

Precision apiculture: development of a wireless sensor network for honeybee hives

Evan Henry

Masters of Science

Department of Bioresource Engineering

Macdonald Campus, McGill University

Montreal, Quebec, Canada

March 2016

A thesis submitted to McGill University in partial fulfillment of the requirements of
the degree of Master of Science

©Evan Henry, 2016

ACKNOWLEDGEMENTS

Dr. Adamchuk, thank you for being a great supervisor. Working under your supervision was inspiring, rewarding, and also a whole lot of work.

I'd like to acknowledge everyone from the Precision Agriculture and Sensor System research group. Working alongside you all has been real and a great experience.

Dr. Buddle, thank you for your support and guidance as a committee member.

Dr. Dhawale, thank you for carefully editing my thesis.

Ms. Hitti, thank you for translating the abstract into French.

To everyone involved in the McGill Apicultural Association, especially Bransilav, thank you.

Last but not least, I would like to thank my friends, and most importantly my family, for all the support, honeybee themed stuff, and support.

PREFACE & CONTRIBUTION OF AUTHORS

The research presented in this thesis has been submitted for publication in *Computers and Electronics in Agriculture*, a peer reviewed journal. The author of this thesis carried out the research experiments, and co-developed the associated software with Trevor Stanhope. Dr. Adamchuk, the author's supervisor, designed the methodology. It was his idea to investigate the effects of electromagnetic radiation on beehives.

PUBLICATIONS RELATED TO THE THESIS:

1. **Henry, E.**, Adamchuk, V., Stanhope, T., Buddle, C. 2016. Precision apiculture: development of a wireless beehive sensor network. *Computers and Electronics in Agriculture* (under review).

ABSTRACT

Wireless in-hive sensor networks show promise in apiary management and research. However, radio frequency electromagnetic radiation (RF-EMR) emitted by wireless technologies could affect honeybees at the individual and the colony level. Prior research has noted that different operation frequencies and power of transferred signals affect insect and larger animal behavior. An on-line wireless sensor network was developed that continuously monitored in-hive temperature, relative humidity, and acoustics. While testing the network performance, a wired version of the developed sensor network was used to substantiate whether or not RF-EMR from Wi-Fi affects honeybee hives through a 30-day study in 2015. Two groups of three beehives were monitored: the first group was subjected to 2.4 GHz Wi-Fi signal while the second group was located outside of radio communication range. During RF-EMR exposure, in-hive temperature increased by an average of 0.09 °C and relative humidity increased by 1.53 %, sound increased in amplitude by 0.03 dB but decreased in frequency by 2.57 Hz. All measured parameters had a higher standard deviation (SD) during the exposure treatment as compared to these changes during RF-EMR exposure. This means that the findings were not significant at the $p < 0.05$ level, Control Hive 1 was on average 14.4 % less humid and 1.57 dB quieter than the other monitored hives and had a 41.5 % *Varroa destructor* infestation rate, 23.2 % higher than the average infestation rate (18.3 %). Based on this research, no evidence of beehive environment change in response to RF-EMR was found. At the same time, the observation of abnormally measured parameters could be linked with a stressed colony (increased

Varroa destructor infestation rate), which might help apiculturists reduce production losses by rapidly reacting to the observed indicators of potential stress. Given the complexity of beehive dynamics, research on other potential effects of RF-EMR is needed before adopting wireless technologies in beehive sensors.

RÉSUMÉ

Les réseaux de capteurs sans fil dans les ruches sont prometteurs dans la gestion des ruchers et en recherche. Cependant, ces technologies sans fil pourraient affectées les abeilles au niveau individuel ainsi que la colonie au complet dues aux émissions de rayonnement électromagnétique de fréquence radio (ERE-FR). Des recherches antérieures sur des fréquences différentes et la puissance des signaux ont noté un effet sur les insectes et le comportement animal. Un réseau en ligne de capteurs sans fil a été développé pour faire un suivi continu sur la température, l'humidité relative et les acoustiques des ruches. En testant les performances du réseau, une autre version des capteurs a été effectuée, cette fois-ci avec câbles. Celles-ci ont été utilisées pour valider si les ERE-FR à partir de connexion Wi-Fi affecte les ruches d'abeilles dans une période de 30 jours (été 2015). Deux groupes de trois ruches ont été suivis: le premier groupe a été soumis à 2.4 GHz de signal Wi-Fi tandis que le deuxième groupe était situé en dehors de la plage de communication radio. Lors de l'exposition de ERE-FR, des changements à l'intérieure des ruches ont été notés tel qu'une augmentation de température moyenne de 0,09 °C, une augmentation de l'humidité relative de 1,53 %, une augmentation de son en amplitude de 0,03 dB, mais une diminution de fréquence de 2,57 Hz. Tous les paramètres mesurés ont un plus grand écart type (ÉT) au cours du traitement d'exposition par rapport à ces changements. Ceci veut donc dire que les résultats n'ont pas été significatifs au niveau $p < 0,05$, la ruche control 1 était en moyenne 14.4 % moins humide et 1.57 dB plus silencieuse en comparaison aux autres ruches surveillées et avait un taux d'infestation de 41,5 %

de *Varroa destructor*, ceci tant 23,2 % supérieur à la moyenne des taux d'infection (18,3 %). Basé sur les résultats de cette recherche, aucune preuve de changement dans l'environnement des ruches en réponse aux ERE-FR a été trouvée. L'observation des mesures anormales des paramètres pourrait souligner une colonie stressée (augmentation d'infection de *Varroa destructor*) ce qui pourrait servir comme outil aux apiculteurs pour réduire le nombre de pertes et réagir en temps lors des indicateurs observés. Étant donné la complexité des dynamiques d'une ruche, plus de recherche sur les effets d'ERE-FR est nécessaire avant l'adoption des technologies sans fil dans les ruches.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
PREFACE & CONTRIBUTION OF AUTHORS	iii
ABSTRACT	iv
RÉSUMÉ	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
1 Introduction	1
1.1 Overview	1
1.2 Justification	1
1.3 Research objectives	1
1.4 Thesis format	2
2 Literature review	3
2.1 Importance of honeybees in agriculture	3
2.2 Relevant aspects of honeybee biology to in-hive sensor systems	4
2.3 Beehive sensor systems	6
2.3.1 Academic	7
2.3.2 Commercial beehive sensors	9
2.4 Review of Effects of Radio Frequency Electromagnetic Radiation on Animals	10
3 Wireless sensor network development	12
3.1 Sensor hardware	12
3.2 Data acquisition and processing	14
3.3 Data Storage	16
3.3.1 Data retrieval	17

3.4	Wireless sensor network	17
3.4.1	Data visualization and web application	18
4	A study on the effects of Wi-Fi communication on measured sensor parameters inside beehive colonies	20
4.1	Experimental design	20
4.1.1	Sensor network configuration	20
4.1.2	Sensor installation	21
4.2	Wi-Fi electromagnetic radiation	22
4.3	Data analysis	23
5	Results and discussion	25
5.1	Results	25
5.2	Discussion	33
5.2.1	Research experiment	33
5.3	Design Considerations	36
5.3.1	Sensor performance and future improvements	36
5.3.2	Apiary economy and practicality	39
6	Conclusion	41
7	Appendix A	48
8	Appendix B	49
9	Appendix C	62

LIST OF TABLES

<u>Table</u>	<u>page</u>
3-1 Cost of the hive sensor hardware components	14
5-1 Summarized climatic data presented as mean \pm SD.	28
5-2 Summarized acoustic data presented as mean \pm SD.	29
5-3 Results of RF-EMR analysis following Equation 4.1 for in-hive climatic conditions.	30
5-4 Results of RF-EMR analysis following Equation 4.1 for in-hive acoustic conditions	31
5-5 Correlation (r^2) between in-hive temperature and relative humidity with external conditions.	32
5-6 Grand mean and SD of the nine hive-pair interactions during RF-EMR exposure and results of the t-test	32
5-7 Grand mean and SD of the nine hive-pair interactions during RF-EMR exposure and results of the t-test	32
5-8 Sampling rate discrepancies between sensors.	37
7-1 Sample data in .csv format	48

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
3-1 RaspberryPi 2 model B V1.1.	13
3-2 Top and bottom views of the wired AM2303 Sensor.	13
3-3 PLMC15 omnidirectional microphone.	14
3-4 A web application displays hive sensor data.	19
4-1 A schematic diagram of the experimental design.	21
4-2 A Langstroth hive diagram (Blackiston, 2012).	21
4-3 Hive sensor installed in the inside of beehive roofs.	22
4-4 An interpolated map of 2.4 GHz signal strength when Wi-Fi was off (left) and on (right) at the Macdonald Campus Apiary, Ste. Anne-de-Bellevue, Québec.	23
5-1 Per hour averaged in-hive climatic data.	27
5-2 Per hour averaged in-hive acoustic data.	28
5-3 Results of RF-EMR analysis following Equation 4.1 for in-hive climatic conditions.	29
5-4 Results of RF-EMR analysis following Equation 4.1 for in-hive acoustic conditions.	30
5-5 Varroa infestation rate.	31
5-6 A hive sensor powered by a solar panel, tested in 2014 at the Macdonald Campus apiary	38

CHAPTER 1

Introduction

1.1 Overview

Honeybees are facing an unprecedented decline in population without a clear understanding of the cause. This phenomenon is called colony collapse disorder (CCD), and it has severe agricultural, economic, and ecological implications. One shortcoming in researching and managing honeybee colonies lies in inadequate data collection. Sensor systems that continuously monitor in-hive characteristics could facilitate new research and apiary management techniques.

1.2 Justification

Utilizing wireless technologies in hive sensors would enable remote access to hive data, but the radio frequency electromagnetic radiation (RF-EMR) from wireless technologies might have ramifications on honeybees. Prior research noted that different operating frequencies and power of transferred signals produce different levels of disturbance that affect insect and larger animal behavior ([Cucurachi et al., 2013](#)). Before adopting wireless technologies in beehive sensor networks, evaluation of such technologies is needed.

1.3 Research objectives

The overall aim of this research is twofold: one goal is to develop a wireless sensor network for on-line monitoring of beehive microclimate parameters, including: temperature, relative humidity, and sound that could improve apicultural management

and research methods. The second goal of is to substantiate whether RF-EMR from Wi-Fi affects the stated in-hive measurements by using a wired version of the developed sensor network.

1.4 Thesis format

Chapter two is a literature review with four parts. Part one details the importance of honeybees in agriculture and the recent honeybee population decline. Part two explains why honeybee hive sensing is advantageous to apiculture based on honeybee biology. Part three reviews the relevant hive sensor technologies developed in academia that are commercially available. Part four reviews the effects of RF-EMR on honeybees and related animals. Chapter three describes the materials and methods of both the developed hive sensor software and hardware and the research experiment investigating the effects of RF-EMR. Chapter four presents the results of the hive RF-EMR experiment. Chapter five is a discussion of the results of the sensors and RF-EMR research. Chapter six contains the conclusion, followed by a bibliography, and three appendices that contain sample data, the hive sensor software, and the hive aggregator software.

CHAPTER 2

Literature review

2.1 Importance of honeybees in agriculture

Although the honeybee is commonly affiliated with honey production, the role of honeybees in agriculture is predominately for pollination, enabling fruit growth. Pollination is the fertilization step of the reproduction process in spermatophytes, or seed bearing plants. Pollen contains the male gametophyte cells. When transferred to a flower's stigma, the male pollen grains germinate, reach the female gamete cells, and this process enables reproduction and fruit development. Wind, along with birds, insects, and other animals act as pollen vectors.

Pollination plays a vital role in agriculture because it enables fruit development. In 2010 in the United States of America, insect mediated crop pollination was valued at \$29 billion USD, and of that \$19.2 billion was a result of pollination by honeybees (Calderone, 2012). Even though there are about 5,000 species of native bees, the introduced European honeybee, *Apis mellifera*, is the most economically important pollinator in North America. Unlike the majority of native bees, honeybees are eusocial insects that live in colonies exceeding 50,000 individuals. As such, beehives have the capacity to pollinate more flowers, and are domesticated for their mass pollination abilities.

There has been a rapid increase of pollinator-dependent crops in North America and a decrease in managed honeybees; this jeopardizes pollination rates and in turn,

agricultural productivity. In the United States, there was a fourfold increase in crops that require animal pollination between 1960 and 2000, (Aizen and Harder, 2009). Also, there are only 2.5 million managed honeybee colonies in the U.S., down from 5 million in 1940 (Johnson, 2010). The practice of transporting beehives, called migratory beekeeping, to large monocrop farms have become an industry standard to counter-balance the decline of pollinators with the increase of crop production. Beekeepers who engage in this practice truck thousands of beehives across North America throughout the growing season to boost pollination rates on monocrop farms, which are too large to be fully pollinated by wild pollinators. While this practice counterbalances the increase in agricultural output and the decrease in domesticated honeybees, it has been shown to contribute to malnourishment, expose hives to pesticides, and cause increased infestation rates of viruses and parasites (Tarpy et al., 2013). Over the course of the season of a migratory beekeeper, beehives face a multiplicity of strains that accumulate and contribute to Colony Collapse Disorder (CCD) (Evans et al., 2009).

