UNIVERSITY OF CALIFORNIA
RIVERSIDE

Design and Implementation of Sustainable Smart Beehive Monitoring

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Electrical Engineering

by

Dev Bhaveshbhai Joshi

December 2022

Thesis Committee:

Dr. Hyoseung Kim, Chairperson
Dr. Daniel Wong
Dr. Shaolei Ren

The Thesis of Dev Bhaveshbhai Joshi is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to deeply thank my thesis advisor Dr. Hyoseung Kim for continously supporting me during the course of this research. He believed in me from day one and also truly encouraged me to overcome unforeseen obstacles during my research. I would also like to thank my committee members Dr. Daniel Wong and Dr. Shaolei Ren for their valuable time and efforts in evaluating my work. I would also like to thank my teammates: Mohsen Karimi, Hoora Sobhani, Johnson Zhang, Shamima Hossain and, Avadhesh Rathi for their valuable help and support during this research.

Also, I would like to express my sincere gratitude to my mother "Meena Joshi", who gave me unconditional love and support throughout my research and made it possible for me to complete this.

To my late father and my grandparents, who were my greatest supporters.

ABSTRACT OF THE THESIS

Design and Implementation of Sustainable Smart Beehive Monitoring

by

Dev Bhaveshbhai Joshi

Master of Science, Graduate Program in Electrical Engineering
University of California, Riverside, December 2022
Dr. Hyoseung Kim, Chairperson

Honeybees are the primary pollinators in the world and are very important for the global food supply. Studies have shown that the honeybee population is declining sharply due to immature apiary management, various infections and multiple other stressors. Thus, autonomous beehive monitoring is extremely important to closely monitor environmental parameters in beehives without interfering with their normal activities. Recent work has studied this issue and various autonomous beehive sensing systems were proposed. However, most of them are powered by non-rechargeable batteries or wall power which limits the portability of the system. Furthermore, some prior work included various renewable energy harvesting techniques but they did not focus on the energy response of the system. In this thesis, we introduce a sustainable and smart beehive monitoring system that utilizes an energy harvesting unit to power a set of devices that collects diverse sensor data from beehives and send it to a server for analysis purposes. In particular, our system includes an energy-aware feedback loop control algorithm that is proposed to dynamically control the sensor period based on energy availability. Our system was deployed in a remote apiary and

collected various environmental parameters for more than 8 months without requiring any maintenance. Simulation results of the energy-aware feedback loop control system showed that our system kept the utilization of the system high as compared to baseline systems, while managing to keep the energy storage to a predefined set value.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Honey bees play a very important role in strengthening the agricultural economy of any nation due to the products such as honey, beeswax, propolis, royal jelly, etc. According to the United States Department of Agriculture (USDA), honey bees are also the primary pollinators in the US, pollinating almost 130 different crops/flowers every year worth $18 billion [2]. However, various environmental and biological stressors to honey bees, including improper apiary management, the spread of varroa mites and other parasites, climate change, the use of pesticides, poor nutrition, etc., lead to Colony Collapse Disorder (CCD) [13], which is the phenomenon where a majority of adult bees disappear, leaving food in short supply for immature bees and queen [37]. Thus, continuously monitoring beehives is extremely important to estimate the health of the bees and take early actions to mitigate the impact of any abnormal condition inside the hive. However, traditional beekeeping involves frequently opening the hive to manually check the health of the bees. Such practices add to the stress levels of the hive, negatively impacting honeybee health. In this thesis, we introduce a smart and sustainable beehive monitoring system that enables automatic and noninvasive beekeeping and help bees to live longer without human intervention.

In recent years, significant efforts have been made by researchers to utilize the Internet of Things (IoT) to create beehive monitoring systems. Precision Beekeeping (PB) [38] is a new branch of Precision Agriculture, where recent technology advancements are used to automate hive monitoring and control. In [5, 20, 35], acoustic signals are used to monitor the status of the beehive and predict potential swarming effects. On the other hand, many researchers focused on imagery data along with micro-climate parameters to monitor beehive health. However, such monitoring systems have very high energy requirements and either use battery or wall power as an energy source. Utilizing non-rechargeable batteries requires frequent maintenance for battery changes and can increase the stress levels of bees. Wall power connections are not generally available in remote apiaries. Therefore, to address such limitations, it is required to develop an approach that can harvest energy from ambient sources and efficiently manage harvested energy while ensuring continuous operation of honeybee monitoring with various sensors.

The aim of this thesis is to design and implement a sustainable and smart beehive monitoring system. Our system utilizes a solar energy harvesting unit to power connected IoT devices gathering important environmental parameters inside the beehive. The environmental parameters collected by our system include temperature, humidity, pressure, CO2, total volatile organic compound (TVOC), and luminosity data. The collected sensor data is transmitted to a server through a gateway node so that beekeepers can view and analyze beehive conditions during the times when traditional beekeeping is not possible (e.g. night time, rainy days). Our system achieves sustainability by introducing energy-efficient sensor scheduling techniques. For this purpose, unlike prior work and conventional bare-metal IoT devices, we use a real-time operating system and follow the multi-threaded execution model to offer better schedulability and more flexible management of various sensors with different data freshness requirements. In addition, our system is smart in that it introduces an energy-aware feedback loop control algorithm to dynamically control the sensor periods based on energy availability and keep the beehive monitoring system running for a longer period of time as compared to conventional monitoring systems. A rudimentary version of the beehive monitoring system proposed in this thesis was deployed to a remote apiary in Riverside, California. Our system showed promising performance by continuously gathering data for 8 months without the need for battery change.

To summarize, the contributions of this thesis are as follows:

- Develop a sustainable beehive monitoring system that collected multiple seasons data for different environmental parameters by in-field deployment of the proposed system for more than 8 months,
- Propose a multi-threaded approach for monitoring tasks on a real-time operating system to improve the schedulability of the system,
- Propose a simple yet efficient energy-aware feedback loop control algorithm to improve the responsiveness of the system based on energy availability.

The organization of the thesis is as follows: Chapter 2 reviews recent work in the related field; Chapter 3 explains the system overview of our approach for both sensor architecture and software characteristics of the proposed system; Chapter 4 gives a brief overview of the in-field deployed system; Chapter 5 briefly notes the challenges faced during the deployment; Chapter 6 presents the results of deployed system and the proposed sensor period control algorithm; Chapter 7 concludes the thesis and briefly talks about the future work.

Chapter 2

Related Work

The work in this thesis combines two areas of research - beehive sensing and energy harvesting system. These areas have been widely explored before and this section discusses some recent related work in the field of beehive sensing and energy harvesting systems.

2.1 Bee Hive Monitoring

Many systems have been proposed to collect micro-climatic parameters from inside and outside the beehive in real time. In [34], the authors proposed an intelligent beehive monitoring system that collects important parameters along with the GPS positioning data of a beehive and uploads the collected data to a cloud-based server for graphical representation and notifications to the beekeepers. The authors of [9] introduced an advanced monitoring system that acquires data like sound, weight, temperature/humidity, and CO2 and analyzes the data with regard to different abnormal activities inside the hive. A WSN-based system is proposed in [16] to count the number of foraging bees using IR LEDs installed at the entrance of the hive. It also collects temperature and humidity values from inside the hive to correlate the bees' incoming/outgoing activities with different sensor values. Although the approach used by these works may give an accurate picture of the current state of the beehive, they have used AC-powered edge sensing nodes which can limit the use of their systems in remote places. A beehive monitoring system has to be a portable system, as the beekeepers regularly have to change the location of hives depending on the season.

In [28], a data acquisition system is proposed where edge devices collect data after every 2 hours and send the data over to the data hub node using a wireless connection. Although the communication medium is wireless, the beekeepers have to physically visit the monitoring site in order to retrieve the data from the data hub node to a smartphone. Another study [21], proposed a bee hive monitoring system that notifies the beekeepers of a potential absconding process by getting the temperature data every hour from inside the hive. The system tries to detect the temperature rise inside the hive using predefined thermal patterns and only alerts the beekeepers when overheating is detected. However, these systems work on battery power and require beekeepers to open the hive for battery change. Honeybees are very sensitive to temperature change and opening the hive frequently increases the stress level because of massive fluctuations in micro-climate parameters inside the hive resulting in degraded bee health.

The authors in [27] introduce a self-powered beehive monitoring system that makes use of piezoelectric, electromagnetic, and bee vibrations energy harvesting techniques to power the system. It captures data from the beehive and also adds electrical systems to keep the temperature of the beehive in an optimal range. Additionally, it introduces a layer of security for the hive by adding flame sensors and digital cameras to alert potential threats to the bees. Although the plethora of energy harvesting mechanisms helps provide power more reliably compared to a single energy harvesting unit, the cost and the complexity added to include all the energy systems can make this design impractical to use. Moreover, the actual amount of energy harvested from harvesting mechanisms described in that paper could be insufficient for systems deployed in the environment due to the varying availability of energy sources.

2.2 Energy Harvesting-enabled Sensing

Modern IoT and WSN technologies are shifting towards sustainable energy sources for sensing applications. Of all sustainable sources of energy, sunlight is the most widely used one. In [30], a solar-powered agricultural monitoring system is proposed to achieve vital information on plant health. The battery performance of two nodes is compared where one node is just running on a LiPo Battery and the other node is identical but has a solar panel connected to it. A proof-of-concept of this system was achieved inside a controlled environment in a lab and the experimental results show that the solar-powered node can run significantly longer than the normal node. This shows that a sustainable energy source can be very helpful to make systems run for a longer period of time.

In [12], an energy-neutral platform for beehive monitoring that uses imagery and sound data to monitor the beehive status was introduced. The solar-powered design used by the authors can help achieve fully autonomous monitoring for longer periods of time. In

[4], the authors proposed a beehive monitoring system that is powered using solar panels and collects the environmental parameters from inside the hive every hour. The system also gathers image data at the entrance of the hive. In [11], authors proposed an algorithm based on decision trees that describes the state of the beehive with noticeable accuracy. It utilizes a solar-powered WSN-based system that collects data 3 to 6 times per day. However, such systems are not responsive to scenarios where sunlight is not available for extended periods because they are following a static period for data acquisition. This can make the system fail when the battery runs out of charge which would result in the loss of valuable data required for monitoring beehive health.

The authors of [3] developed a system that is used for aquatic environmental monitoring. The system design includes an energy harvesting unit that contains a solar panel and a twin battery setup to harvest solar energy. The energy harvesting unit switches the power from the batteries once the input power of the current battery drops below 4 V. This twin battery setup can help to offload the power load to a single battery and also help to achieve a longer operating period. However, this design lacks an energy-saving mechanism and thus may fail to detect events when energy harvesting is significantly lower than consumption.

