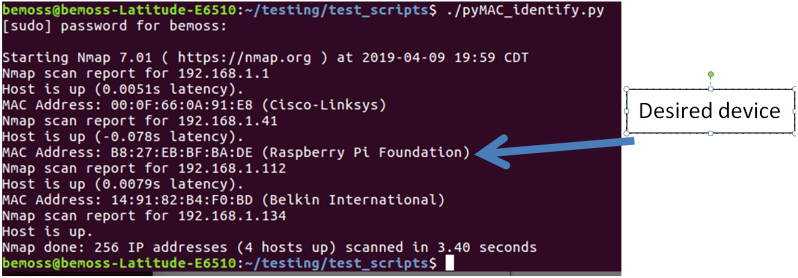


Figure 17: WiFi search and control method output

Outline

BRADLEY University

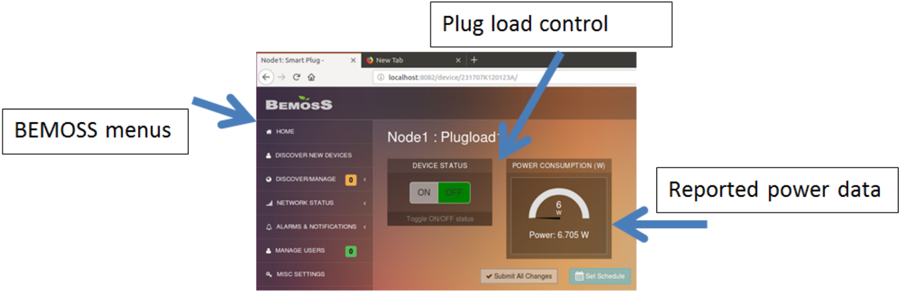


Figure 18: BEMOSS plugload information

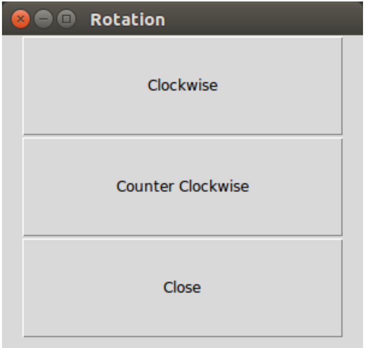


Figure 19: Motor control GUI

Engineering Efforts

Future Work

- Utilize logic converters to allow data exchange between the 5V logic H-bridge and encoder to the 3.3v logic XBee module (Pictured in "Remote Circuit 10/27/18)
- Employ I2C protocol in place of XBee radio communication between nodes to explore a cheaper communication method
- Implement feedback control using the motor encoder to control the position and speed of the motor
- Further energy efficiency by only powering the encoder during times of motor rotation (through use of relays)
- Further energy efficiency by using an algorithm to open curtains in times of environment and lighting control needs

BRADLEY University

Summary

- Design and introduce a new supported device within BEMOSS
- Configure BEMOSS to work on a single board computer like raspberry pi
- Develop an energy saving algorithm for BEMOSS

Discussion

Questions?

For Further Reading I

P. M. Ferreira
Neural networks based predictive control for thermal comfort and energy savings in public buildings/
Energy and Buildings, 2012

V. Reppa
A Distributed Architecture for HVAC Sensor Fault Detection and Isolation.
IEEE Transactions on Control Systems Technology, 2015.

X. Zhang
Deploying IoT devices to make buildings smart: Performance evaluation and deployment experience.
2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)