2.2 Relevant aspects of honeybee biology to in-hive sensor systems

Opening a beehive to collect data aggravates the colony, disrupts honeybee behavior, and limits the scope of what can be researched. As a result, little quantitative data exists on internal dynamics, and observations from hive inspections are not often integrated into honeybee research (Mezquida and Martínez, 2009). The difficulties of beehive data collection result in methodological shortcomings and sampling biases. For instance, hives are sampled only after an incident is reported in Colony Collapse Disorder (CDD) research on beehives (Kaplan, 2008). Having data on the hives

before an incident is instrumental in identifying the causal factors of CCD, in addition to improving management practices and reducing labor.

Beehive sensor systems are advantageous given honeybee hive biology, beehive seasonal growth patterns, and apicultural management practices. Honeybee colonies have been described as super organisms, which means individual honeybees in a hive act similarly to cells in multicellular organisms (Seeley, 1989; Moritz and Southwick, 2012). As such, honeybees thermoregulate and produce relative humidity changes inside the hive (Human et al., 2006). Honey production also creates humidity. Honey, which has a 20% water content, originates from nectar, which has 80% water content. Worker bees expel the water from the nectar by fanning their wings. Also, worker bees maintain the brood nest¹ at higher temperatures for optimal egg-laying conditions (Southwick, 1992). Additionally, temperature and humidity data could be useful for winter hive monitoring to detect when a hive dies. Such climatic traits of beehives lend themselves to a hive monitoring system for detecting egg laying conditions and over-wintering survival.

Honeybees produce sound waves by vibrating their wing muscles to communicate. The fundamental frequency of honeybees ranges between 100 - 200 Hz. When the hive is disturbed, worker bees emit sounds above their fundamental frequency to 200 - 250 Hz, in a behavior called worker piping (Seeley and Tautz, 2001; Lindauer, 1971). Swarming, the process of forming a new colony by the presence of a second queen cell, is initiated by auditory signals between the old queen and the newborn virgin queen.

¹ The brood nest is the area where eggs are laid inside the hive.

The existing queen emits a 1000-1500 Hz “tooting signal” that propagates through the comb to the unhatched virgin queen, who if old enough to respond, emits a 2000 - 2500 Hz “quacking response” (Hrncir et al., 2005; Gould et al., 1988). If swarming occurs, the older queen leaves the colony with 60 % of the bees and forms a new hive elsewhere. The swarming period is crucial to apiary management because it presents an opportunity to split the queens and make two hives, and if swarming occurs the resulting hive’s population is reduced to 40 %, which limits honey production. The auditory traits of beehives lend themselves to an auditory hive monitoring systems for detecting when the colony is aggravated or for detecting swarming.

Hive weight relates to seasonal population growth and honey weight. In 1997, McLennen derived a non-linear equation that explains 97 % of total variance relating hive weight to honey and hive population:

$$H = -1416.0 + 0.7604C - 57.142D + 0.487D^2 + 0.00142CD \quad (2.1)$$

where H represents honey in grams (g), C represents colony weight (g), D represents days from beginning for the first nectar flow (McLellan, 1977). In addition, this study found total hive weight highly correlated to the weight of honey ($r = 0.95$, $p < 0.001$). Hive weight informs beekeepers when to extract honey, and relative health levels by comparing hive weights within an apiary. Additionally, hive weight could be used in winter to monitor the rate of food stores consumption.

2.3 Beehive sensor systems

Using sensors to take continuous measurements inside the beehive has been a interest of apiculturalists for over 100 years (Gates, 1914). In the intervening

time, technological advancements have made it possible to design beehive sensors systems that monitor various in-hive characteristics continuously, unobtrusively, and semi-autonomously. Sensors developed in academia focus on enabling new methodologies and data creation, whereas industry developed hive sensors detect less characteristics but are modular and compact. A review of the ways in which in hive variables, including temperature, humidity, weight, gases, vibrations, have been continuously monitored by academics and industry is presented here.

2.3.1 Academic

Academic in-hive sensor research is rooted in using continuous in-hive measurement systems to gain insight into hive dynamics. This emerging field is called “Precision Apiculture” and is a subsection of precision agriculture ([Zacepins and Karasha, 2013](#)). Most of the field’s research has been exploratory and been concerned with interpreting longitudinal data of physical characteristics measured in various positions inside the hive.

Climatic conditions: temperature, humidity, and gas composition

Honeybees thermo-regulate the hive, so temperature can be related to in-hive activity and colony health. With temperature sensors installed above the upper hive box, a year-long monitoring experiment revealed brood rearing periods relate to temperature patterns ([Stalidzans and Berzonis, 2013](#)). This relationship can be used to help synchronize apiary management with hive development cycles. One

study determined that screened bottom boards² do not change honeybees ability to thermoregulate the hive (Sánchez et al., 2015).

However, in-hive temperature decreases away from the broodnest, and the location of the brood nest shifts as the queen lays eggs throughout the frames (Visscher, 1986). In a study comparing temperature sensors in various in-hive positions, temperature at the top of the honey super were found to be significantly different from the temperature in the brood nest (Meikle et al., 2015). Both positions revealed circadian sinusoidal temperature patterns, but the brood nest temperature ranged 5 °C while honey super temperature ranged over 15 °C.

By embedding sensors into wax frames in the honey super, Human et al., determined that honeybees regulate humidity at a level above external conditions (2006). In addition to detecting water vapor, CO_2 and O_2 sensors have been placed in hives, but never integrated into a continuous sensor platform. One explanation for this is the difficulty of positioning gas sensors that will not be affected by honeybees' tendency to cover foreign objects with propolis (Meikle and Holst, 2015).

Hive weight

Hive weight monitoring has the advantage that it can be detected from outside the hive. Unlike the climatic gradient inside a hive, weight is a single value. Scales

² Screen bottom boards are a variation of conventional wooden bottom boards, which is the base of the hive that rests on the ground. Screened bottom boards have a layer of wire mesh above a wooden base that prevents varroa mites that fall off bees from re-entering the hive. Screened bottom boards allow more air to enter the hive.

connected to data loggers detected a swarming event via a sharp decline in weight, a colony death via a flatline , and foraging via daily midday drops in weight (Meikle et al., 2008). In 2015, the same research group found that weight data positively correlates to population increases during nectar flows (Meikle et al., 2015). Despite the insights gained from weight, there have not been many recent attempts to integrate this parameter into research experiments.

2.3.2 Commercial beehive sensors

Apiara³ was founded in 2012, (Apiara Hive Technologies Inc., Portland, USA) and created a low-cost, battery powered, web-connected, hive scale by placing a load cell underneath the hive. The web application graphs hive weight and also temperature and rainfall taken from the closest weather station. Under a shared license, Apiara and the beekeeper have joint ownership of the data.

Arnia (Arnia Limited, United Kingdom) developed a 15 mm sensor platform that runs on 4 AAA batteries. It fits through the hive entrance and detects temperature, humidity, brood temperature, and acoustics. An additional computer, situated within 30 meters receives the hive data and broadcasts it the 2G network. Arnia remains proprietary control over the data, but shares it with the beekeeper via a web application.

³ Mention of a trade name, proprietary product, or company name is for presentation clarity and does not imply endorsement by the authors, or McGill University, nor does it imply exclusion of other products that may also be suitable.

Solutionbee (Solutionbee LLC, Raleigh, USA) manufactures scales that sit below a beehive. Like Arnia, it is battery powered and requires an on-site computer that sends data via Wi-Fi or 3G to web and mobile applications.

2.4 Review of Effects of Radio Frequency Electromagnetic Radiation on Animals

The effects of radio frequency (3 kHz - 300 GHz) electromagnetic radiation (RF-EMR) on animals are unclear and potentially harmful to honeybees. Yet, electromagnetic radiation is emitted wherever humans use electronic devices and are used in commercially available beehive sensor systems. Also, wireless sensor networks used in precision agriculture are another source of concern for pollinator health. Cucurachi et al. reviewed 113 studies on the ecological effects of RF-EMR, and reported 50 % of animal and 75 % of plant research found effects at high and low dosages without a discernible relationship between effect and RF-EMR dosage (Cucurachi et al., 2013). Of the 25 studies on the effects of RF-EMR on insects, 22 found a significant effect on insect fertility, behavior, and development (Cucurachi et al., 2013).

One of the principal concerns of the effects of RF-EMR on animals is tissue heating due to the specific absorption rate of the propagated wave. For instance, Bernardi et al. found that 50 minutes of cell phone exposure increased ear temperature by 0.22 °C to 0.43 °C in humans (Bernardi et al., 2003). Also tissue heating is the most widely accepted mechanism of microwave radiation with biological systems and RF-EMR exposure is employed in pest management for stored grain pests (Nelson, 1996).

Magnetoreception is the main mechanism of honeybee navigation and orientation. Magnetite crystals located in honeybee abdomens, Fe_3O_4 , are embedded in tissue with unpaired electrons and a light sensitive photoreceptor (Hsu et al., 2007). This process might be sensitive to RF-EMR exposure at certain levels. In migratory robins, which possess identical magnetite-based navigation systems to honeybees, RF-EMR affect navigation (Engels et al., 2014). In honeybees, pulsed RF-EMR of 0.9 GHz (GSM) induced worker piping (Favre, 2011). In 1981, the effects of 2.45 GHz were studied by trapping, marking, and releasing thousands of bees to find the rate of return compared to non-exposed bees, and 95% of the bees that returned successfully to the apiary returned to the correct hive (Westerdahl and Gary, 1981). In a similar experiment in 2006, Kimmel et al. exposed 8 colonies to 1.9 GHz RF-EMR, trapped 15 honeybees from the exposed colonies and from 8 controls, and found a significant difference in the return rate when released 500 m away (2007). The effects of frequencies of 4G and Wi-Fi networks on honeybees are not published (Meikle and Holst, 2015; Cucurachi et al., 2013).

CHAPTER 3

Wireless sensor network development

A low-cost, embedded, wireless sensor network for monitoring beehive health in real-time was developed. The system monitors temperature, humidity, and sound using off-the-shelf components. This chapter contains the following: a technical explanation of the in-hive sensor system's software, instrumentation, and the network architecture.

3.1 Sensor hardware

RaspberryPi 2 B V1.1 (Raspberry Pi Foundation LLC, Cambridge, UK) microcomputer (Figure 3-1) was used. This 85 x 56 x 17 mm microcomputer contains a Broadcom BCM2836 ARM Cortex-A7 based quad-core processor which runs at 900 MHz and holds 1 GB of ram memory. The RaspberryPi 2 B V1.1 receives 5 V 2 Amp power through a micro USB socket, provides 4 USB ports, a microSD slot, and an ethernet port. An 8 GB micro SD card ran a free, Raspberry Pi specific Linux operating system, called Raspbian. The developed hive-sensor software was installed on the Raspbian operating systems to facilitate data acquisition,

Each in-hive sensor system used an AM2303 temperature-humidity (Adafruit, Inc., New York, USA) sensor. The AM2303 is a wired version of the DHT22 (Adafruit, Inc., New York, USA); it contains a capacitive humidity sensor and a thermistor to detect temperature. The 27 x 59 x 13.5 mm sensor comes pre-calibrated and has a 0.5 Hz sampling rate, and a ± 2 % relative humidity and ± 0.5 °C



Figure 3–1: RaspberryPi 2 model B V1.1.



(a) AM2303 top view (b) AM2303 bottom view

Figure 3–2: Top and bottom views of the wired AM2303 Sensor.

temperature accuracy (Figure 3–2). To power and enable data acquisition from this sensor, the red, yellow, and black wires connected to the GPIO 3-5V pin, data input pin, and ground pin, respectively.

A PLMC15 Omnidirectional Microphone (Pyle Audio Inc., New York, USA) connected to the Pi via a USB sound card adapter. This microphone had a 3.5mv/Pa sensitivity rating and a frequency response range of 100 - 15000 Hz (Figure 3–3). The microphone was connected to the Raspberry Pi via a USB sound card adapter. These off the shelf components served as the basis for an inexpensive hive monitoring platform. Components were chosen for size, durability, and cost.



Figure 3–3: PLMC15 omnidirectional microphone.