2.3 Power Management

The overall power consumption of a system is basically a summation of dynamic power and static power. Dynamic power is the power consumption of a computing system when it is actively performing operations. On the other hand, static power (or leakage power) is the power consumed by the system, when it is in the idle or sleep state. There are many techniques available like dynamic voltage scaling (DVS), dynamic frequency scaling (DFS), and dynamic voltage/frequency scaling (DVFS) which focuses on lowering the operating voltage and/or frequency of the system to decrease the dynamic power consumption. However, such scaling techniques can dramatically increase the execution time of the system and eventually result in high energy consumption or negatively impact the responsiveness of the system. Moreover, the majority of microcontrollers used in low-power computing devices do not support such scaling techniques. Thus, to deal with this problem, the time spent by the system in sleep mode should be increased to reduce the power consumption of the system.

In [29], authors introduced the Rate Harmonized Scheduling (RHS) technique that increases the sleep time of the system by clustering the execution of tasks. The authors also proposed the Energy Saving Rate Harmonized Scheduling (ES-RHS) which utilizes a periodic sleep task that aims to convert all the idle states of the system to the sleep state. Also, [10] proposed ES-RHS+, Energy Saving Rate Monotonic Scheduling (ES-RMS), SyncSleep and Max-SyncSleep Scheduling that relies on the ES-RHS technique that we talked about earlier and focuses on maximizing the time spent by the system in deep sleep mode to achieve maximum power/energy saving. Such scheduling policies can help increase the time spent by the system in sleep mode dramatically and efficiently. However, these scheduling policies only support constant periodic tasks that are known beforehand.

Chapter 3

System Design

This section presents the design of our monitoring system. We first provide our system architecture and then give a detailed explanation of the software and algorithmic components.

3.1 Sensing System Architecture

Figure 3.1 shows the overall connection of our system components. To achieve the aforementioned goals, the system includes a set of two edge sensing devices, each of which collects different environmental parameters of the beehive in real time. The edge sensing devices are powered by a solar panel when sunlight is available and by a LiPo battery in the absence of sunlight. These devices send the collected data to a gateway node installed in the vicinity of the beehives through a wireless connection. The data received by the gateway node is forwarded to a cloud-based server using an Internet connection. Lastly, the data saved on the cloud-based server can be accessed by the end user (beekeepers) in

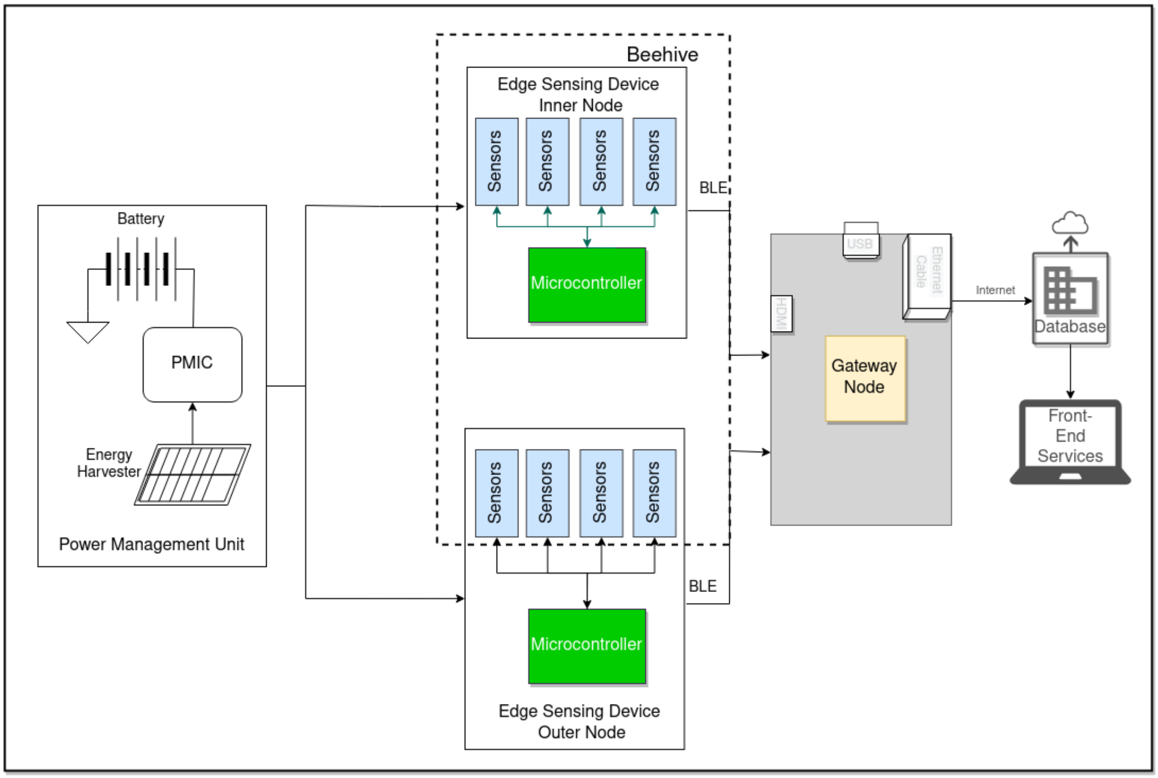


Figure 3.1: Sensing System Architecture

real-time using any device with an active Internet connection, which will help beekeepers to monitor the beehive's environmental conditions without physically visiting the beehive.

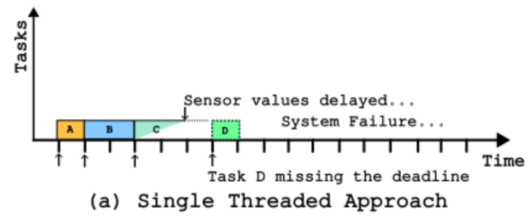
A wired connection between edge sensing devices and the gateway node can limit portability, which is important because beekeepers seasonally migrate the beehives. Thus, having a wireless connection is of utmost importance for a beehive monitoring system. In our system, Bluetooth Low Energy (BLE) is used for data transfer between the edge devices and the gateway node. A wireless communication medium for beehive monitoring has to be reliable because the density inside the hive increases as the bees work and generate honey making it more difficult to connect to the gateway node. Along with that, it has to be low power consumption as the data acquisition nodes are running on battery power in the absence of sunlight and higher energy consumption for radio communication can dramatically decrease the battery life of the system.

As shown in figure 3.1, one edge sensing device is placed inside the hive (i.e. on the center frame of the beehive) and the other device is placed outside the beehive. The device installed on the center frame collects different data like temperature, humidity, pressure, CO_2 (Carbon Dioxide), luminosity, accelerometer, and TVOC (Total Volatile Organic Compound). Another edge sensing device collects the temperature of all 8 corners of the beehive and also the ambient temperature for reference. Beehives use different techniques to keep the temperature inside the hive in the optimal range (i.e. 33 - 35 °C) [15]. Many studies have shown that temperature/humidity is a vital metric to understand the absconding and swarming effects in beehives. Hence, measuring the temperature/humidity inside the beehive can help understand the overall health of the beehive. Additionally, previous research has shown that collecting temperatures from multiple points inside the beehive can help understand the distribution of the bees and brood volume. Although no algorithms were developed to estimate the brood distributions but the system can be extended to implement such algorithms, left as future work.

Along with temperature/humidity, CO_2 is also a very vital data/metric for beehives. As discussed in [24], honey bees maintain CO_2 gas levels inside the hive and the authors in [6] explain how CO_2 levels can help bees control varroa mites during the winters. To mitigate the presence of varroa mites inside the beehive, beekeepers evaporate formic acids inside the hive as mites do not have resistance against organic acids. Utilizing TVOC sensors can help estimate volatile organic compound levels after formic acid treatments inside the hive as described in [31].

Luminosity and accelerometer metrics are used for safety concerns of the beehive. Beehives are always kept closed and in a particular orientation. The change in orientation or luminosity values inside the hive implies that the hive can be opened or blown away by wind or some intruder. Hence, collecting such metrics can help beekeepers to check if the beehives are intact or need their attention. Thus, collecting different environmental parameters described earlier provides insights into the performance of the beehive and helps beekeepers access the health and security of a colony.

One of the major goals of our design is to make a sustainable system that provides continuous real-time data from inside the hive over extended periods. To achieve this goal, we installed an energy harvesting unit that can power both edge sensing devices continuously without the need of changing batteries. The reason behind this is that opening the beehive for maintenance brings massive fluctuation in the hive's micro-climate resulting in increased stress levels for bees and thus decreased productivity. This energy harvesting system consists of a Power Management Integrated Circuit (PMIC) connected to a solar panel. The system is designed to continuously monitor the beehive during the night and for extended periods without sunlight by utilizing a rechargeable Li-Po battery. The role of the PMIC is to charge the Li-Po battery and power the boards simultaneously whenever sunlight is available. Additionally, in the absence of sunlight, PMIC powers both boards from battery storage. Hence, our system design avoids opening the beehive for changing the battery by using a combination of PMIC chip, solar panels, and batteries.



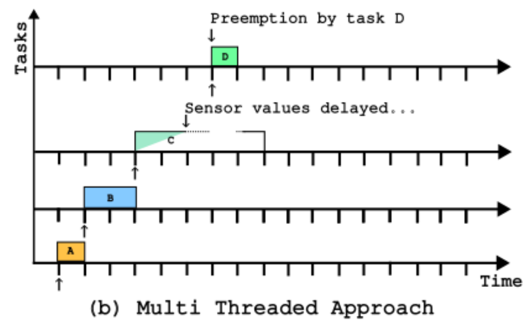


Figure 3.2: Single Threaded vs Multi Threaded Task behavior

3.2 Software Architecture

In this section, we introduce the design choices made by us in the software architecture to support a smart beehive monitoring system. Specifically, we use the multi-threaded execution model that offers better schedulability and supports assigning different periods for various sensor data freshness requirements.

The majority of the previous work [12, 9, 11, 26] that focused on smart beehive monitoring had utilized an Arduino or Raspberry Pi, which is usually implemented using a single-threaded approach. In that approach, the microcontroller polls all the sensors in a sequential manner and repeats the same set of actions after a finite amount of sleep time. Although such a system is very easy to implement and can be run without the help of an operating system, it can introduce a lot of issues that can lead to system failure and poor responsiveness. For example, as shown in figure 3.2, it is possible that the periodic task C gets blocked for a finite time while waiting for a sensor to generate the required data. Due to this, it is possible that the next task D in the execution path might miss its deadline and can negatively affect data freshness. Hence, the responsiveness of the single-threaded system is very poor as compared to the multi-threaded system. Lastly, a single-threaded design is neither easily scalable nor modular to support additional sensors.

To overcome the shortcomings of the single-threaded execution model, we followed a multi-threaded execution model where each sensing thread can communicate and receive raw data from a single sensor. In this approach, multiple periodic tasks/threads collect and send various data metrics from inside the hive to the gateway node. This approach follows the producer-consumer thread model. In this model, the periodic tasks are divided into two main groups: Producer and Consumer. The producer group consists of the periodic tasks that are communicating with the sensors on board and collecting the hive parameters. These collected parameters are placed in a shared memory of the microcontroller and the consumer group consists of a single periodic task that sends this data to the gateway node using a wireless connection protocol BLE.