Table 3–1: Cost (CAD) of the hive sensor hardware components.

Component	Cost
AM3203 Temperature and Humidity Sensor:	\$15.00
PLMC15 Omnidirectional Microphone:	\$13.00
USB Sound Card Adapter:	\$5.99
USB AC Adapter:	\$12.99
USB to MicroUSB cable:	\$6.99
Raspberry Pi 2 B V1.1:	\$39.95
SanDisk 8GB SD Card:	\$8.23
Total Cost:	\$102.15

3.2 Data acquisition and processing

Two pieces of original software developed in the Python language (Python Software Foundation, Delaware, USA) facilitated data acquisition and data storage. One software, named “hive node”, controlled the in-hive data acquisition, processing, and communicated with another computer that ran the second piece of software, named “hive aggregator”, which received and stored hive data in a database.

The “hive node” software was programmed to run continuously whenever the RaspberryPi turned on. It instructed the AM3203 to take a temperature measurement and a relative humidity measurement, which required no calibration or signal processing.

Several signal processing steps were required to compute the dominant sound amplitude and frequency. First, the “hive node” software instructed the microphone to take 5 s recordings. To convert the analog signal into a digital signal, the array of the recorded audio waveform was sampled at evenly-spaced time-points, and each sample was represented as a precise number:

$$Audio = \frac{M_r}{M_c} * M_{rs} \quad (3.1)$$

where M_r was the defined microphone rate, of 16000 kHz, M_c was microphone chunk, which was 2048 bytes per sample, and M_{rs} and the length of recording, which was 5 s. The output array was then converted into a 16 bit numeric string (See Appendix B, line 155).

The format of the audio data could then undergo signal processing. To calculate the amplitude in decibels, a Fourier Transform was used, which converted a sequence of signals from an original time domain, into a frequency domain ([Weisstein, 2015](#)):

$$X_k = \sum_{n=0}^{N-1} X_n \cdot (\cos(-2\pi k \frac{n}{N}) + i \sin(-2\pi k \frac{n}{N})), k \in z \quad (3.2)$$

where N is the number of time samples, n is the current sample, x_n is the value of the signal at time n , k the is the current frequency, and X_k are the frequency components of the signal (See Appendix B, line 177).

Lowpass and highpass filters were then applied to the output array to filter any noise outside of the known frequency range of honeybees:

$$H_p < P < L_p \quad (3.3)$$

where H_p was a high-pass filter of 110 Hz and L_p was a low-pass filter of 880 Hz, which encompassed the audible range of honeybee sounds (Tsujiuchi et al., 2007). From this array of frequencies, the dominant frequency was obtained by taking the median frequency of the array of frequencies. Equations 3.3 correspond with the “hive node” software listed in Appendix B, on line 173.

To compute the decibels, or the amplitude of the waveform, the base ten log of the Fast Fourier Transform output was multiplied by twenty, and the mean of the resulting amplitudes was taken as the average amplitude over the 5 s recording (Appendix B, lines 179-180).

$$dB = 20 \log_{10}(X_k) \quad (3.4)$$

3.3 Data Storage

Once the audio has been processed and the temperature and humidity data logged, this information is stored in a javascript object notation (JSON) string format. Here, the “hive node” program reaches the end of the loop and sends the sensor data to a remote computer that stores the hive data in a database. Over the IEEE 802.11n ad-hoc protocol, the remote computer that ran the “hive aggregator” program and hive sensors interface via ZeroMQ (iMatix Corporation, Brussels, Belgium) middleware software. The ZeroMQ is an open source asynchronous middleware software used in distributed and concurrent applications. This framework

enables up to 253 in-hive sensors running “hive node” to communicate independently with a central computer that ran the “hive aggregator” software.

The computer that ran “hive aggregator” software stored all incoming hive data in MongoDB (MongoDB Inc. New York City, USA) databases. MongoDB is an opensource non-relational database; instead of using the structured query language (SQL), the entries are stored in dynamic “schemas” that can change structure to incorporate new data types. The “NoSQL” database was chosen for its flexibility to accept new data types, for future sensor integration, with little user adjustment required.

3.3.1 Data retrieval

A new database was created by the “hive aggregator” program for each day of the experiment. Limiting the size of each database protects against data loss from exceeding the database capacity (Line 128 of Appendix C). To access the data stored in JSON format into a comma-separated-value (.csv) file, a MongoDB command-line tool was used (Listing 8.2 of Appendix A). Within each daily database, each hive sensor was stored separately and converted into a .csv individually (See appendix A for sample data in .json and .csv).

3.4 Wireless sensor network

Using the IEEE 802.11 protocol and a DIR-651A1 D-Link router (D-Link Inc., Taipei, Taiwan), a local area network (LAN) was implemented (Cali et al., 2000). This enabled wireless communication between the hive sensors and the computer that ran the hive aggregator software. Wireless sensor networks are increasing in popularity in precision agriculture and apicultural fields for their capabilities to

remotely collect data on animal tracking, pest control, environmental variables, and monitor farm assets (Anisi et al., 2015). In addition to improving management, wireless sensor networks could be used in researching regional effects on beehives through creating beehive sensor networks across numerous apiaries (Meikle and Holst, 2015). The potential applications of wireless technologies in precision apiculture could benefit managers, and several of the commercially available hive sensing products use wireless communication.

3.4.1 Data visualization and web application

A web application was developed that visualized the hive-sensor data in real time. The web application was developed using the D3 and jQuery/AJAX JavaScript libraries (Oracle, California, USA). Functionalities were built in to the web application that allowed the user to select which sensor parameters to plot for user-specified lengths of time (Figure 3–4). It also allowed the user to download hive sensor data from the website. The computer that ran the hive-aggregator software hosted the web application.

The web application was accessible by users within the physical broadcasting range of the router. This is because the wireless sensor network was created only as a proof of concept; the router was not connected to the internet via an internet service provider. The wireless components of the sensor network were temporarily removed and a wired version of the network using ethernet cables was installed to research the effects on 2.4 GHz RF-EMR.

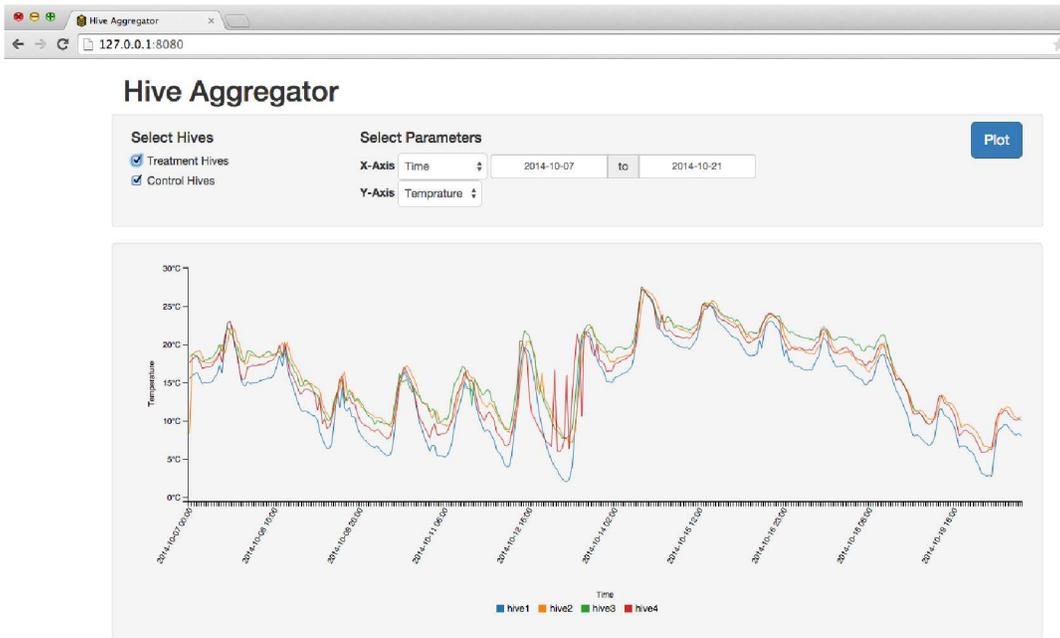


Figure 3–4: A web application displays hive sensor data.

CHAPTER 4

A study on the effects of Wi-Fi communication on measured sensor parameters inside beehive colonies

In 2015, the hive sensor network was used to research the effects of RF-EMR on the measured beehive parameters. Two groups of three Langstroth beehives were monitored: a treatment group subjected to 2.4 GHz from a Wi-Fi router, and a control group located outside of the Wi-Fi range (Figure 4-1). Wi-Fi was turned on and off every 72 hours between September 5th and October 8th. Hives consisted of 1 deep super, and 2 medium supers (Figure 4-2). The experimentation took place at the Macdonald Campus of McGill University, Ste. Anne-de-Bellevue, Québec, Canada. The hives were bred at the McGill Apicultural Association and possess hybridized Italian and Russian genetics.

4.1 Experimental design

4.1.1 Sensor network configuration

Electricity was supplied to the hive sensors via extension cables running from the barn and an empty beehive box housed the power supplies for the hive sensors (Figure 4-1). Ethernet cables connected each in-hive sensor to a D-Link (D-Link Inc., Taipei, Taiwan) TL-SG1008D 8 port Ethernet VPN router located in an empty beehive box between the treatment colonies. From here, an ethernet cable sent the sensor data through a DIR-651A1 D-Link router to the a D525 Intel Atom (Intel Inc, Santa Clara, California, USA) acting as a server located in the apiary barn (Figure

4-1). For this experiment, the network used Ethernet cables instead of wireless communication via Wi-Fi. This hardwired system eliminated additional sources of RF-EMR.

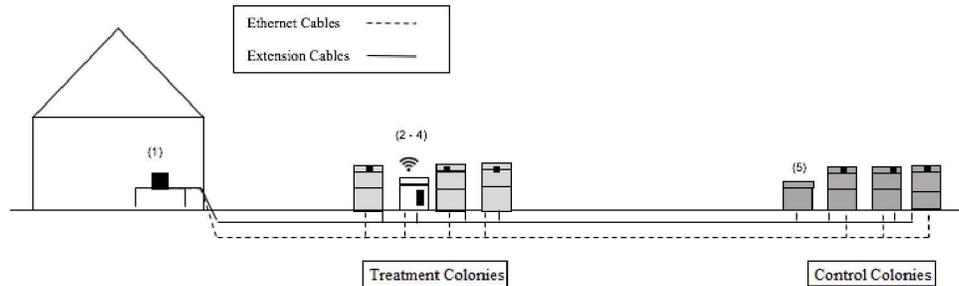


Figure 4-1: A schematic diagram of the experimental design at Macdonald Campus, McGill University, Ste. Anne-de-Bellevue, Quebec. (1) Server, (2-4) Router, computer, and treatment group sensor power supplies, (5) control group sensor power supplies.

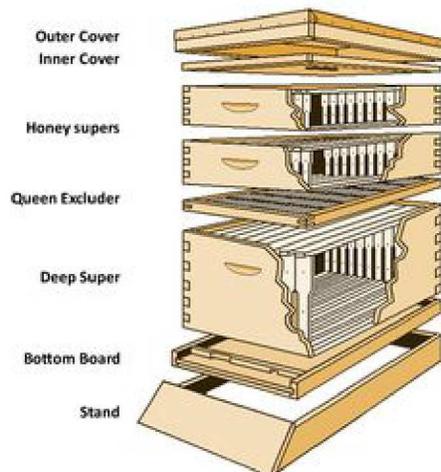


Figure 4-2: A Langstroth hive diagram (Blackiston, 2012).

4.1.2 Sensor installation

The hive node hardware was installed in plastic containers fastened to the inside of the hive roofs to prevent moisture damage. The microphone and temperature

and humidity sensor were located outside the container facing into the hive (Figure 4-3). Instead of an inner cover (Figure 4-2) A propolis trap was placed between the roof and the top super that separated sensors from the colony, prevented propolis formation on sensor components, and prevented burr comb formation inside the hive roof. Queen excluders, used to prevent egg laying in upper supers to maximize honey production, were not used during the experiment because honey production was not a focus (Figure 4-2).

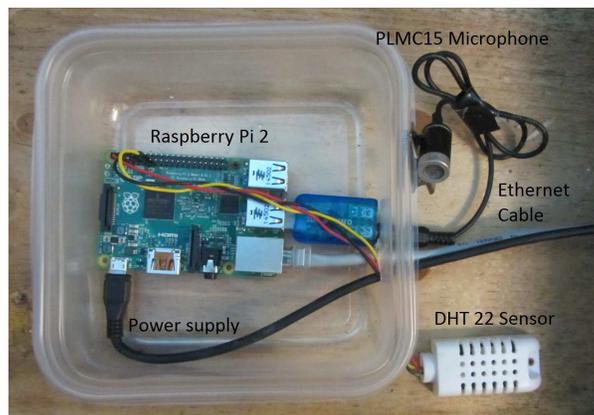


Figure 4-3: Hive sensor installed in the inside of beehive roofs.

4.2 Wi-Fi electromagnetic radiation

To create a reliable and strong Wi-Fi signal, i.e. the RF-EMR treatment, a second D525 Intel Atom (Intel Inc., Santa Clara) sent continuous ping flood requests to a C2 AC750 TP-Link router (TP-Link, Shenzhen, China), both installed in the empty beehive box between the treatment hives (Figure 4-1). This router was configured to broadcast at a maximum power of 0.001 mW, or -30 dBm, and was the only source of relatively strong RF-EMR in proximity to the beehives and only source

of Wi-Fi (Figure 4-4). Wi-Fi was not used to transmit data from the treatment colonies during Wi-Fi exposure in order to standardize the sensor configuration across treatment and control groups, and not introduce other sources of RF-EMR into the experiment.



Figure 4-4: An interpolated map of 2.4 GHz signal strength when Wi-Fi was off (left) and on (right) at the Macdonald Campus Apiary, Ste. Anne-de-Bellevue, Québec.

To determine where to place the control colonies safely outside of the Wi-Fi RF-EMR range, the distribution of signal strength from the router was mapped using a RF Explorer (Nuts About Nets, Redmen, WA, USA). Control hives were placed where the signal strength before and during Wi-Fi exposure were equivalent (Figure 4-4).

4.3 Data analysis

Sensor data were downloaded from the database and hourly averages were taken. Since every hive presents a unique microenvironment, the comparison should reveal the differences between treatment and control hives during Wi-Fi exposure, while

accounting for the differences wherein the router was turned off. Therefore, to analyze the effect of RF-EMR, the following equation was used:

$$D_{(i,j)n} = (X_{(i)n}^T - \overline{X_{(off)}^T}) - (X_{(j)n}^C - \overline{X_{(off)}^C}) \quad (4.1)$$

where $D_{(i,j)n}$ was the difference between i^{th} treatment and j^{th} control for the n^{th} record; $X_{(i)n}^T$ was the measured parameter for the i^{th} treatment and n^{th} record; $\overline{X_{(off)}^T}$ was the average measured parameter for the n^{th} treatment when Wi-Fi was off. $X_{(j)n}^C$ was the measured parameter for the j^{th} control and n^{th} record; $\overline{X_{(off)}^C}$ was the average measured parameter for the n^{th} control when Wi-Fi was off.

It was assumed that there were not RF-EMR induced differences between treatment and control hives when the Wi-Fi was off (average $D_{(i,j)n}$, or $\overline{D_{(i,j)off}}$ for every treatment / control combination equaled zero). This resulted that the $D_{(i,j)n}$, or $\overline{D_{(i,j)on}}$, during Wi-Fi exposure revealed the effect of the treatment. As nine treatment control combinations (m=9) existed in this study, a t-test was used to evaluate the significance between treatment effect values ($\overline{D_{(m)on}}$) when Wi-Fi was on compared to when Wi-Fi was off where $\overline{D_{(i,j)on}}$ equalled 0.

$$t = \frac{\overline{D_{(m)on}} - 0}{\frac{\sigma}{\sqrt{9}}} \quad (4.2)$$

For the duration of the study, climate data (temperature and humidity), from the Montreal Island Weather Station, Dorval, Quebec, was monitored. These records were compared using the coefficient of determination to the recorded measurement to define the influence of environmental parameters on the status of beehives.

CHAPTER 5

Results and discussion

5.1 Results

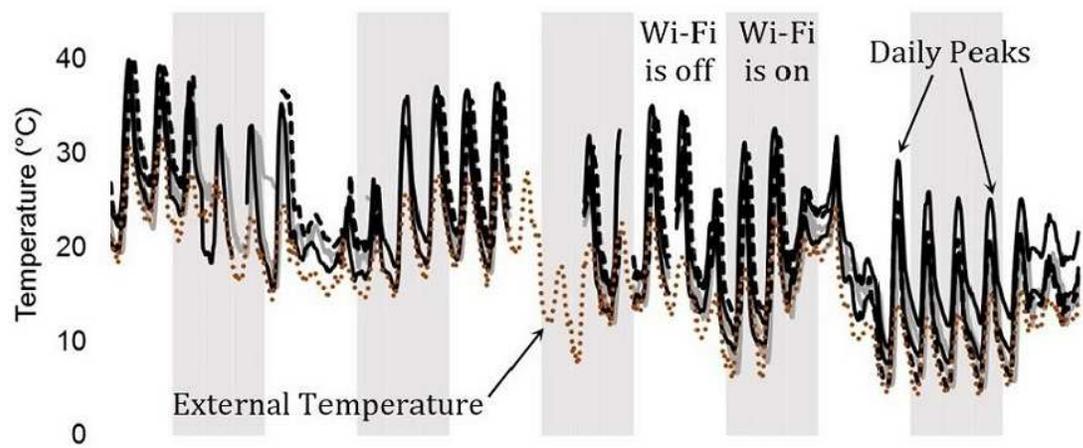
The results of the sampling periods are shown in Figure 5–1 for in-hive climatic conditions and in Figure 5–2 for in-hive acoustic conditions. In-hive climatic conditions and sound amplitude exhibit circadian patterns throughout the experiment, and in-hive temperature correlated with external temperature ($r^2 = 0.87$), unlike in-hive relative humidity, which did not follow the external relative humidity trend ($r^2 = 0.10$). Tables 5–1 and 5–2 and summarize all sensor measurements obtained for each group of hives with and without the Wi-Fi exposure.

The results of the effect of RF-EMR according to Equation 4.1 are shown in Figures 5–3 and 5–4. Table 5–6 presents the grand mean, $\overline{\overline{D_{(m)on}}}$, of all hive pairs from the results from Equation 4.1 during RF-EMR exposure and the significance of the effect of RF-EMR. Over the 5 Wi-Fi exposure periods, the treatment group in-hive temperature increased by 0.09 C, humidity increased by 1.53 %, and amplitude increased by 0.03 dB, and frequency decreased by 2.57 Hz (Table 5–6). However, these changes were much lower than standard deviation for $D_{(i,j)n}$ during any of the RF-EMR exposure periods. Thus, the treatment effect was found insignificant at the $p < 0.05$ level (Table 5–6).

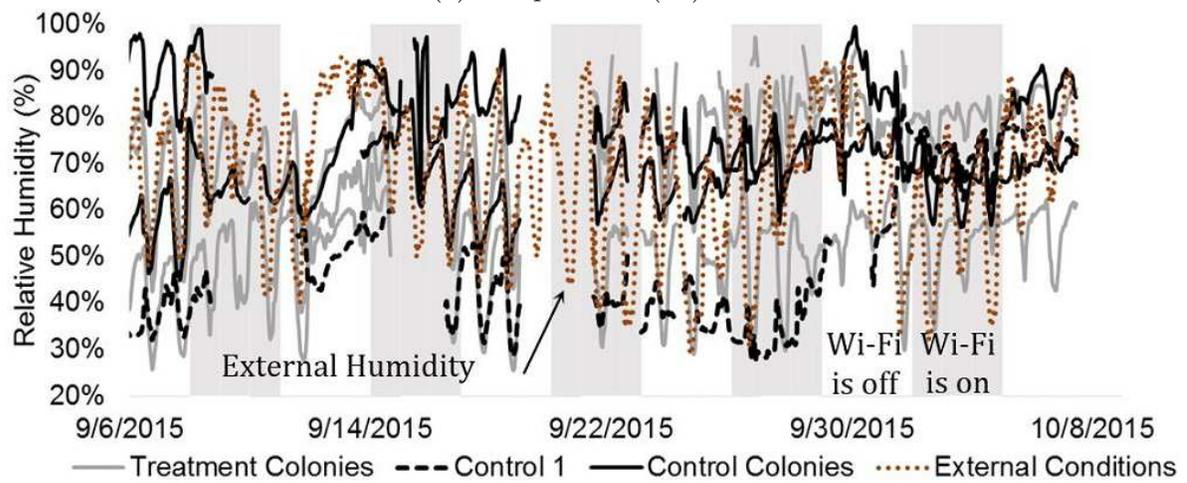
The in-hive sensors captured several inter-colony discrepancies. As it can be seen from Figure 5–2, the control colonies experienced several spikes in frequency and

amplitude at night. Hypothetically, this could be due to wild animal disturbances, which might be an excellent tool to inform the apiary manager about a potentially hazardous situation. However, in our case no surveillance was done to support or deny this hypothesis in terms of the source of beehive disturbance.

In addition, Control Hive 1 was on average -14.37 % humid and -1.57 dB than the other 5 monitored colonies. After an in-depth investigation, it was found that Varroa mites were visually noticeable in several hives. On October 1, the infestation rates were estimated by taking approximately 300 sample bees from the brood chamber in the 5th frame from the left in the bottom super. The Varroa mites were dislodged and separated from the sampled honeybees using the sugar roll method (Dietemann et al., 2013). Powdered sugar was added to the sample, rolled for 30 s, and shaken for 1 min through a wire mesh screen. According to Figure 5-5, Varroa infestation rates varied between 5.90 % in Treatment Hive 2 and 41.52 % in Control Hive 1. This Control Hive 1 anomaly is highlighted in Figures 5-1 and 5-2 and in Tables 5-1 - 5-4. The statistical analysis was redone without Control Hive 1, and all measured parameters were not significant at the $p < 0.2$ level (Table 5-7). This supports that the “dry” and “quiet” environment in Control Hive 1 could be attributed to the Varroa infestation in this colony. No other anomalies could be found through weekly visual observations of colonies in each of the hives.

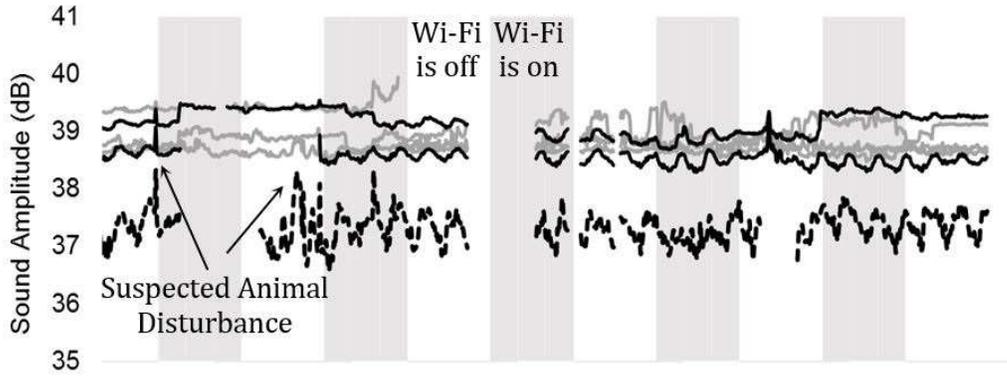


(a) Temperature (°C)

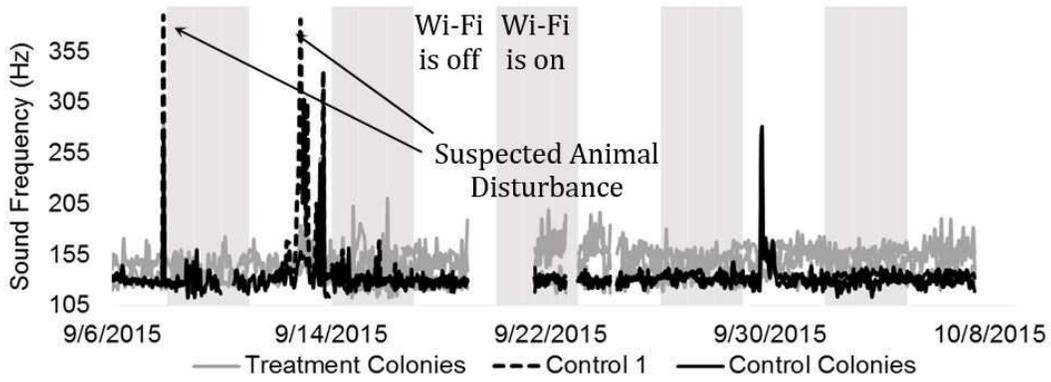


(b) Humidity (%)

Figure 5–1: Per hour averaged in-hive climatic data, gray vertical bars represent periods of Wi-Fi exposure.