3.3 Energy-aware Control of Sensor Periods

The responsiveness of the system with respect to energy availability is a critical factor for reliable beehive monitoring. As discussed before, most conventional approaches use a static sensor period without considering energy availability. Because of this, if there is little to no sunlight for an extended period (typical for the rainy season), it is possible that the edge device might exhaust all the energy and stop working in a short time (typically a few days). Additionally, if the energy availability exceeds the energy requirement, a responsive system can gather data more frequently as compared to non-responsive systems. Hence, it is fundamental to support a dynamic sensor data acquisition period that is adjusted based on energy availability.

To achieve the aforementioned goal, we propose an energy-efficient feedback loop control system utilizing Proportional-Integral-Derivative (PID) controller. The goal is to keep the battery voltage (V_{batt}) to a user-defined value. For ease of modeling, the voltage and energy are assumed to be proportional in such a way that at the maximum capacity of the energy storage device, the output voltage is at its peak and as the energy reduces gradually, the voltage also decreases. Thus, monitoring the battery voltage can be helpful in estimating the amount of energy left inside the battery and the total operating time. Our system calculates the error between instantaneous and desired battery voltage ($V_{desired}$), and depending on the output of the PID controller, a suitable sensor period is calculated. Sensor period values (T) are bounded by a maximum value to maintain the data freshness. Although PID is a very simple feedback controller, it is effective enough for our purpose and can be implemented on microcontrollers with low overhead.

$$e(t) = V_t - V_{desired} \tag{3.1}$$

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{\partial e}{\partial t} \tag{3.2}$$

The nature of the PID controller is to control a parameter by providing continuous feedback. In our system, PID controls the battery operating voltage (V_{batt}). As discussed before, the battery operating voltage covers a range of output voltages $[V_{min}, V_{max}]$ depending on the amount of energy $[E_{min}, E_{max}]$ stored inside it. The desired battery voltage value ($V_{desired}$) and a control period ($T_{control}$) can be set by the user depending on the typical sunlight availability onsite. After every control period, the system will read the battery voltage V_t at time t, and calculates the error $e(t)$ as shown in equation 3.1. This error is fed into the PID controller and it outputs the change required in the battery voltage depending on the control parameters: proportional gain (K_p), integral gain (K_i) and, derivative gain (K_d) as shown in equation 3.2. The proportional response (P) depends on the instantaneous error of the voltage. The integral response (I) adds all the errors until time t, and hence it depends on all the past errors until time t. Lastly, derivative response (D) calculates the rate of change of errors and controls the dampness of the system. The optimal values of the control parameters listed above are very crucial for the PID controller in order to get an ideal response from the feedback control system. Thus, this simple yet effective feedback loop control system increases the responsiveness as per energy availability and helps to collect environmental parameters from the beehive effectively. The next section describes how Rate Harmonized Scheduling can be applied to our energy-aware sensor period task model.

3.4 Energy-efficient Scheduling

As discussed above, an energy-aware feedback loop control system will change the sensor period depending on the energy availability of the attached battery. This will dramatically change the scheduling behavior of the system and can increase the energy consumption of the system. We can decrease energy consumption by coalescing the execution of all tasks. The aim is to maximize the time spent by the microcontroller in deep sleep mode. In this section, we discuss how an already developed energy-saving scheduling policy can be used to decrease the power consumption of our system.

Most microcontrollers support multiple operating modes (i.e. Idle, Sleep, Deep Sleep), where each mode consumes a different amount of energy. Depending on the operating mode, it may take different amounts of time to switch between the modes. The time taken to switch to an operating mode is inversely proportional to the amount of energy consumed in that mode. For example, the deep sleep mode (i.e. lowest energy consumer) requires more time to switch into or out of that state as compared to the idle mode which consumes more energy than the sleep mode but less than the active mode. Therefore, to decrease the overall power consumption, the microcontroller has to maximize the time spent in deep sleep mode. The relation between energy consumption and the time taken to switch to different operating modes is described in equation 3.3.

$$E_{active} > E_{idle} >> E_{sleep} \equiv T_{active} < T_{idle} << T_{sleep} \tag{3.3}$$

The Rate Harmonized Scheduling (RHS) policy [29] is known to be effective in reducing the energy consumption of a microcontroller by coalescing task execution and maximizing deep sleep time. In this scheduling, the execution of all the tasks in a task set utilizes a set of periodic values called harmonizing periods T_H. According to RHS, any task is only eligible to execute after the next immediate boundary of the harmonic period T_H.

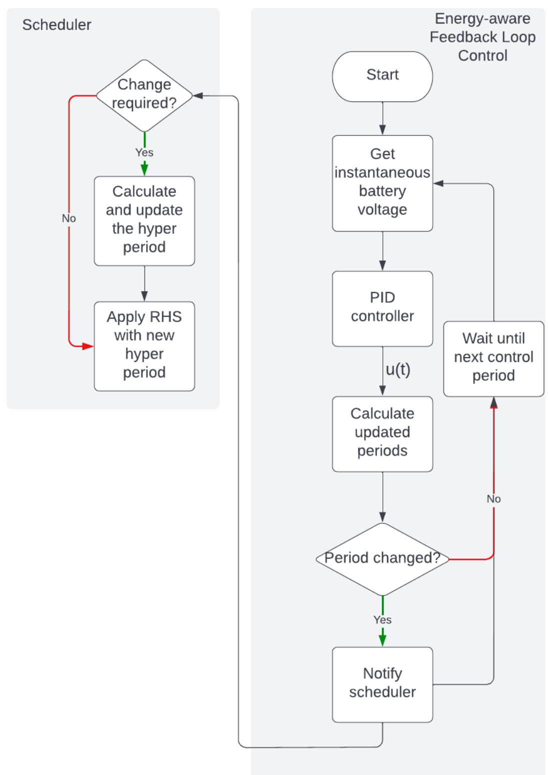


Figure 3.3: Software Architecture Overview

For basic RHS, harmonizing period for all the tasks is the same and equal to or less than the highest priority task ($T_H <= T_i$ and $T_H = T_H^1 = T_H^2 = T_H^3$). The priority of a task is decided by the Rate Monotonic (RM) policy, where the highest priority is given to the task with the lowest period.

However, the existing RHS method cannot be directly used with our approach as the period of the tasks is dynamically changing. If we do not change the harmonic period (T_H) each time the sensor periods are changed, it can lead to multiple tasks missing their deadline and eventually cause a system failure. Hence, we proposed a scheduling scheme that calculates the harmonizing period dynamically every time the period of the tasks is adjusted by the PID controller. As shown in figure 3.3, the scheduler is called by the energy-aware feedback loop control system after each control period. The scheduler gets the updated values of the sensor periods for the sensing tasks, and changes the harmonizing period equal to the period of the highest priority task and applies RHS scheduling. This set of operations repeats every time the period of tasks is changed by the feedback loop control system.

3.5 Abnormal Condition Detection

In addition to making the beehive monitoring system aware of energy availability, it should also be responsive to abnormal conditions inside the beehive. Premature swarming and absconding processes are two examples of such conditions. Swarming is a process where 60% of the bees including the queen leave the parental hive and look for a new hive. Absconding is a process where all the honey bees in a beehive leave the original hive and look for a new suitable hive. Both of these can happen because of several reasons like unsuitable environmental conditions, lack of food resources, predators, etc. In [33], it is explained that before a swarming or absconding process, the temperature values inside the hive increase rapidly. To detect such abnormal activities, we added a feature that checks for an abrupt increase in the temperature values and depending on that reconfigures the periods of the sensing tasks to 1 minute for the next 1 hour. If such abnormal activities inside the hive can be detected beforehand, beekeepers can take necessary steps to stop them. Currently, this feature is implemented using static threshold values for temperature metrics inside the hive. However, a more advanced algorithm can be easily implemented in the future to incorporate additional environmental parameters as well.

Chapter 4

Implementation

In this section we describe the implementation of our design described in the previous chapter.

4.1 Inner Edge Sensing Device

As shown in figure 4.1, one of the edge sensing nodes is installed on the center frame inside the beehive. An Adafruit feather sense board equipped with 1MB Flash, 256KB SRAM and a 2.4GHz radio module is used as the micro-controller unit for this node. Along with low power consumption, the board offers an ample amount of sensors attached to the board and the radio module provides Bluetooth Low Energy functionality for wireless communication. This node is equipped with sensors SHT31 (Temperature and Humidity), BMP280 (Temperature and Pressure), APDS9960 (Luminosity), LSM6DS33 (Accelerometer), SCD41 (CO2), BME680 (TVOC). All sensors are connected to the board using the Inter-Integrated Communication (I2C) protocol. As shown in figure 4.1, a 3-

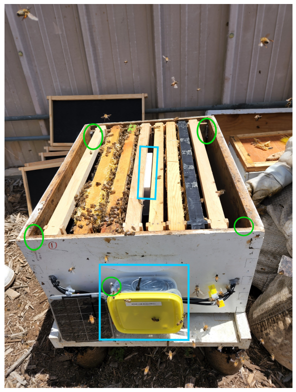


Figure 4.1: Components placement

Figure 4.2: System deployment

D printed box is used to enclose the edge device along with additional sensors. This is required to isolate the sensing device as honeybees tend to cover any foreign element within the beehive with wax.

4.2 Outer Edge Sensing Device

As shown in figure 4.1, this node is installed outside the beehive and closer to the solar panel. This node is equipped with 9 Dallas Temperatures DS18B20 sensors, where 8 of them are located across all the corners of the beehive and one sensor is located outside the beehive to acquire the ambient temperature for reference. The reason behind collecting temperature from all corners of the hive is that future analysis can predict the volume of the brood and the distribution of bees within the hive. Similar to the inner sensing device, the Adafruit feather sense board is used that communicates with all the 9 sensors using a 1-Wire communication protocol through a general purpose input-output (GPIO) pin on board. To make the system less invasive to the honeybees, we decided to place the PCB board outside the beehive in a container as shown in figure 4.2. The current choice of container can be replaced by a more sophisticated 3D-printed box in the future, to make the system robust during difficult environments.

4.3 Gateway Node

The gateway node is installed several meters away from the beehive. Gateway Node consists of a Raspberry Pi 4 equipped with 8GB RAM along with a 256GB SD card. A Raspberry Pi was used because of Bluetooth Low Energy support to communicate with the edge nodes and WiFi support to send the data over the internet to a data server. The role of the gateway node is to collect data from the edge sensing devices periodically and send them to a time series database server (InfluxDB) running on a virtual machine hosted by the ITS Department of the University of California, Riverside. Data stored on the database server is made available to the end users (beekeepers) using an open observability platform(Grafana). Thus, the system is designed to provide a very easy-to-use platform for the beekeepers to continuously monitor the micro-climatic parameters inside the beehive using any device connected to the internet.

4.4 Power Management

Both the data acquisition nodes are powered by a (6V 2W) Solar panel which is placed on top of the bee hive as shown in figure 4.2. The Solar Panel is connected to BQ24074 Power Management Integrated circuit (PMIC) from Adafruit. Additionally, a 5500 mAH LiPo battery is also attached to the PMIC chip to keep the edge sensing devices running during the absence of sunlight. The PMIC chip charges the battery along with powering the nodes whenever sunlight is available, and it delivers the power to the nodes from the battery during the absence of sunlight. The battery provides enough power to keep both nodes running for approximately 2 weeks without sunlight. This implementation of a power management circuit helps to keep the sensing device running continuously without the need of replacing batteries every few weeks.

4.5 Software Specifications

Both the edge sensing devices run on a Real-time Operating System(RTOS) called Zephyr which is an Open-Source Project fostered by the Linux Foundation. Two programming languages are used for this project: Python3.8 is used to create the application running on the Raspberry Pi (Gateway Node), while the sensing devices used C language for both the main application and the driver scripts that communicate with the sensors. The data acquisition period for both nodes is set to 20 minutes. This gives 72 total readings per day for both sensors.

As shown in Figure 4.3, the edge sensing devices gather the environmental data from the sensors using I2C or 1 Wire protocol and send the data over to the gateway using Bluetooth Low Energy communication protocol. Immediately after completing a whole cycle of data retrieval and transmission, both nodes are put into a deep sleep until it is time to perform another cycle. While the nodes are sleeping until the next wakeup event, it is expected that the system switches off all the power-hungry components (i.e. Bluetooth module, CPU) until they are not required. However, due to compatibility issues with Zephyr

RTOS, the Bluetooth module remained on all the time and this led to slightly high energy consumption for our approach.

Figure 4.4 graphically explains the role of the gateway node. As can be seen from the figure, the gateway node receives data from both the edge sensing devices and sends the data to an InfluxDB server. Raspberry Pi 4 utilizes a multiprocessing package to communicate with both sensing devices simultaneously. Each process receives environmental data from the devices and forwards it to the server individually. Since the apiaries are located in a very remote place, it is very likely that the internet connection is lost for an extended period of time. Keeping this problem in mind, the gateway node is programmed to save a backup copy of the data in the local memory(SDCard). This data can be easily retrieved by doing a remote SSH connection to the gateway node whenever the internet is available. This feature ensures that no data points are lost when the internet connection is lost. Lastly, all the data stored on the server can be accessed by using a front-end service called Grafana. This service provides extensive options to visualize the data, which can be very helpful to beekeepers in checking the beehive conditions anytime during the day.

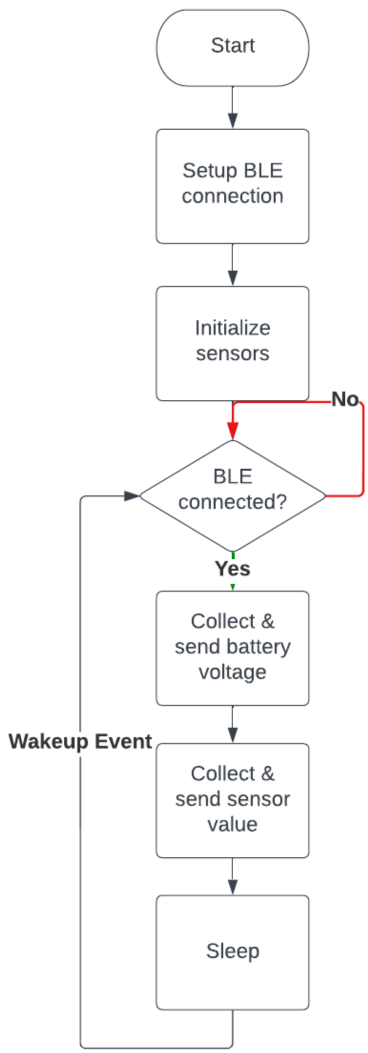


Figure 4.3: Edge Sensing Device

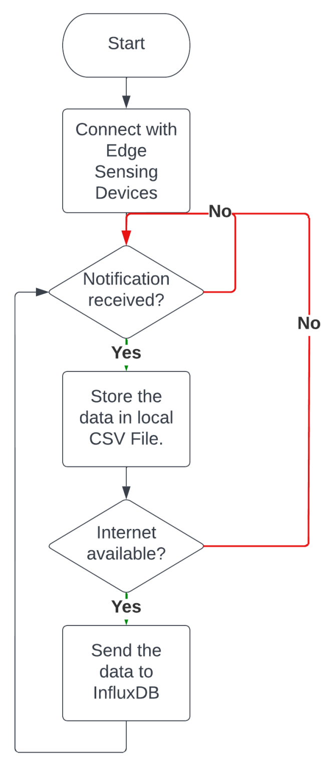


Figure 4.4: Gateway Node

Figure 4.5: Figure on the left shows the code flow for the deployed edge sensing devices. The figure on the right shows the code flow for the Gateway node.

Chapter 5

Challenges

5.1 Bluetooth Disruption Issue

One of the major challenges that we faced during the evaluation of our system was the BLE connection issue between the Edge sensing devices placed inside the hive and the gateway node. During the initial phase of deployment, connectivity issues were not faced because all the frames inside the beehive were empty. But after the queen is added to the beehive, it starts laying eggs inside the hive which turn into larvae and all the worker bees start gathering pollen to feed the larvae and produce honey. This makes the beehive a very dense structure and it is hard for the radio signals to penetrate through the walls and connect to the gateway node. As it can be seen in Figure 4.1, the edge sensing device is placed on the center frame enclosed in a 3-D printed box. The structure of the box is such that there are tiny holes on all the sides of the box. Bees have a tendency of covering any foreign substance inside the beehive with wax. This issue can be tackled by removing the wax accumulated in the tiny holes of the sensor box during honey collection season.

5.2 Internet Disruption Issue

Another problem our system encountered was the internet connectivity issue with the gateway node. As described earlier, the gateway node needs to be connected to the internet in order to send the received data from the data acquisition nodes to the database server. Since the beehive apiary was located in a remote location, it faced internet connectivity issues several times during the deployment. To tackle this challenge, we incorporated saving the received sensor data in a CSV file per sensor. Hence, regardless of internet availability, our system will save the data in a CSV file, which can be recovered remotely by connecting to the gateway node through an SSH connection. Moreover, we have also implemented an algorithm on the gateway node that will send all the stale data (i.e. data that is just stored locally on the gateway node and not on the server) to the database server whenever the internet connection is available.

Chapter 6

Experimental Results

This chapter describes the experimental results of the beehive monitoring system explained in the previous chapter. The experimental system was deployed in May 2022, and it is still operating after 8 months of deployment. The system was deployed in one of the apiaries at UCR with the help of the Department of Entomology, UCR. For experiment purposes, the system was deployed only for one beehive. However, the system can be easily scalable for multiple beehives in the future. The data from the beehive was collected on a data server hosted by the Department of ITS at UCR, and Influxdb services are used for data storage. Additionally, Grafana services were used to display the collected real-time data from the beehive.

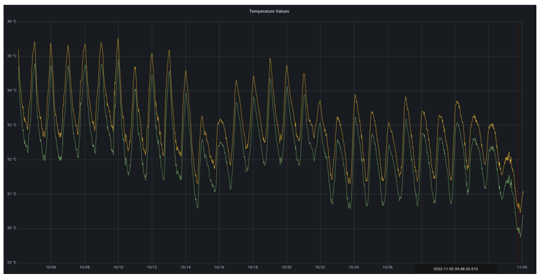
6.1 Hive Microclimate Analysis

In this section, we will present the findings of the different environmental parameters collected during the period of deployment. For simplicity, environmental parameter values for only a 1-month window period (10/03/2022 - 11/03/2022) are shown in this section.

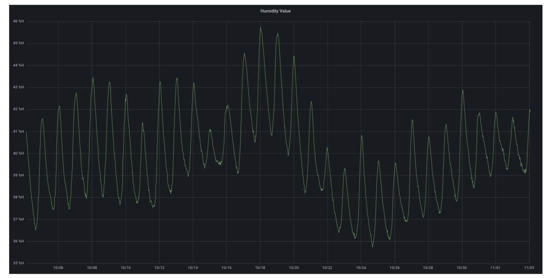
6.1.1 Temperature Values

Figure 6.1(a) shows the graph of temperature values of two different sensors placed at the center frame of the beehive. As can be seen from the figure that the temperature values follow a diurnal pattern with values ranging from 31.0 - 35.0 °C. Regardless of the external temperature values, these values are maintained inside the hive and that indicates that the bees perform the fanning (during the day) and shivering activities (during the night) to keep the temperature values in a suitable range. Typically, the error in temperature values was in the range of ±3.0°C.

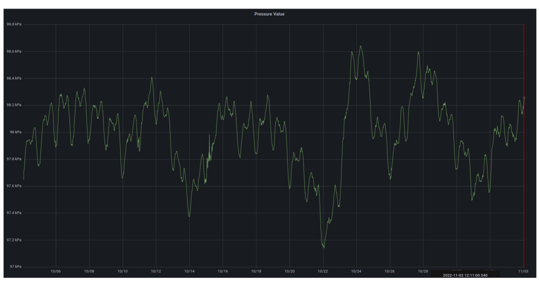
Figure 6.2 shows the graph of temperature values of the 8 peripherals placed on all the 8 corners of the hive and 1 peripheral outside the beehive. The temperature values in all of the corners changed according to ambient temperature. However, for two corners the temperature values were more stable and did not fall below a certain threshold. This can indicate that the brood distribution inside the beehive is such that there are more number of bees present near these two corners as compared to all the other corners. More in-depth analysis in the future can provide more detailed view of brood distribution inside the hive.