(a) Amplitude (dB)

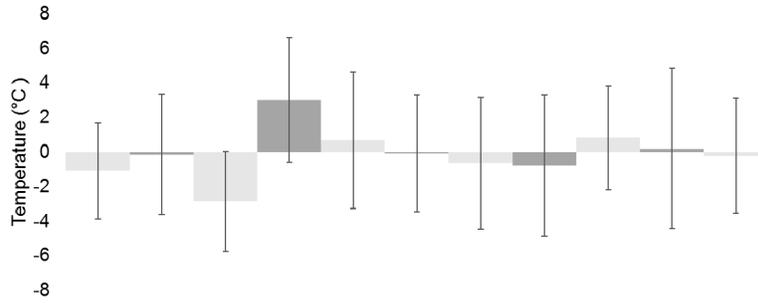


(b) Frequency (Hz)

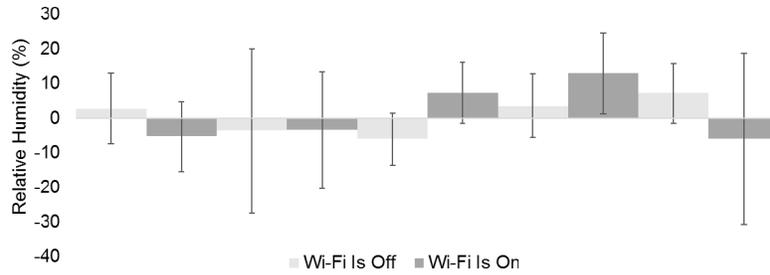
Figure 5–2: Per hour averaged in-hive acoustic data, gray vertical bars represent periods of Wi-Fi exposure.

Table 5–1: Summarized climatic data presented as mean \pm SD.

	Temperature ($^{\circ}$ C)		Humidity (%)	
	Wi-Fi Off	Wi-Fi On	Wi-Fi Off	Wi-Fi On
Treatment Hives:	21.50 \pm 6.43	19.36 \pm 8.10	64.25 \pm 10.99	66.01 \pm 10.22
Control Hives:	21.46 \pm 6.70	20.94 \pm 7.89	66.13 \pm 11.83	65.07 \pm 11.89
Control Hive 1:	21.47 \pm 7.51	20.62 \pm 8.74	51.22 \pm 15.15	49.40 \pm 16.96
Control Hive 2 and 3:	21.45 \pm 6.29	21.11 \pm 7.47	73.58 \pm 10.17	72.90 \pm 9.36



(a) Adjusted Effect Temperature (°C)



(b) Adjusted Effect Humidity (%)

Figure 5–3: Results of RF-EMR analysis following Equation 4.1 for treatment group in-hive climatic conditions, presented as average value per exposure period and standard deviation.

Table 5–2: Summarized acoustic data presented as mean \pm SD.

	Amplitude (dB)		Frequency (Hz)	
	Wi-Fi Off	Wi-Fi On	Wi-Fi Off	Wi-Fi On
Treatment Hives:	38.86 \pm 0.17	38.91 \pm 0.21	145.81 \pm 12.10	143.39 \pm 16.12
Control Hives:	38.33 \pm 0.19	38.34 \pm 0.20	133.56 \pm 17.44	131.88 \pm 15.53
Control Hive 1:	37.30 \pm 0.28	37.33 \pm 0.26	135.81 \pm 27.65	132.18 \pm 20.28
Control Hive 2 and 3:	38.85 \pm 0.15	38.85 \pm 0.17	132.44 \pm 12.33	131.72 \pm 13.16

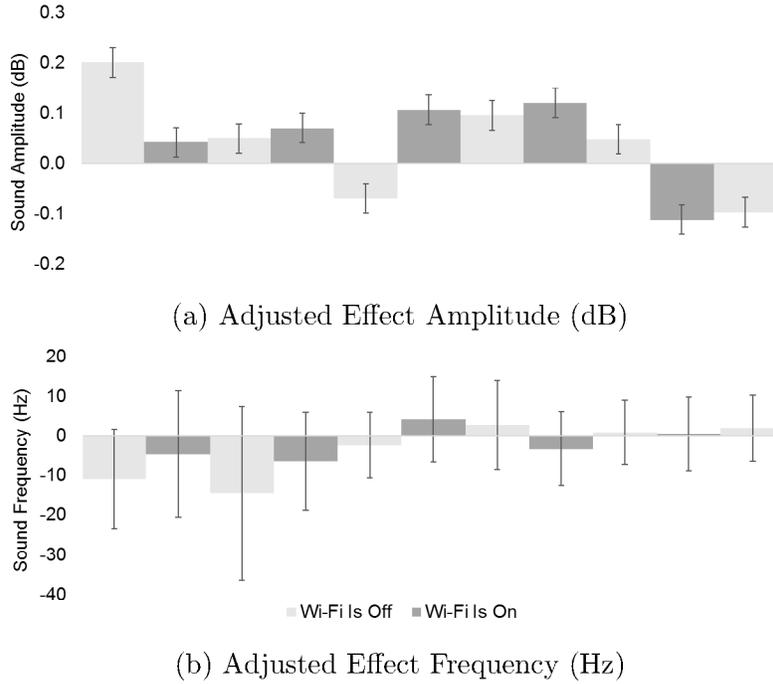


Figure 5-4: Results of RF-EMR analysis following Equation 4.1 for treatment group in-hive acoustic conditions, presented as average value per exposure period and standard deviation.

Table 5-3: Results of RF-EMR analysis following Equation 4.1 for in-hive climatic conditions. Treatment effect values are presented as mean \pm SD for total exposure and non-exposure periods pair Interactions containing Control Hive 1 are in bold.

Interaction	Temperature ($^{\circ}$ C)		Humidity (%)	
	Wi-Fi Off	Wi-Fi On	Wi-Fi Off	Wi-Fi On
Trt_1 and $Crtl_1$:	0.00 \pm 2.89	0.42 \pm 3.76	0.00 \pm 11.95	2.07 \pm 18.61
Trt_1 and $Crtl_2$:	0.00 \pm 3.76	0.56 \pm 4.71	0.00 \pm 15.89	5.10 \pm 20.84
Trt_1 and $Crtl_3$:	0.00 \pm 3.92	0.78 \pm 4.89	0.00 \pm 9.94	-3.21 \pm 16.90
Trt_2 and $Crtl_1$:	0.00 \pm 2.89	-0.52 \pm 3.30	0.00 \pm 7.77	0.07 \pm 11.43
Trt_2 and $Crtl_2$:	0.00 \pm 4.12	-0.39 \pm 4.59	0.00 \pm 10.69	3.10 \pm 14.29
Trt_2 and $Crtl_3$:	0.00 \pm 3.86	0.30 \pm 4.17	0.00 \pm 7.63	-5.38 \pm 8.25
Trt_3 and $Crtl_1$:	0.00 \pm 2.89	-0.22 \pm 3.26	0.00 \pm 8.12	3.54 \pm 11.42
Trt_3 and $Crtl_2$:	0.00 \pm 3.74	-0.08 \pm 4.01	0.00 \pm 12.80	6.57 \pm 13.58
Trt_3 and $Crtl_3$:	0.00 \pm 3.85	0.14 \pm 4.18	0.00 \pm 6.79	-1.74 \pm 8.43

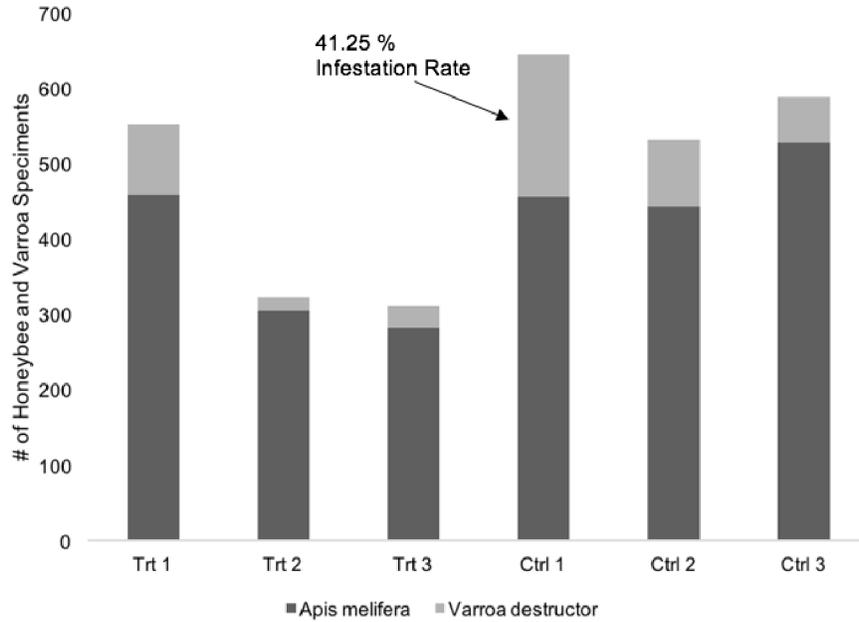


Figure 5-5: Varroa infestation rate.

Table 5-4: Results of RF-EMR analysis following Equation 4.1 for in-hive acoustic conditions. Treatment effect values are presented as mean \pm SD for total exposure and non-exposure periods pair Interactions containing Control Hive 1 are in bold.

Interaction	Amplitude (dB)		Frequency (Hz)	
	Wi-Fi Off	Wi-Fi On	Wi-Fi Off	Wi-Fi On
Trt_1 and $Ctrl_1$:	0.00 \pm 0.33	0.11 \pm 0.37	0.00 \pm 9.63	-8.71 \pm 19.50
Trt_1 and $Ctrl_2$:	0.00 \pm 0.18	0.13 \pm 0.26	0.00 \pm 9.98	-8.70 \pm 13.98
Trt_1 and $Ctrl_3$:	0.00 \pm 0.26	0.09 \pm 0.33	0.00 \pm 11.00	-6.95 \pm 12.80
Trt_2 and $Ctrl_1$:	0.00 \pm 0.25	0.03 \pm 0.34	0.00 \pm 14.49	-5.84 \pm 25.18
Trt_2 and $Ctrl_2$:	0.00 \pm 0.19	0.06 \pm 0.29	0.00 \pm 14.55	-5.83 \pm 19.98
Trt_2 and $Ctrl_3$:	0.00 \pm 0.22	0.01 \pm 0.32	0.00 \pm 15.11	-3.23 \pm 14.04
Trt_3 and $Ctrl_1$:	0.00 \pm 0.22	-0.03 \pm 0.24	0.00 \pm 6.34	2.35 \pm 10.90
Trt_3 and $Ctrl_2$:	0.00 \pm 0.07	-0.01 \pm 0.09	0.00 \pm 6.71	2.37 \pm 8.56
Trt_3 and $Ctrl_3$:	0.00 \pm 0.16	-0.05 \pm 0.25	0.00 \pm 7.30	4.11 \pm 11.28

Table 5–5: Correlation (r^2) between in-hive temperature and relative humidity with external conditions.

	Temperature	Humidity
Control Hives:	0.86	0.09
Treatment Colonies:	0.87	0.11

Table 5–6: Grand mean and SD of the nine hive-pair interactions, $\overline{D_{(m)on}}$, during RF-EMR exposure and results of the t-test.

	Grand Mean	SD	p - Value
Temperature ($^{\circ}$ C):	0.11	0.44	0.48
Humidity (%):	1.12	3.97	0.42
Sound Amplitude (dB):	0.03	0.07	0.14
Sound Frequency (Hz):	-3.38	5.04	0.08

Table 5–7: Grand mean and SD of the six hive-pair interactions that do not contain Control Hive 1, $\overline{D_{(m-c1)on}}$, during RF-EMR exposure and results of the t-test.

	Grand Mean	SD	p - Value
Temperature ($^{\circ}$ C):	0.22	0.43	0.26
Humidity (%):	0.74	4.85	0.72
Sound Amplitude (dB):	0.03	0.07	0.27
Sound Frequency (Hz):	-3.03	5.21	0.21

5.2 Discussion

5.2.1 Research experiment

The developed sensor network provided a rich dataset applicable for academic and apiary management applications. The results of this study indicate 72 hour periods of exposure to high levels of 2.4 GHz RF-EMR do not affect beehives according to the measured climatic and acoustic conditions. There is not sufficient evidence to suggest that the means were different thus the null hypothesis that 2.4 GHz RF-EMR has no effect on in-hive dynamics was not rejected. However this does not confirm the safety of using 2.4 GHz wireless communications in beehive sensors.

There are similarities in the literature with the presented in-hive climatic conditions. The low correlation between external and internal relative humidity reflects the beehive's ability to regulate internal humidity (Human et al., 2006). Meikle et al. found .903 r^2 correlation between external and internal temperatures detected from the inside the top of the beehive box (Meikle et al., 2015).