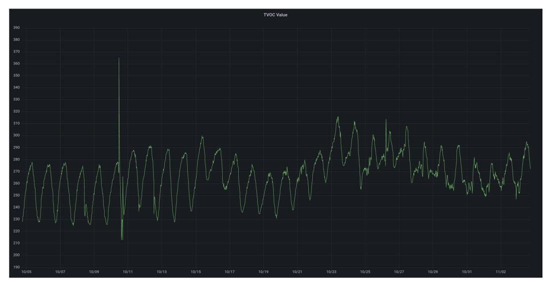
(a) Temperature Values



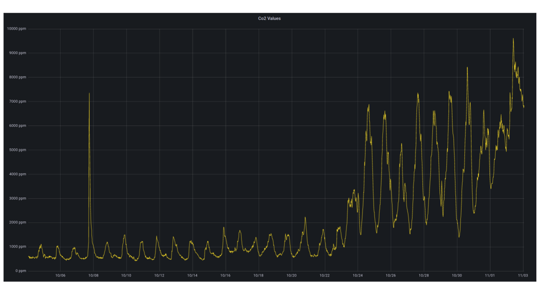
(b) Humidity Values



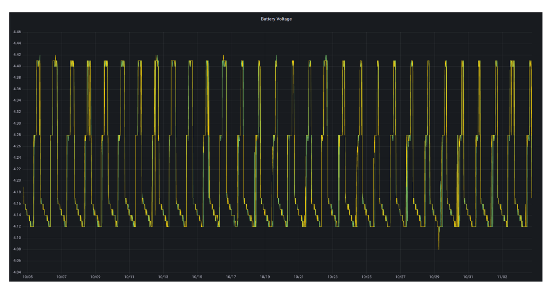
(c) Pressure Values



(D) TVOC Values



(e) CO2 Values



(f) Votlage Values

Figure 6.1: Collected environmental values of the beehive for the consideration window

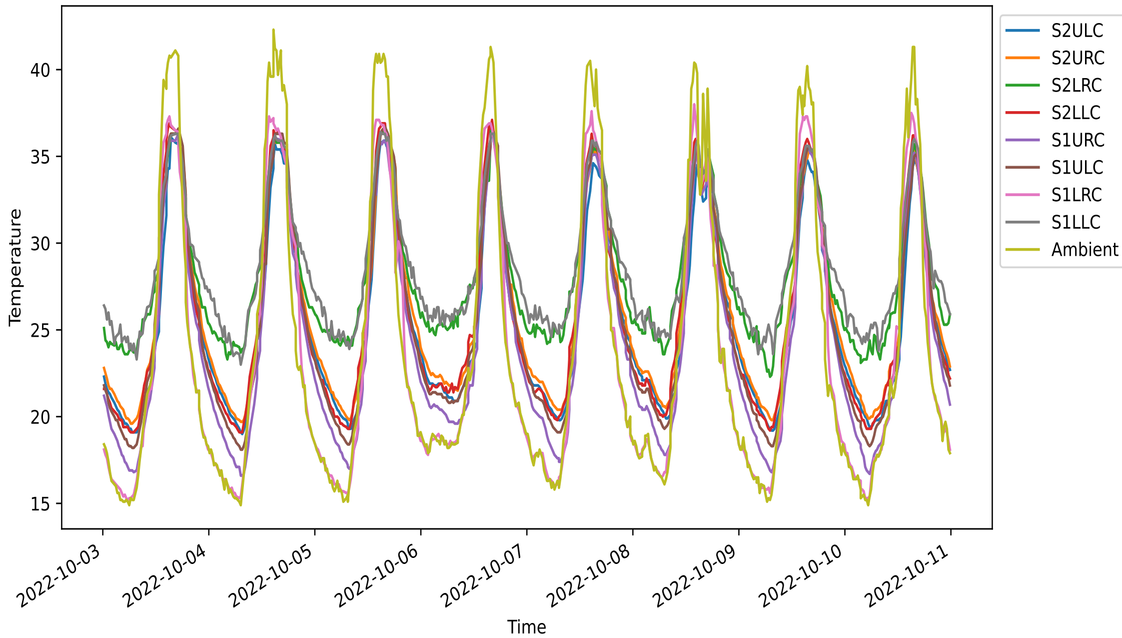


Figure 6.2: Temperature values for all the corners of the hive

6.1.2 Humidity Values

Figure 6.1(b) shows the graph of humidity values of the SHT31 sensor placed at the center frame of the beehive. Similar to temperature values, humidity values also followed a diurnal pattern with values ranging between 48.0 - 57.0 %RH. According to literature [1], a healthy bee colony with a queen normally regulate the humidity values between 40 - 60 %RH. Since, the beehives are located in Riverside, CA where little to no rain is expected during the deployment period, massive fluctuations in the humidity values were not observed.

6.1.3 Carbon Dioxide Values

Carbon Dioxide (CO2) is an essential metric for honey bees. Higher levels of CO2 can indicate that the temperature inside the hive is getting pretty low and bees need to work harder to keep the hive warmer. Previous work [32] shows that CO2 levels inside the hive regulate the fanning behavior inside the hive. As it can be seen from figure 6.1(e), CO2 also follows a diurnal pattern with values ranging between 419 - 9618 Parts Per Million (PPM). During late October, the ambient temperature started to plummet during the night and the CO2 values showed larger variations. This is because bees like to keep a constant temperature inside the beehive and they need to work harder to keep the beehive temperature constant. The high peaks in the CO2 reading perfectly coincide with the low peaks in the outside temperature and hence it explains the higher concentration of CO2 throughout the consideration window.

6.1.4 Voltage Values

The battery voltage also followed a diurnal pattern throughout the deployment period. If the input voltage from the solar panel is higher than the input voltage from the Li-Po battery, the PMIC chip automatically switches the input source to the solar panel. Hence, during the daytime for most of the sunny days, the input voltage values read 4.4 V as it was the cut-off voltage for the PMIC used in the system. Conversely, after sunset hours, the PMIC chip switches the input source to the Li-Po battery and hence the input voltage continuously drops throughout the night. Throughout the period in consideration, the voltage values ranged between 4.11 - 4.4 V. This ensures that the deployed system can remain operational for a very long period of time without requiring any significant maintenance or battery replacement.

6.1.5 Total Volatile Organic Compound Values

Total Volatile Organic Compound (TVOC) is the concentration of various volatile compounds in the atmosphere which can be hazardous for breathing. TVOC is also a major factor in deciding the varroa infection inside the beehive [31]. Figure 6.1(f) shows the TVOC values inside the beehive for the consideration period. The values in the figure are shown in Kilo Ohms (KΩ), and the interpretation of the values is that the lower the value is, the higher the concentration of volatile compounds in the air. Varroa Mites are a major threat to honey bee colonies and constantly monitoring Co2 and TVOC values can help detect and treat hives with varroa infection. Similar to all the climate parameters, TVOC also followed a diurnal pattern with values ranging from 213 - 365 KΩ.

6.2 Energy-aware Control Evaluation

In this section we present the simulation results of the energy-aware feedback loop control system. The PID controller as discussed before generates an output $u(t)$ (in our case change in V_{batt}) and feeds it into the plant as shown in figure 6.3. The plant calculates a suitable period for each task such that the next operating voltage of the battery V_{batt} will be closer to the desired voltage level $V_{desired}$ in the battery. Next we will define the system design in consideration of the simulation results presented in this section.

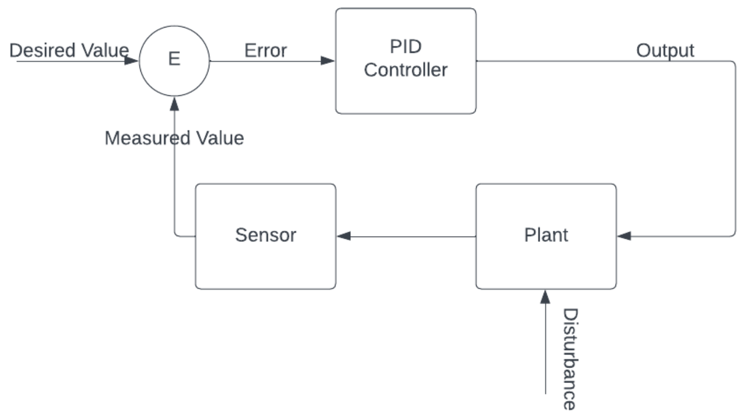


Figure 6.3: Block Diagram of PID Controller

6.2.1 Controller

The PID controller has control parameters such as proportional gain (K_p), integral gain(K_i), derivative gain (K_d), control period ($T_{control}$) and desired voltage($V_{desired}$). The control period of a PID is the time interval after which controller will read the input and generate an output. We consider the control period ($T_{control}$) to be equal to 60 minutes. Each component gain of the PID, affects the responsiveness and these values are extremely important for overall performance of control system. Thus, after careful experimentation the values of the gain were chosen as $K_p = 0.7; K_i = 0.001; Kd = 0.3$. The $V_{desired}$ of the PID controller was set to 4V for our experiments.

6.2.2 Plant

The plant component is the battery and the microcontroller in our feedback control loop system which will be simulated for this experiment. For the battery, we assumed that the average battery operating voltage is $3.7V$ with a capacity of $2.4mAh$, which results

Task	C_i	T_i	Data Metric
τ_1	1	10	Temp. and Humidity
τ_2	2	20	TVOC
τ_3	3	50	CO2

Table 6.1: Sensing Taskset for Energy-aware Feedback Loop Control Algorithm Simulation

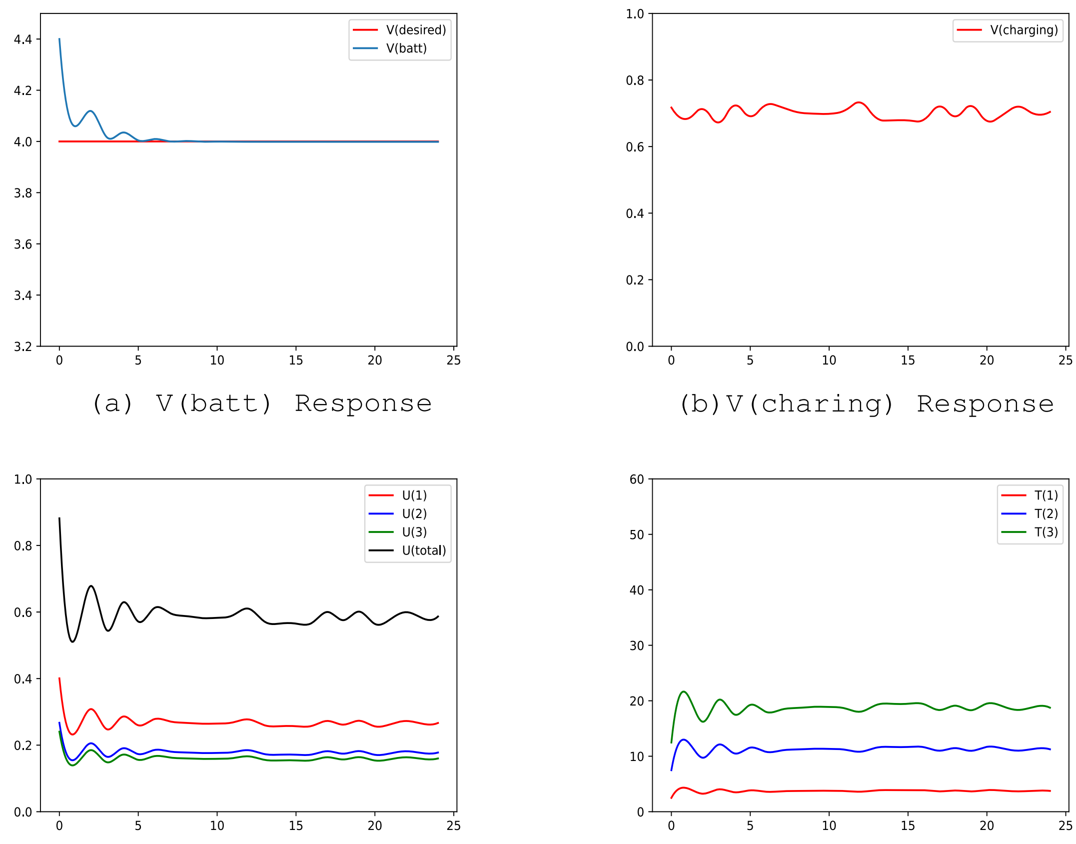
in energy storage capacity of $8.88mWh$. The battery voltage (V_{batt}) ranges from $4.4V$ (at full capacity) to $3.2V$ (at no capacity). For the microcontroller unit, we have taken exact numbers of the Nordic Semiconductors nrf52840 microcontroller which is used in our deployed system. The operating voltage of the microcontroller V_{dd} is $3V$ and the current consumption during the active mode I_{active} is $3mA$ and during sleep mode I_{sleep} is $17\mu A$. Here, during a control period the processor can either be in active state or in sleep state. Hence, the summation of time spent by the microcontroller in active state and sleep state should be equal to the control period ($T_{control} = T_{active} + T_{sleep}$).