Effects of electromagnetic radiation

These observations do not indicate honeybee colonies are sensitive to high-power 2.4 GHz RF-EMR emissions from a Wi-Fi router. The effect of the RF-EMR is insignificant at the $p < 0.05$ level (Table 5-6). In general the results of this study indicate high levels of 2.4 GHz RF-EMR during 72 hour periods did not affect temperature, humidity or acoustic measurements as compared to the hives that did not receive such an exposure. Furthermore, when not considering the varroa infested Control Hive 1, the results were insignificant at the $p < 0.2$ level (Table 5-7). Indirectly, this means that there was no change in the beehive environment that could

be directly linked to the RF-EMR exposure. On the other hand, this does not mean that long-term RF-EMR exposure would not affect honeybees at the physiological level and the question on safe use of wireless data communication through apiaries remains open. Despite the results of this study indicate that RF-EMR has no effect on in-hive temperature, relative humidity, and sound as detected from inside the hive-roof, other dynamics have been affected by RF-EMR , such as initiating worker piping (.9 GHz) and navigation outside of the hive (1.9 GHz) ([Kimmel et al., 2007](#); [Favre, 2011](#)).

Parasite infestation

The Varroa mite is a severe pest to honeybees worldwide and crucial to hive management. Interestingly, Control Hive 1 had a 41.52 % infestation rate and was the least humid and the quietest, on average -14.37 % humid and -1.57 dB than average. The sensor data suggests that high Varroa infestation is correlated a hampered ability to produce humidity and sound amplitude.

It is well documented that the Varroa impacts the honeybee at the individual level. Adult Varroa live on the back of honeybees and feed on their hemolymph ([Oldroyd, 2007](#)). This dynamic is known to physically injure the bee physically and morphologically, but how these effects manifest changes in in-hive dynamics is not understood. [Kraus and Velthuis](#) reported that high in-hive humidity limits reproduction of Varroa and brood development cycles and internal climatic conditions are drivers of the Varroa population (1997). Varroa mites are highly effective vectors of the deformed wing virus, a virus causing crumpled wings in adults and renders emerging honeybees flightless ([Yañez et al., 2012](#)). Honeybees create sound and

produce humidity through evaporating water via wing movement (Kraus and Velthuis, 1997). The added weight of the Varroa on the honeybee in combination with crumpled wings due to the deformed wing virus transmission could reduce in-hive sound levels, and it is suspected that this was occurred in Control Hive 1.

These findings support the idea that Varroa produced detectable in-hive changes of sound and humidity levels, and that humidity and sound sensors could be used to detect Varroa infestations. Decision support systems rooted in detecting important in-hive dynamics from sensor data has been suggested as a mid-term goal for precision apiculture (Zacepins et al., 2015). Future research is needed on this relationship to develop a Varroa infestation detection system through in-hive humidity and sound sensors.

Sound

Around midnight, spikes in frequency and amplitude were observed in the control hives (Figure 5-2). We suspect this noise corresponds to a skunk that attempted to raid the hive, a common pest to beehives (O'Brien and Marsh, 1990). This could not be validated and no visual evidence was found. The control hives were placed closed to the apiary's eastern tree line, which could explain why the control hives experienced spikes and not the treatment hives (Figure 4-4). This provides preliminary evidence for the idea of a sound based detection system to protect a beehive from animal predators; this might be an excellent tool to be used to inform apiary managers about potentially hazardous situations.

5.3 Design Considerations

Designing an unobtrusive in-hive sensor system that provides useful data poses challenges both for the hive and for the apiary. Many types of sensors have been tested for beehive monitoring, but it is unclear how useful individual parameters are (Meikle et al., 2015). The relationship between the location of the sensor and the usefulness of data is also unclear. Embedding sensors in the brood nest may complicate the required manual beehive inspections (Sánchez et al., 2015). Other considerations include protecting the sensor equipment from being damaged by the honeybees, whether it be moisture or propolis formation on sensor components.

The developed sensor system allowed for easy hive inspections without having to adjust sensor components. This location also allowed easy access to the sensors, which is desirable for sensor maintenance and installation, and also allowed for the sensors to be moved to different hives without colony interference. However, sensor data varies depending on proximity to the queen and brood nest Meikle and Holst (2015). Convenience of sensor location might be a trade-off with quality of data, and between sensor location and ease of use, but further research on in-hive sensor placement will contribute to a better understanding of this dichotomy.

5.3.1 Sensor performance and future improvements

The designed sensor system functioned well, but hardware and software vulnerabilities were identified. Over the course of the experiment, the sampling rate changed and varied between hive sensors (Table 5–8). This could be due to two factors, water damage and code inefficiencies.

Table 5–8: Per minute sampling rates of sensors on one day for each week of the experiment.

Hive	Per day hive sampling rates (samples/minute)					
	Treatment 1	Treatment 2	Treatment 3	Control 1	Control 2	Control 3
September 6, 2015:	2.86	2.86	2.86	2.86	5.63	5.74
September 14, 2015:	2.40	2.41	2.43	2.41	5.05	4.75
September 21, 2015:	2.86	2.85	2.86	2.85	3.21	2.45
September 28, 2015:	2.86	2.86	2.86	2.86	3.21	2.43
October 5, 2015:	2.86	2.86	2.86	8.95*	3.14	2.44

* replaced Control 1 Raspberry Pi on October 2

Moisture damage from in-hive humidity permanently damaged several microSD cards that acted as the hive sensor operating system. During the experiment, three microSD cards had to be replaced because the RaspberryPi microcomputers did not turn on. When the RaspberryPi in control hive 1 failed to turn on it was replaced and the sampling rate increased to 8.95 samples per minute (Table 5–8). The lack of humidity protection could be fixed easily by sealing the sensors better.

The second source of delayed sampling frequency was a consequence of running the "hive-node" software continuously. The amount of memory it consumed reduced the performance of the computers. In a future edition, CRON could be employed and the hive-node could be re-programmed to run one time, i.e. take one sample and send it to the aggregator computer, then close. CRON is a time-based program scheduler built into the Linux operating system. Considering that a 1-hr moving average was used to reduce the data, CRON could be used in the hive-node program at a less frequent interval, for example every 30 minutes. This approach is used by the open-source hive sensing project HiveTool, which has a sampling rate of once every 5 minutes ([HiveTool](#)).

Twice during the experiment, the power went out in the apiary barn, resulting in data loss. Using other sources of power should be explored. Many of the commercially available hive sensors use disposable batteries and low-power wireless technologies. In 2014, a hive equipped with a version of the hive node software powered by a solar panel was tested but it was more costly, required constant sunlight (Figure 5-6).



Figure 5-6: A hive sensor powered by a solar panel, tested in 2014 at the Macdonald Campus apiary

With the solar panel, the hive node software ran once every 15 minutes, allowing for the solar panel to charge sufficiently.

The developed sensor network used the IEEE 802.11 standard, used by Wi-Fi, but several others exist. An increasingly popular standard is IEEE 802.15.4, which serves as the basis for the Zigbee protocol, used by the [Mezquida and Martínez \(2009\)](#) and [Murphy et al. \(2015\)](#) hive sensor platforms. IEEE 802.15.4 is the subject of much discussion in the low-power wireless community as a means for integrating

low-power wireless devices into the TCP/IP network stack by the implementation of a new network protocol. One such effort is known as 6LoWPAN, which would allow the IPv6 protocol to operate on IEEE 802.15.4 wireless hardware for low-power devices like sensor networks (Gazis et al., 2012).

In terms of hardware improvements, Raspberry Pi released the model Zero in February 2016, which costs \$5 (USD) and measures 65 x 30 x 5 mm, less than half the size of the Model 2 B V1.1 that was used for this research. Smaller sensor components make it easier to embed inside a hive, in addition to lowering costs, which is a barrier for beekeepers to adopt in-hive sensor systems.

5.3.2 Apiary economy and practicality

There are limitations in designing in-hive sensors and beehive sensor networks that limit the sensor systems' applicability in the industry. Commercial beekeeping for migratory pollination services manage the largest amount of hives and involves moving hundreds and thousands of hives on trucks to farms. Because the developed system relied on access to electricity and stationary beehives, beekeeping for honey production involves less movement of beehives but the scale of commercial beekeeping alone makes implementing in-hive sensors for an entire apiary financially and logistically intractable.

In two interviews with professional beekeepers, they both thought the idea of equipping every hive in a commercial operation with sensors was unrealistic and undesirable to administer and install the sensor systems (Babic, 2015; Giovenazzo, 2016). Dr. Giovenazzo preferred a hive sensor device that could be mobile and brought to each hive for quick measurements, such that one hive sensor would be

needed per apiary (Giovenazzo, 2016). While this solution would keep sensor costs low for the commercial beekeeper, this was not considered during this research.

Currently, hive sensor technology is more suitable for researchers and hobbyists, who manage a smaller number of hives in the vicinity of electricity. Researchers can apply and benefit from hive sensing technologies; those who can use in-hive sensors to gain insight on in-hive dynamics, enable new research methodologies, and develop better hive sensor designs. Wireless sensor networks, if proven not to harm all aspects of honeybee behavior, have the potential to simplify in-hive sensor deployment and management.

The majority of experiments on hive sensors are exploratory, and there are still few established or documented correlations between in-hive data with beehive phenomena. With more research on detecting important hive dynamics for beekeepers, hive sensors might become more practical for non-hobbyist beekeepers. However, the most salient aspect of hive sensors is how it improves the apicultural research.

CHAPTER 6

Conclusion

A wireless sensor network was developed for continuous beehive monitoring. The network measured in-hive temperature, relative humidity, sound amplitude, and sound frequency, and plots data in real time on a web application.

A wired version of the developed sensor network was used to determine whether or not RF-EMR from Wi-Fi affects honeybee hives through a 30-day study in 2015. Two groups of three beehives were monitored: the first group was subjected to 2.4 GHz Wi-Fi signal while the second group was located outside of radio communication range. No significant effect of short term 2.4 GHz RF-EMR exposure on the measured in-hive conditions was found. However, this study did not consider potential effects outside of the measured parameters. Given the complexity of beehive dynamics, research on other potential effects of RF-EMR is needed before adopting wireless technologies in beehive sensors.

Abnormal in-hive conditions that were observed could be linked with a stressed colony (increased Varroa destructor infestation rate), which can help apiculturists reduce production losses by rapidly reacting to the observed indicators of potential stress. The developed sensor network provided a rich dataset applicable for academic and apiary management applications.

REFERENCES

- Aizen, M.A., Harder, L.D., 2009. The global stock of domesticated honey bees is growing slower than agricultural demand for pollination. *Current biology* 19, 915–918.
- Anisi, M.H., Abdul-Salaam, G., Abdullah, A.H., 2015. A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture. *Precision Agriculture* 16, 216–238.
- Babic, B., 2015. Personal Interview. ApiGuru Inc. April 10, 2015.
- Bernardi, P., Cavagnaro, M., Pisa, S., Piuze, E., 2003. Specific absorption rate and temperature elevation in a subject exposed in the far-field of radio-frequency sources operating in the 10-900-mhz range. *Biomedical Engineering, IEEE Transactions on* 50, 295–304.
- Blackiston, H., 2012. *Building a Beehive - For Dummies*. Wiley.
- Calderone, N.W., 2012. Insect pollinated crops, insect pollinators and us agriculture: trend analysis of aggregate data for the period 1992–2009. *PloS one* 7, e37235.
- Cali, F., Conti, M., Gregori, E., 2000. Ieee 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. *Selected Areas in Communications, IEEE Journal on* 18, 1774–1786.

- Cucurachi, S., Tamis, W., Vijver, M., Peijnenburg, W., Bolte, J., de Snoo, G., 2013. A review of the ecological effects of radiofrequency electromagnetic fields (rf-emf). *Environment international* 51, 116–140.
- Dietemann, V., Nazzi, F., Martin, S.J., Anderson, D.L., Locke, B., Delaplane, K.S., Wauquiez, Q., Tannahill, C., Frey, E., Ziegelmann, B., et al., 2013. Standard methods for varroa research. *Journal of Apicultural Research* 52, 1–54.
- Engels, S., Schneider, N.L., Lefeldt, N., Hein, C.M., Zapka, M., Michalik, A., Elbers, D., Kittel, A., Hore, P., Mouritsen, H., 2014. Anthropogenic electromagnetic noise disrupts magnetic compass orientation in a migratory bird. *Nature* 509, 353–356.
- Evans, J.D., Saegerman, C., Mullin, C., Haubruge, E., Nguyen, B.K., Frazier, M., Frazier, J., Cox-Foster, D., Chen, Y., Underwood, R., et al., 2009. Colony collapse disorder: a descriptive study. *PloS one* 4, e6481.
- Favre, D., 2011. Mobile phone-induced honeybee worker piping. *Apidologie* 42, 270–279.
- Gates, B.N., 1914. The temperature of the bee colony. 96, US Dept. of Agriculture.
- Gazis, V., Sasloglou, K., Frangiadakis, N., Kikiras, P., 2012. Wireless sensor networking, automation technologies and machine to machine developments on the path to the internet of things, in: *Informatics (PCI), 2012 16th Panhellenic Conference on, IEEE*. pp. 276–282.
- Giovenazzo, P., 2016. Personal Interview, Chair of Apiology, University of Laval, Jan. 25, 2016.
- Gould, J.L., Gould, C.G., et al., 1988. The honey bee. *Scientific American Library*.
- HiveTool, . URL: <http://hivetool.org/w/index.php?title=HiveTool.org>.

- Hrnčir, M., Barth, F.G., Tautz, J., 2005. 32 vibratory and airborne-sound signals in bee communication (hymenoptera). *Insect Sounds and Communication: Physiology, Behaviour, Ecology, and Evolution* , 421.
- Hsu, C.Y., Ko, F.Y., Li, C.W., Fann, K., Lue, J.T., 2007. Magnetoreception system in honeybees (*Apis mellifera*). *PLoS One* 2, e395–e395.
- Human, H., Nicolson, S.W., Dietemann, V., 2006. Do honeybees, *Apis mellifera* *scutellata*, regulate humidity in their nest? *Naturwissenschaften* 93, 397–401.
- Johnson, R., 2010. Honey bee colony collapse disorder. Congressional Research Service Washington.
- Kaplan, K., 2008. Colony collapse disorder: a complex buzz. *Agricultural Research Magazine* URL: <http://agresearchmag.ars.usda.gov/2008/may/colony>.
- Kimmel, S., Kuhn, J., Harst, W., Stever, H., 2007. Effects of electromagnetic exposition on the behavior of the honeybee. *Apis mellifera* , 1–6.
- Kraus, B., Velthuis, H., 1997. High humidity in the honey bee (*Apis mellifera* L.) brood nest limits reproduction of the parasitic mite *Varroa jacobsoni* Oud. *Naturwissenschaften* 84, 217–218.
- Lindauer, M., 1971. Communication among social bees .
- McLellan, A., 1977. Honeybee colony weight as an index of honey production and nectar flow: a critical evaluation. *Journal of Applied Ecology* , 401–408.
- Meikle, W., Holst, N., 2015. Application of continuous monitoring of honeybee colonies. *Apidologie* 46, 10–22.

- Meikle, W., Weiss, M., Stilwell, A., 2015. Monitoring colony phenology using within-day variability in continuous weight and temperature of honey bee hives. *Apidologie* , 1–14.
- Meikle, W.G., Rector, B.G., Mercadier, G., Holst, N., 2008. Within-day variation in continuous hive weight data as a measure of honey bee colony activity. *Apidologie* 39, 694–707.
- Mezquida, D.A., Martínez, J.L., 2009. Short communication.: Platform for bee-hives monitoring based on sound analysis. a perpetual warehouse for swarm s daily activity. *Spanish journal of agricultural research* , 824–828.
- Moritz, R., Southwick, E.E., 2012. Bees as superorganisms: an evolutionary reality. Springer Science & Business Media.
- Murphy, F.E., Popovici, E., Whelan, P., Magno, M., 2015. Development of an heterogeneous wireless sensor network for instrumentation and analysis of beehives, in: *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International, IEEE*. pp. 346–351.
- Nelson, S.O., 1996. Review and assessment of radio-frequency and microwave energy for stored-grain insect control. *Transactions of the ASAE* 39, 1475–1484.
- O'Brien, J.M., Marsh, R.E., 1990. Vertebrate pests of beekeeping.
- Oldroyd, B.P., 2007. What's killing american honey bees? *PLoS biology* 5.
- Sánchez, V., Gil, S., Flores, J.M., Quiles, F.J., Ortiz, M.A., Luna, J.J., 2015. Implementation of an electronic system to monitor the thermoregulatory capacity of honeybee colonies in hives with open-screened bottom boards. *Computers and Electronics in Agriculture* 119, 209–216.

- Seeley, T.D., 1989. The honey bee colony as a superorganism. *American Scientist* 77, 546–553.
- Seeley, T.D., Tautz, J., 2001. Worker piping in honey bee swarms and its role in preparing for liftoff. *Journal of Comparative Physiology A* 187, 667–676.
- Southwick, E., 1992. Physiology and social physiology of the honey bee. Graham J. *The Hive and the Honey Bee* , 171–183.
- Stalidzans, E., Berzonis, A., 2013. Temperature changes above the upper hive body reveal the annual development periods of honey bee colonies. *Computers and electronics in agriculture* 90, 1–6.
- Tarpy, D.R., Lengerich, E.J., Pettis, J.S., et al., 2013. Idiopathic brood disease syndrome and queen events as precursors of colony mortality in migratory beekeeping operations in the eastern united states. *Preventive veterinary medicine* 108, 225–233.
- Tsujiuchi, S., Sivan-Loukianova, E., Eberl, D.F., Kitagawa, Y., Kadowaki, T., 2007. Dynamic range compression in the honey bee auditory system toward waggle dance sounds. *PLoS One* 2, e234.
- Visscher, P.K., 1986. Effect of location within the nest on acceptance of queen cells in honeybee colonies. *Journal of Apicultural Research* 25, 154–157.
- Weisstein, E.W., 2015. Fast fourier transform .
- Westerdahl, B.B., Gary, N.E., 1981. Longevity and food consumption of microwave-treated (2.45 ghz cw) honeybees in the laboratory. *Bioelectromagnetics* 2, 305–314.

- Yañez, O., Jaffé, R., Jarosch, A., Fries, I., Moritz, R.F., Paxton, R.J., de Miranda, J.R., 2012. Deformed wing virus and drone mating flights in the honey bee (*apis mellifera*): implications for sexual transmission of a major honey bee virus. *Apidologie* 43, 17–30.
- Zacepins, A., Brusbardis, V., Meitalovs, J., Stalidzans, E., 2015. Challenges in the development of precision beekeeping. *Biosystems Engineering* 130, 60–71.
- Zacepins, A., Karasha, T., 2013. Application of temperature measurements for the bee colony monitoring: a review, in: *Proceedings of the 12th International Scientific Conference Engineering for Rural Development*, pp. 126–131.

CHAPTER 7 Appendix A

Listing 7.1: Sample of data stored in MongoDB database .json format.

```

{"_id":{"_id":{"$oid":"55a5dac62252b40e5fa0d69b"}},
  "db":38.47,
  "dht_h":72.4000015258789,
  "dht_t":20.100000381469727,
  "hive_id":"hive6",
  "hz":125,
  "time":{"$date":"2015-07-15T00:00:06.925Z"}},
"type":"sample"}
{"_id":{"_id":{"$oid":"55a5dad02252b40e5fa0d6a0"}},
  "db":38.621,
  "dht_h":72.4000015258789,
  "dht_t":20.100000381469727,
  "hive_id":"hive6",
  "hz":125,
  "time":{"$date":"2015-07-15T00:00:16.524Z"}},
"type":"sample"}

```

Table 7–1: Sample data in .csv format.

time	hive id	dht t	dht h	hz	db
2015-07-15T00:00:06Z	"hive6"	20.10000038146973	72.40000152587891	125.0	38.47
2015-07-15T00:00:16Z	"hive6"	20.10000038146973	72.40000152587891	125.0	38.621

Listing 7.2: Example of data retrieval and conversion from .json to .csv.

```

mongoexport --db 20150715 --collection hive6 --fields 'time','hive_id',
'dht_t','dht_h','hz','db' --csv --out h11016.csv

```

CHAPTER 8

Appendix B

Listing 8.1: Hive-node software

```
1 # Libraries
2 import zmq
3 import ast
4 import json
5 import os
6 import sys
7 try:
8     import pyaudio
9 except Exception:
10 pass
11 import cherryypy
12 import numpy as np
13 import random
14 import urllib2
15 from datetime import datetime
16 from serial import Serial, SerialException
17 from ctypes import *
18 from cherryypy.process.plugins import Monitor
19 from cherryypy import tools
```

```

20 import logging
21 import socket
22 import cv2
23 from matplotlib.mlab import find
24 import pyaudio
25 import numpy as np
26 import math
27 import time
28 # Constants
29 try:
30     CONFIG_FILE = sys.argv[1]
31 except Exception as err:
32     CONFIG_FILE = None
33 # Error Handling
34 ERROR_HANDLER_FUNC = CFUNCTYPE(None, c_char_p, c_int, c_char_p, c_int, c_char_p)
35 def py_error_handler(filename, line, function, err, fmt):
36     pass
37 C_ERROR_HANDLER = ERROR_HANDLER_FUNC(py_error_handler)
38 # Node
39 class HiveNode:
40     ## Initialize
41     def __init__(self, config):
42     # Configuration
43     if not config:

```

```
44 self.REBOOT_ENABLED = False
45 self.ZMQ_ENABLED = True
46 self.ZMQ_SERVER = "tcp://192.168.0.100:1980"
47 self.ZMQ_TIMEOUT = 5000
48 self.MICROPHONE_ENABLED = True
49 self.MICROPHONE_CHANNELS = 1
50 self.MICROPHONE_RATE = 16000
51 self.MICROPHONE_CHUNK = 2048
52 self.MICROPHONE_FORMAT = pyaudio.paInt16
53 self.MICROPHONE_RECORD_SECONDS = 5
54 self.MICROPHONE_LOWPASS = 880 # hz
55 self.MICROPHONE_HIGHPASS = 110
56 self.CAMERA_ENABLED = False
57 self.CAMERA_INDEX = 0
58 self.DHT_PIN = 4
59 self.CHERRYPY_PORT = 8081
60 self.CHERRYPY_ADDR = "0.0.0.0"
61 self.PING_INTERVAL = 1
62 self.LOG_ENABLED = True
63 self.LOG_FILE = "log.txt"
64 else:
65 self.load_config(config)
66 # System Globals
67 self.HIVE_ID = socket.gethostname()
```

```

68 self.NODE_DIR = os.path.dirname(os.path.abspath(__file__))
69 # Mandatory Initializers
70 self.init_tasks()
71 # Optional Initializers
72 if self.ZMQ_ENABLED:
73     self.init_zmq()
74 if self.LOG_ENABLED:
75     self.init_logging()
76 if self.DHT_ENABLED:
77     self.init_DHT()
78 if self.MICROPHONE_ENABLED:
79     self.init_mic()
80 ## Load Config File
81 def load_config(self, config):
82     self.log_msg('CONFIG', 'Loading Config File')
83     with open(config) as config_file:
84         settings = json.loads(config_file.read())
85     for key in settings:
86         try:
87             getattr(self, key)
88         except AttributeError as error:
89             self.log_msg('CONFIG', '%s : %s' % (key, str(settings[key])))
90             setattr(self, key, settings[key])
91 ## Initialize tasks

```

```

92 def init_tasks(self):
93     self.log_msg('ENGINE', 'Initializing cherrypy monitor tasks ...')
94     try:
95         Monitor(cherrypy.engine, self.update, frequency=self.PING_INTERVAL).subscribe()
96     except Exception as error:
97         self.log_msg('ENGINE', 'Error: %s' % str(error))
98     ## Initialize ZMQ messenger
99     def init_zmq(self):
100         self.log_msg('ZMQ', 'Initializing ZMQ client ...')
101         try:
102             self.context = zmq.Context()
103             self.socket = self.context.socket(zmq.REQ)
104             self.socket.connect(self.ZMQ_SERVER)
105             self.poller = zmq.Poller()
106             self.poller.register(self.socket, zmq.POLLIN)
107             msg = 'OK'
108         except Exception as error:
109             msg = 'Error: %s' % str(error)
110             self.log_msg('ZMQ', msg)
111         ## Initialize Logging
112         def init_logging(self):
113             self.log_msg('LOG', 'Initializing logging ...')
114             try:
115                 logging.basicConfig(filename=self.LOG_FILE, level=logging.DEBUG)

```

```

116 msg = 'OK'
117 except Exception as error:
118 msg = 'Error: %s' % str(error)
119 self.log_msg('LOG', msg)
120 ## Initialize DHT Sensor
121 def init_DHT(self):
122 self.log_msg('DHT', 'Initializing DHT Sensor')
123 try:
124 pass
125 except Exception as error:
126 self.log_msg('DHT', 'Error: %s' % str(error))
127 ## Initialize audio
128 def init_mic(self):
129 """ part of revised code for audio processing """
130 self.log_msg('MIC', 'Initializing mic ...')
131 # Start audio stream
132 try:
133 self.p = pyaudio.PyAudio()
134 self.microphone = self.p.open(
135 format = self.MICROPHONE_FORMAT,
136 channels = self.MICROPHONE_CHANNELS,
137 rate = self.MICROPHONE_RATE,
138 input = True,
139 frames_per_buffer = self.MICROPHONE_CHUNK)