We assume 3 sensing tasks for this experiment, that are responsible for acquiring temperature, humidity, carbon dioxide (CO2) and total volatile organic compound (TVOC). Each task has an assigned Worst-Case Execution Time (C) and nominal period(T_{nom}) initially as shown in table 6.1. Utilization(U) of a task is the fraction of processor time spent executing the task ($U_i = C_i/T_i$) [8]. For the task set in consideration, nominal utilization $U_{nominal}$ of the system is 0.22. Since data freshness is extremely important for beehive monitoring, the utilization of the microcontroller must have a lower bound. This is because, the system will eventually go to sleep to keep the battery voltage closer to desired value when there is little to no charging offered by the solar panel (during nights). Thus, we considered that each of the tasks needs to be scheduled at least once during the control period (60 minutes) and hence, the lowest utilization required U_{lowest} is 0.1 ($U_{lowest} = T_{active}/T_{control} = 6/60$).

$$V_{t+1} = V_t + V_{charge} - V_{discharge} \equiv V_t \pm u(t) \tag{6.1}$$

$$[\%E] = [P_{active} * T_{active} + P_{sleep} * T_{sleep}] = V_{batt} * [I_{active} * T_{active} + I_{sleep} * (T_{control} - T_{active})] \tag{6.2}$$

The charging voltage V_{charge} from the energy harvesting system is considered to be constant for this experiment and a 5% error is introduced to simulate real scenario. The minimum discharge voltage V_{dmin} that can be consumed by the system is bounded by U_{lowest} which is 0.12 V. The next battery voltage after a control period directly depends on the output of the PID controller as defined in equation 3.2. Since, we assumed energy storage and voltage of the battery to be proportional to each other, we can obtain the total utilization of the system for a $V_{discharge}$ value by calculating the time that the microcontroller should spend in active state T_{active} from equation 6.2.2. Change in T_{active} as compared to nominal active time of the microcontroller can easily be calculated and the period values of all the tasks can be updated by this change. Hence, in this way, the feedback loop control system calculats the optimal sensor period values from the change in energy storgae of the battery required.



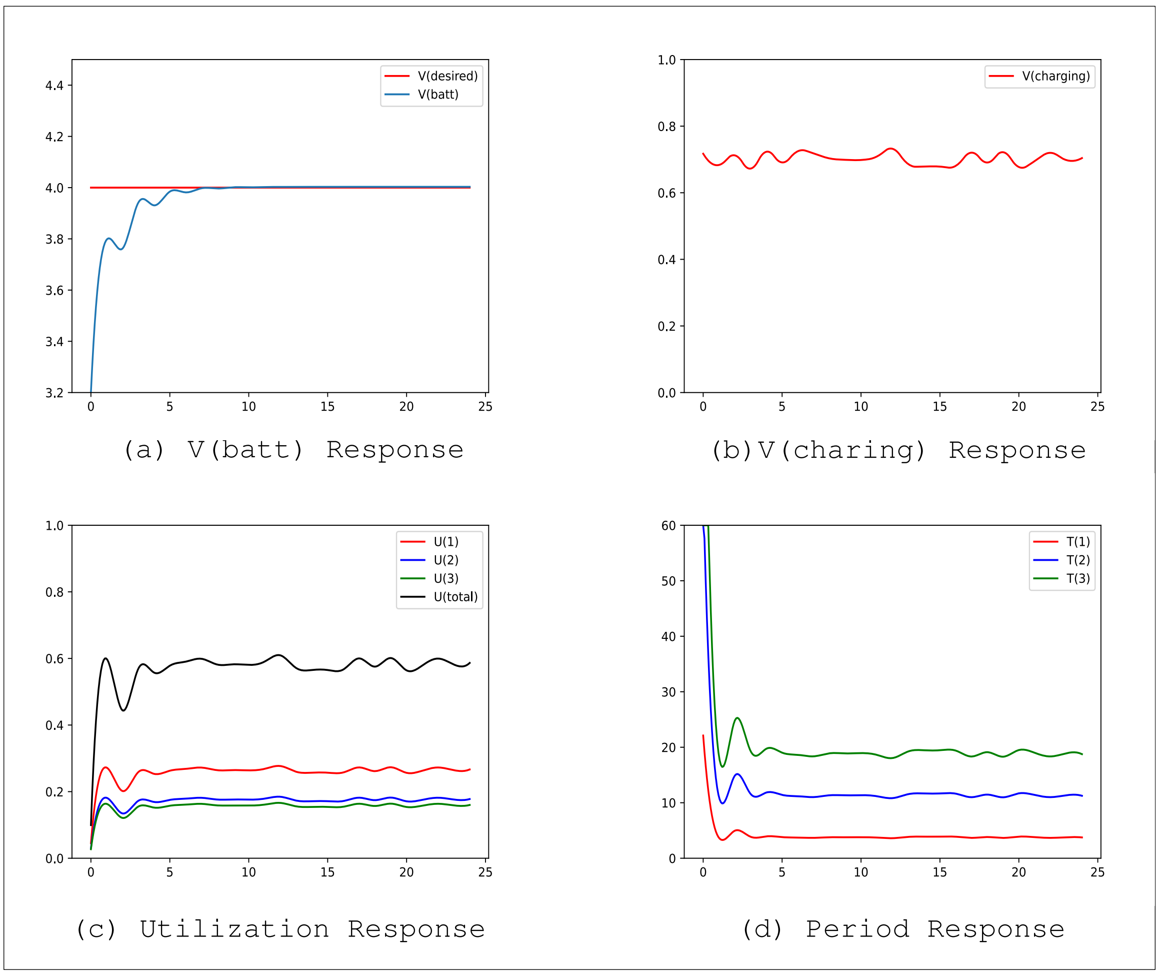
(a) V(batt) Response (b)V(charing) Response

(c) Utilization Response (d) Period Response

Figure 6.4: Control System Response at $V_{init} = 4.4V$

6.2.3 Results

Figure 6.4 shows the control system response with initial voltage V_{init} equal to 4.4V. Similarly, Figure 6.5 shows the simulation output of the system with initial voltage V_{init} equal to 3.2V. The simulation shows that the battery voltage V_{batt} response for 24 hours with constant charging voltage and it can be seen from the figure 6.4(a) & 6.5(a) that the battery voltage V_{batt} oscillates initially and then slowly converges into $V_{desired}$.



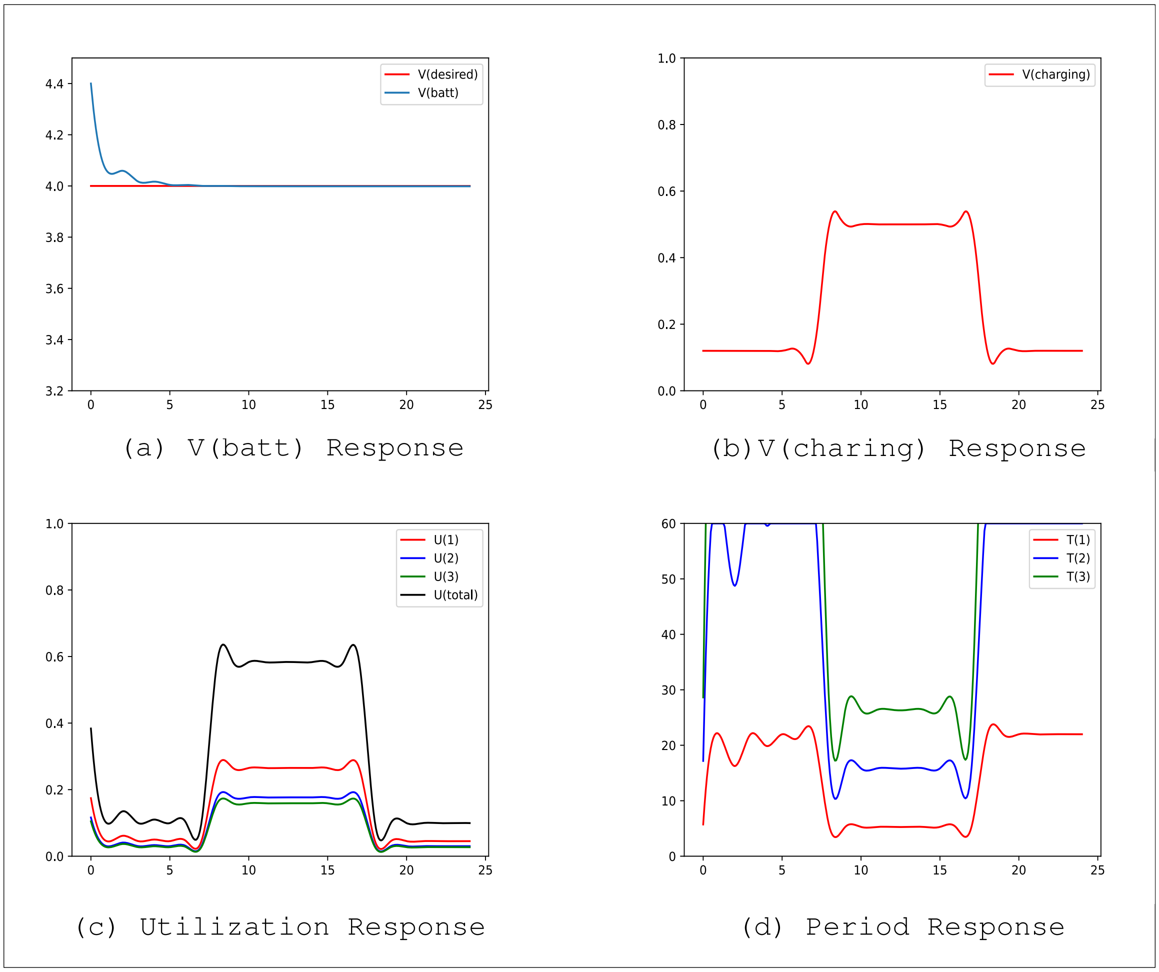
(a) V(batt) Response

(b) V(charing) Response

(c) Utilization Response

(d) Period Response

Figure 6.5: Control System Response at $V_{init} = 3.2V$



(a) V(batt) Response

(b) V(charing) Response

(c) Utilization Response

(d) Period Response

Figure 6.6: Control System Response with V(charge) simulating typical day

Also, the response of the task periods and utilization changes linearly with the V_{batt} until it converge, and after that the response varies depending on the V_{charge} to stay in equilibrium at ($V_{batt} = V_{desired}$).

In order to emphasize the need for sensor period control system, we show the results for charging voltage V_{charge} changing drastically throughout the day as it is observed in a typical day (high values during the day and negligible values during the night). Figure

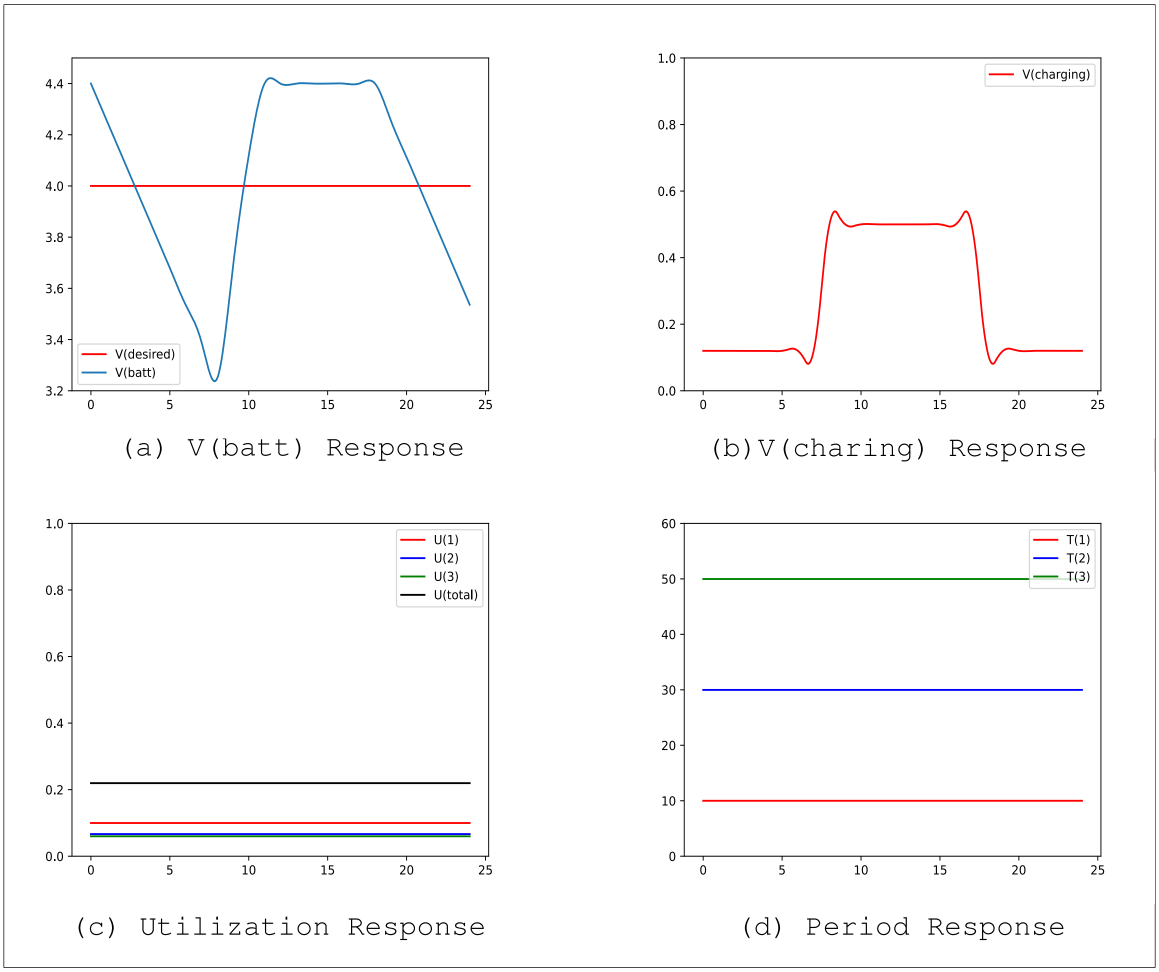


Figure 6.7: Conventional(static period) System Response with V(charge) simulating typical day

6.6[a-d] shows the response graphs for V_{batt}, V_{charge}, U_i and T_i for the feedback loop control system. It can be seen from the graphs, that the utilization and period of each task adapts to change in V_{charge} very rapidly and keeps the V_{batt} equal to $V_{desired}$. For comparison, figure 6.7 [a-d] shows response graphs for same metrics for the conventional systems with static period and it can be seen that the battery voltage drops linearly with time when the V_{charge} is at its minimum value. Again, as V_{charge} increases, V_{batt} increases linearly with it and reaches the maximum value meaning full battery storage. However, this excess of energy is wasted by conventional systems because the energy demand of the system does not change with respect to energy availability. The average overall utilization for the proposed system was found to be 0.3 in this case and that for the conventional system was 0.22. Thus, our proposed system increased the utilization of the system by 36% as compared to conventional systems while managing to keep the battery voltage to the desired voltage level. Hence, this proves that the energy-aware feedback loop control system utilizes the available energy more effectively.

6.3 Case Study for Rate Harmonized Scheduling

In this section, we compare the overall power consumption in Rate Harmonized Scheduling (RHS) and Rate Monotonic Scheduling (RMS) policies. For the comparison, we considered another example taskset with multiple producer threads and a single consumer thread from the producer-consumer model that we explained earlier in the thesis. Each task in this study can be characterized as follows:

Task	C_i	T_i	Task Description
τ_1	1	5	Producer Task
τ_2	1	7	Producer Task
τ_3	2	9	Consumer Task

Table 6.2: Example Taskset for Case Study

$$\tau_j = < T_j, C_j, P_j >,$$

$where,$

T_j : Period of task τ_j,

C_j : Worst Case Execution Time (WCET) of task τ_j,

P_j : Priority of task τ_j,

In Table 6.2, we consider τ_1 and τ_2 to be producer tasks and τ_3 to be consumer task. We assumed that all the tasks in the taskset follows implicit deadline ($T_i = D_i$, where D is deadline of a task). Priority of all the tasks in the taskset are given using Rate Monotic policy for priority assignment (i.e. task priority is inversely proportional to task period). Hence, priority assignment for our example taskset will be $P_1 > P_2 > P_3$. C_{sleep} is the time taken by the system to switch to the deep sleep mode. If any task arrives in less than C_{sleep} time, system cannot switch to sleep mode and have to wait in idle until the task arrives. C_{sleep} for our taskset is assumed to be 2 time units. Lastly, since we are considering the basic RHS policy, we took harmonizing period ($T_H = T_1 = 5$) of all the tasks to be equal to period of the highest priority task τ_1.

Scheduling Policy	Energy Consumption (μWT)
RHS	2322
RMS	3018

Table 6.3: Energy Consumption for RHS and RMS

As shown in figure 6.8, the Rate Monotonic Scheduling system spends 18-time units in sleep mode and 7-time units in idle mode. On the other hand, the Rate Harmonized Scheduling system spends 22-time units in sleep mode and 4-time units in idle mode. The assumed current consumption of the microntroller for sleep mode (I_{sleep}) is 17 μA and for idle mode (I_{idle}) is 100 μA. Even this small difference between I_{sleep} and I_{idle} can cause severe degradation to the system's energy response when it is accumulated over longer periods.

Table 6.3 shows energy consumption for RMS and RHS scheduling and it can be seen that RHS utilizes significantly less amount of energy as compared to RMS. Hence, keeping the system in sleep mode for a longer period by coalescing the execution of threads can be very much beneficial for the longevity of a beehive monitoring system.

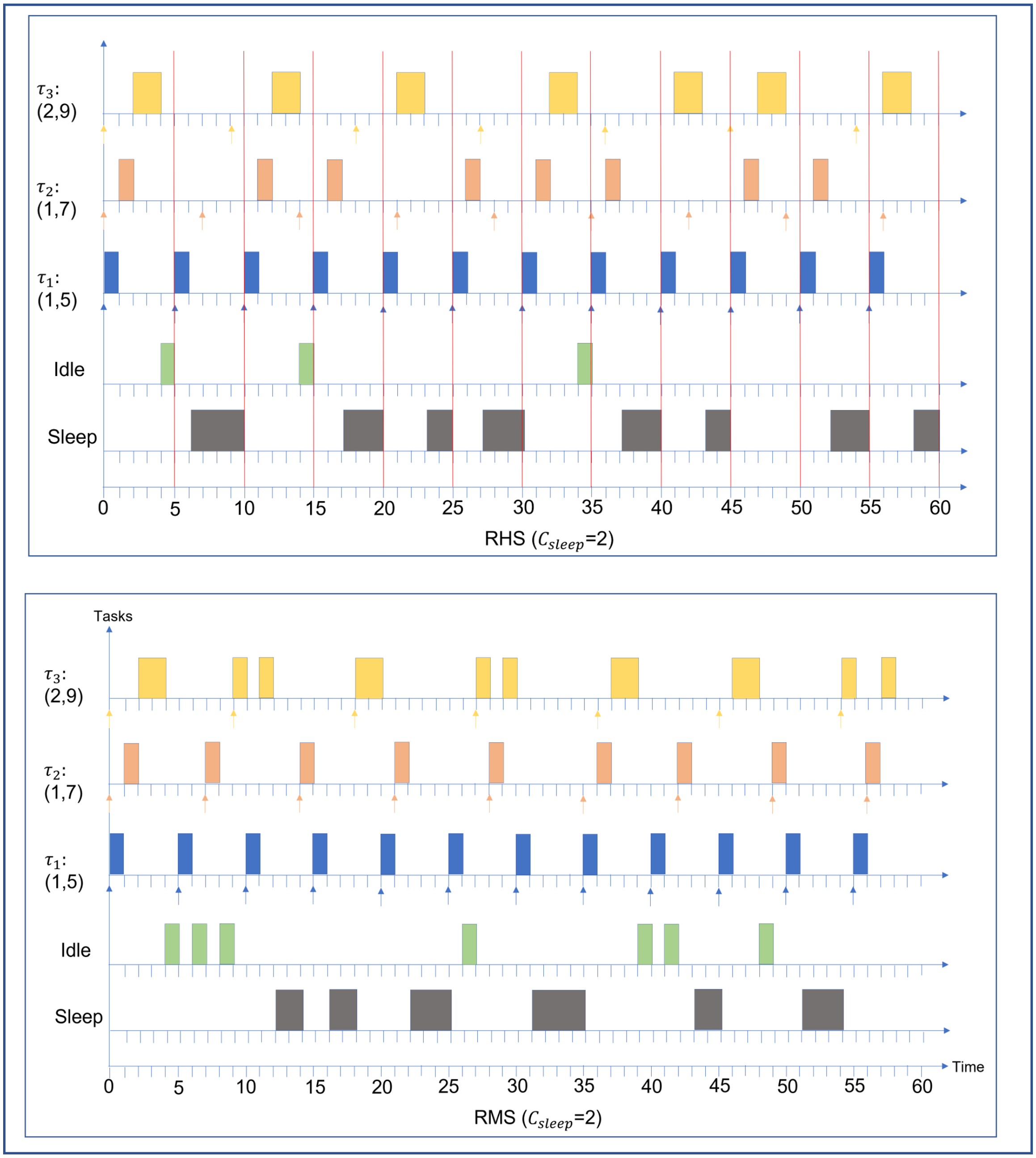


Figure 6.8: Example Taskset from Table 6.2. In top, RHS is shown with $T_H = 5$ marked in red vertical bars and RMS is shown in bottom.

Chapter 7

Conclusions

In this thesis, we introduce a smart and sustainable beehive monitoring system that monitors the beehive continuously for a longer period of time without requiring a battery change. A feedback loop control system was proposed to dynamically change the periods of the sensing tasks to ensure the responsiveness of the system to the input voltage. Results showed that such a control system can outperform the conventional beehive monitoring systems by keeping the battery voltage constant over time while increasing the utilization of the system. Lastly, a naive scheduler was also proposed to utilize Rate Harmonized Scheduler to achieve maximum energy saving.

The proposed beehive monitoring system was deployed in-field for more than 8 months and results were present for a 1-month period. During the deployment period, the system continuously delivered micro-climatic parameters without requiring any maintenance. This shows the usefulness of the system in a real-world deployment.

7.1 Future Work

Currently, the system is only designed to collect environmental parameters from the hive and store them on a cloud-based server. This provides a platform for researchers to study the micro-climate inside the beehive while correlating it with the ambient climate. However, the system can be further extended to incorporate deep learning algorithms for time-series sensor data [7] to achieve an intelligent system that can detect abnormal beehive status in the real-world and alert beekeepers of potential threats.

Much research has been already done to completely eliminate the use of batteries and make use of intermittently-powered devices [19, 36, 17, 18]. Such approaches can also be incorporated into the existing system to harvest energy from vibrations inside the hive or from RF signals and make the system battery-less.

Bibliography

[1] Arnia remote hive monitoring. https://www.arnia.co/post/humidity-in-the-hive, 2022. accessed: November 2022.

[2] United states department of agriculture. https://www.usda.gov/peoples-garden/pollinators/honey-bees, 2022. accessed: November 2022.

[3] Cesare Alippi, Romolo Camplani, Cristian Galperti, and Manuel Roveri. A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring. *IEEE Sensors Journal*, 11(1):45–55, 2010.

[4] Doreid Ammar, Jean Savinien, and Lionel Radisson. The makers' beehives: Smart beehives for monitoring honey-bees' activities. In *Proceedings of the 9th International Conference on the Internet of Things*, pages 1–4, 2019.

[5] NIRMAL ANAND, V BHARATH RAJ, MS Ullas, and ASHUTOSH SRIVASTAVA. Swarm detection and beehive monitoring system using auditory and microclimatic analysis. In *2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C)*, pages 1–4. IEEE, 2018.

[6] Rassol Bahreini and Robert W. Currie. The potential of bee-generated carbon dioxide for control of varroa mite (mesostigmata: Varroidae)in indoor overwintering honey bee (hymenoptera: Apidae) colonies. *Journal of Economic Entomology*, 108(5):2153–2167, 2015.

[7] Abdulrahman Bukhari, Seyedmehdi Hosseinimotlagh, and Hyoseung Kim. An open-world time-series sensing framework for embedded edge devices. In *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 61–70. IEEE, 2022.

[8] Giorgio C Buttazzo. *Hard real-time computing systems: predictable scheduling algorithms and applications*, volume 24. Springer Science & Business Media, 2011.

[9] Stefania Cecchi, Susanna Spinsante, Alessandro Terenzi, and Simone Orcioni. A smart sensor-based measurement system for advanced bee hive monitoring. *Sensors (Switzerland)*, 20, 2020.

[10] Sandeep DSouza, Anand Bhat, and Ragunathan Rajkumar. Sleep scheduling for energy-savings in multi-core processors. In *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 226–236. IEEE, 2016.

[11] Fiona Edwards-Murphy, Michele Magno, Pádraig M. Whelan, John O'Halloran, and Emanuel M. Popovici. B+wsn: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture*, 124, 2016.

[12] Fiona Edwards-Murphy, Emanuel Popovici, Padraig Whelan, and Michele Magno. Development of an heterogeneous wireless sensor network for instrumentation and analysis of beehives. volume 2015-July, 2015.

[13] James D Ellis, Jay D Evans, and Jeff Pettis. Colony losses, managed colony population decline, and colony collapse disorder in the united states. *Journal of Apicultural Research*, 49(1):134–136, 2010.

[14] Agbotiname Lucky Imoize, Samuel Damilola Odeyemi, and John Adetunji Adebisi. Development of a low-cost wireless bee-hive temperature and sound monitoring system. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 8(3):476–485, 2020.

[15] Hasila Jarimi, Emmanuel Tapia-Brito, and Saffa Riffat. A review on thermoregulation techniques in honey bees'(apis mellifera) beehive microclimate and its similarities to the heating and cooling management in buildings. *Future Cities and Environment*, 6(1), 2020.

[16] Joe-Air Jiang, Chien-Hao Wang, Chi-Hui Chen, Min-Sheng Liao, Yu-Li Su, Wei-Sheng Chen, Chien-Peng Huang, En-Cheng Yang, and Cheng-Long Chuang. A wsn-based automatic monitoring system for the foraging behavior of honey bees and environmental factors of beehives. *Computers and Electronics in Agriculture*, 123:304–318, 2016.

[17] Mohsen Karimi, Hyunjong Choi, Yidi Wang, Yecheng Xiang, and Hyoseung Kim. Real-time task scheduling on intermittently powered batteryless devices. *IEEE Internet of Things Journal*, 8(17):13328–13342, 2021.

[18] Mohsen Karimi and Hyoseung Kim. Energy scheduling for task execution on intermittently-powered devices. *ACM SIGBED Review*, 17(1):36–41, 2020.

[19] Mohsen Karimi, Yidi Wang, and Hyoseung Kim. Energy-adaptive real-time sensing for batteryless devices. In *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 205–211. IEEE, 2022.

[20] Dimitrios I Kiromitis, Christos V Bellos, Konstantinos A Stefanou, Georgios S Stergios, Thomas Katsantas, and Sotirios Kontogiannis. Bee sound detector: An easy-to-install, low-power, low-cost beehive conditions monitoring system. *Electronics*, 11(19):3152, 2022.

[21] Douglas S. Kridi, Carlos Giovanni N. de Carvalho, and Danielo G. Gomes. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Computers and Electronics in Agriculture*, 127, 2016.

[22] Douglas S Kridi, Carlos Giovanni N de Carvalho, and Danielo G Gomes. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Computers and Electronics in Agriculture*, 127:221–235, 2016.

[23] Young-Kyu Lee and Hyun-Joong Kim. The effect of temperature on vocs and carbonyl compounds emission from wooden flooring by thermal extractor test method. *Building and Environment*, 53:95–99, 2012.

[24] William G. Meikle, Aidan Barg, and Milagra Weiss. Honey bee colonies maintain co2 and temperature regimes in spite of change in hive ventilation characteristics. *Apidologie*, 53(5), 2022.

[25] Fiona Edwards Murphy, Michele Magno, Liam O'Leary, Killian Troy, Pádraig Whelan, and Emanuel M Popovici. Big brother for bees (3b)—energy neutral platform for remote monitoring of beehive imagery and sound. In *2015 6th International Workshop on Advances in Sensors and Interfaces (IWASI)*, pages 106–111. IEEE, 2015.

[26] Fiona Edwards Murphy, Michele Magno, Padraig Whelan, and Emanuel Popo Vici. b+ wsn: Smart beehive for agriculture, environmental, and honey bee health monitoring—preliminary results and analysis. In *2015 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2015.

[27] Elias Ntawuzumunsi, Santhi Kumaran, and Louis Sibomana. Self-powered smart beehive monitoring and control system (sbmacs). *Sensors*, 21(10):3522, 2021.

[28] V Pandimurugan, Rahul Mandviya, Abhinav Gadgil, Kumar Prakhar, and Aditya Datar. Iot based smart beekeeping monitoring system for beekeepers in india. In *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, pages 65–70. IEEE, 2021.

[29] Anthony Rowe, Karthik Lakshmanan, Haifeng Zhu, and Ragunathan Rajkumar. Rate-harmonized scheduling for saving energy. In *2008 Real-Time Systems Symposium*, pages 113–122. IEEE, 2008.

[30] Sebasttian Sadowski and Petros Spachos. Solar-powered smart agricultural monitoring system using internet of things devices. 2019.

[31] Thomas Schweizer, Diren Senger, Carolin Johannsen, Thorsten Kluss, and Michael J Vellekoop. Supervising varroa treatment of honey bees with smart sensor systems. In *MikroSystemTechnik Congress 2021; Congress*, pages 1–4. VDE, 2021.

[32] Thomas D Seeley. Atmospheric carbon dioxide regulation in honey-bee (apis mellifera) colonies. *Journal of insect physiology*, 20(11):2301–2305, 1974.

[33] Thomas D Seeley, Marco Kleinhenz, Brigitte Bujok, and Jürgen Tautz. Thorough warm-up before take-off in honey bee swarms. *Naturwissenschaften*, 90(6):256–260, 2003.

[34] Chau-Chung Songa, Geng-Yi Lina, Chi-Chung Pengc, and Chung-Wen Hungd. Development of intelligent beehive and network monitoring system for bee ecology. 2022.

[35] Alessandro Terenzi, Stefania Cecchi, Simone Orcioni, and Francesco Piazza. Features extraction applied to the analysis of the sounds emitted by honey bees in a beehive. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 03–08. IEEE, 2019.

[36] Kacper Wardega, Wenchao Li, Hyoseung Kim, Yawen Wu, Zhenge Jia, and Jingtong Hu. Opportunistic communication with latency guarantees for intermittently-powered devices. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1093–1098. IEEE, 2022.

[37] Geoffrey R Williams, David R Tarpy, Marie-Pierre Chauzat, Diana L Cox-Foster, Keith S Delaplane, Peter Neumann, Jeffery S Pettis, Richard EL Rogers, Dave Shutler, et al. Colony collapse disorder in context. *Bioessays*, 32(10):845, 2010.

[38] Aleksejs Zacepins and Valters Brusbardis. Precision beekeeping (precision apiculture): research needs and status in latvia. *Poljoprivreda i Sumarstvo*, 61(1):135, 2015.