```

```

140 except Exception as e:
141     self.log_msg("MIC", "ERROR: %s" % str(e))
142     ## Close Microphone
143     def close_mic(self):
144         """cleanly back out and release sound card."""
145         self.microphone.stop_stream()
146         self.p.close(self.microphone)
147         # Capture Audio
148         def capture_audio(self, trimBy=10):
149             self.log_msg('MIC', 'Capturing audio ...')
150             db = None
151             hz = None
152             try:
153                 # Capture Audio and convert to numeric
154                 audio = []
155                 for i in range(0, (self.MICROPHONE_RATE / self.MICROPHONE_CHUNK)
156                     * self.MICROPHONE_RECORD_SECONDS):
157                     try:
158                         audioString = self.microphone.read(self.MICROPHONE_CHUNK)
159                         audioNumeric = np.fromstring(audioString, dtype=np.int16)
160                         audio.append(audioNumeric)
161                     except IOError as e:
162                         pass
163                 # Calculate Pitch

```

```

164 self.log_msg('MIC', 'Calculating dominant frequencies ...')
165 pitch = []
166 for signal in audio:
167     crossing = [math.copysign(1.0, s) for s in signal]
168     index = find(np.diff(crossing));
169     f0 = round(len(index) * self.MICROPHONE_RATE / (2.0 * np.prod(len(signal))), 2)
170     pitch.append(f0)
171     pitch = np.array(pitch)
172     pitch_bandpass = pitch[np.logical_and
173         (pitch < self.MICROPHONE_LOWPASS, pitch > self.MICROPHONE_HIGHPASS)]
174     hz = np.median(pitch_bandpass)
175     # Calculate Decibels
176     self.log_msg('MIC', 'Calculating average decibel level ...')
177     left,right=np.array_split(np.abs(np.fft.fft(audio)),2)
178     db = np.add(left,right[::-1])
179     db = np.multiply(20,np.log10(db)) # db
180     db = np.round(np.mean(np.multiply(20,np.log10(db))), 3) # convert to dB
181     except Exception as error:
182     self.log_msg('MIC', 'Error: %s' % str(error))
183     result = { "db" : db, "hz" : hz}
184     self.log_msg('MIC', 'OK: %s' % str(result))
185     return result
186     ## Read DHT (if available)
187     def read_DHT(self, pin=4):

```

```

188 self.log_msg('DHT', 'Reading from DHT ...')
189 try:
190     sensor=Adafruit_DHT.DHT22
191     humidity, temperature = Adafruit_DHT.read_retry(sensor, pin, retries=5, delay_seconds=1)
192     result = {
193         "dht_t" : temperature,
194         "dht_h" : humidity
195     }
196     self.log_msg('DHT', 'OK: %s' % str(result))
197 except Exception as error:
198     result = {}
199     self.log_msg('DHT', 'Error: %s' % str(error))
200     return result
201     ## Send sample to aggregator
202     def zmq_sample(self, sample):
203         self.log_msg('ZMQ', 'Pushing to aggregator ...')
204         try:
205             dump = json.dumps(sample)
206             self.socket.send(dump)
207             socks = dict(self.poller.poll(self.ZMQ_TIMEOUT))
208             if socks:
209                 if socks.get(self.socket) == zmq.POLLIN:
210                     dump = self.socket.recv(zmq.NOBLOCK)
211                     response = json.loads(dump)

```

```

212 self.log_msg('ZMQ', str(response))
213 result = response
214 else:
215 self.log_msg('ZMQ', 'Error: Poll Timeout')
216 result = None
217 else:
218 self.log_msg('ZMQ', 'Error: Socket Timeout')
219 except Exception as error:
220 self.log_msg('ZMQ', 'Error: %s' % str(error))
221 return result
222 ## Generate blank sample
223 def blank_sample(self):
224 sample = {
225 'type' : 'sample',
226 'time' : datetime.strftime(datetime.now(), '%Y-%m-%d %H:%M:%S'),
227 'hive_id' : self.HIVE_ID
228 }
229 return sample
230 ## Update Clock
231 def update_clock(self, secs):
232 self.TIME_ERROR = time.time() - secs
233 self.log_msg('CLOCK', 'dt=%d' % secs)
234 ## Log Message
235 def log_msg(self, task, msg):

```

```

236 date = datetime.strftime(datetime.now(), '%d/%b/%Y:%H:%M:%S')
237 print('[%s] %s %s' % (date, task, msg))
238 ## Shutdown
239 def shutdown(self):
240     self.log_msg('ENGINE', 'Shutting Down')
241     try:
242         self.arduino.close()
243     except Exception as e:
244         self.log_msg('CTRL', str(e))
245     try:
246         self.microphone.close()
247     except Exception as e:
248         self.log_msg('MIC', str(e))
249     try:
250         self.camera.release()
251     except Exception as e:
252         self.log_msg('CAM', str(e))
253     os.system("sudo reboot")
254 ## Update to Aggregator
255 def update(self):
256     # Blank Sample
257     sample = self.blank_sample()
258     # Mic
259     if self.MICROPHONE_ENABLED:

```

```

260 microphone_result = self.capture_audio()
261 sample.update(microphone_result)
262 # DHT
263 if self.DHT_ENABLED:
264     DHT_result = self.read_DHT()
265     sample.update(DHT_result)
266 # ZMQ Push/Pull Handler
267 try:
268     response = self.zmq_sample(sample)
269     if response['type'] == 'clock':
270         self.log_msg('ZMQ', 'Caught time update request', '')
271         self.update_clock['secs']
272     if response['type'] == 'config':
273         self.log_msg('ZMQ', 'Caught Reload Config Request', '')
274 except:
275     if self.REBOOT_ENABLED:
276         self.shutdown()
277 ## Render Index
278 @cherryypy.expose
279 def index(self):
280     with open('static/index.html') as html:
281         return html.read()
282 # Main
283 if __name__ == '__main__':

```

```
284 node = HiveNode(config=CONFIG_FILE)
285 currdir = os.path.dirname(os.path.abspath(__file__))
286 cherrypy.server.socket_host = node.CHERRYPY_ADDR
287 cherrypy.server.socket_port = node.CHERRYPY_PORT
288 conf = {
289     '/': {'tools.staticdir.on':True,
290         'tools.staticdir.dir':os.path.join(currdir,'static')},
291     '/data': {'tools.staticdir.on':True,
292              'tools.staticdir.dir':os.path.join(currdir,'data')},
293     '/js': {'tools.staticdir.on':True,
294            'tools.staticdir.dir':os.path.join(currdir,'static','js')},
295 }
296 cherrypy.quickstart(node, '/', config=conf)
```

CHAPTER 9 Appendix C

Listing 9.1: Hive-aggregator software

```
1 # Libraries
2 import json
3 import ast
4 import cherryPy
5 import os
6 import sys
7 import numpy as np
8 from datetime import datetime, timedelta
9 from cherryPy.process.plugins import Monitor
10 from cherryPy import tools
11 from pymongo import MongoClient
12 from bson import json_util
13 import zmq
14 from sklearn import svm
15 # Constants
16 try:
17     CONFIG_FILE = sys.argv[1]
18 except Exception as err:
19     CONFIG_FILE = None
```

```

20  ## Pretty Print
21  def pretty_print(task, msg):
22      date = datetime.strftime(datetime.now(), '%d/%b/%Y:%H:%M:%S')
23      print('[%s] %s %s' % (date, task, msg))
24  # HiveAggregator CherryPy server
25  class HiveAggregator:
26      ## Initialize
27      def __init__(self, config_path):
28          # Configuration
29          if not config_path:
30              self.USER_ID = "evan"
31              self.AGGREGATOR_ID = "MAA"
32              self.ZMQ_SERVER = "tcp://*:1980"
33              self.CHERRYPY_LISTEN_INTERVAL = 0.1
34              self.CHERRYPY_BACKUP_INTERVAL = 1500
35              self.CHERRYPY_CHECK_INTERVAL = 1500
36              self.CHERRYPY_PORT = 8080
37              self.CHERRYPY_ADDR = "0.0.0.0"
38              self.MONGO_ADDR = "127.0.0.1"
39              self.MONGO_PORT = 27017
40              self.MONGO_DB = "%Y%m%d"
41              self.TIME_FORMAT = "%Y-%m-%d %H:%M:%S"
42              self.DATA_PATH = "data/"
43              self.LOGS_FILE = "logs.json"

```

```

44 self.SAMPLES_FILE = "samples.json"
45 self.CSV_FILE = "samples.csv"
46 self.ALL_PARAMETERS = ["time","dht_t","dht_h","hz","db","volts","amps"]
47 else:
48 self.load_config(config_path)
49 # Initializers
50 self.init_zmq()
51 self.init_tasks()
52 self.init_mongo()
53 ## Load Configuration
54 def load_config(self, config_path):
55 pretty_print('CONFIG', 'Loading Config File')
56 with open(config_file) as config:
57 settings = json.loads(config.read())
58 for key in settings:
59 try:
60 getattr(self, key)
61 except AttributeError as error:
62 print('\t' + key + ' : ' + str(settings[key]))
63 setattr(self, key, settings[key])
64 ## Initialize ZMQ
65 def init_zmq(self):
66 pretty_print('ZMQ', 'Initializing ZMQ')
67 try:

```

```

68 self.context = zmq.Context()
69 self.socket = self.context.socket(zmq.REP)
70 self.socket.bind(self.ZMQ_SERVER)
71 except Exception as error:
72 pretty_print('ERROR', str(error))
73 ## Initialize Tasks
74 def init_tasks(self):
75 pretty_print('CHERRYPY', 'Initializing Monitors')
76 try:
77 Monitor(cherrypy.engine, self.listen,
78 frequency=self.CHERRYPY_LISTEN_INTERVAL).subscribe()
79 Monitor(cherrypy.engine, self.backup,
80 frequency=self.CHERRYPY_BACKUP_INTERVAL).subscribe()
81 Monitor(cherrypy.engine, self.check,
82 frequency=self.CHERRYPY_CHECK_INTERVAL).subscribe()
83 except Exception as error:
84 pretty_print('ERROR', str(error))
85 ## Initialize MongoDB
86 def init_mongo(self):
87 pretty_print('MONGO', 'Initializing Mongo')
88 try:
89 self.mongo_client = MongoClient(self.MONGO_ADDR, self.MONGO_PORT)
90 except Exception as error:
91 pretty_print('ERROR', str(error))

```

```

92  ## Query Samples in Range to JSON-file
93  def query_db(self, days, query_type):
94  pretty_print('MONGO', 'Querying samples for last %s days' % str(days))
95  result = []
96  for d in range(days):
97  date = datetime.now() - timedelta(days = d)
98  db_name = datetime.strftime(date, self.MONGO_DB)
99  mongo_db = self.mongo_client[db_name]
100  for name in mongo_db.collection_names():
101  if not name == 'system.indexes':
102  matches = mongo_db[name].find({'type':query_type})
103  for sample in matches:
104  sample['time'] = datetime.strftime(sample['time'], self.TIME_FORMAT)
105  result.append(sample)
106  return result
107  ## Dump tp JSON
108  def dump_json(self, results, filename):
109  with open(self.DATA_PATH + filename, 'w') as jsonfile:
110  dump = json_util.dumps(results, indent=4)
111  jsonfile.write(dump)
112  ## Receive Sample
113  def receive_message(self):
114  pretty_print('ZMQ', 'Receiving Message')
115  try:

```

```

116 packet = self.socket.recv()
117 message = json.loads(packet)
118 pretty_print('ZMQ', 'OKAY: %s' % str(message))
119 return message
120 except Exception as error:
121 pretty_print('ERROR', str(error))
122 ## Store Sample
123 def store_sample(self, sample):
124 pretty_print('MONGO', 'Storing Sample')
125 try:
126 sample['time'] = datetime.now()
127 # this is the mongo db it saves to
128 db_name = datetime.strftime(datetime.now(), self.MONGO_DB)
129 mongo_db = self.mongo_client[db_name]
130 hive = mongo_db[sample['hive_id']]
131 sample_id = hive.insert(sample)
132 pretty_print('MONGO', 'Sample ID: %s' % str(sample_id))
133 return str(sample_id)
134 except Exception as error:
135 pretty_print('ERROR', str(error))
136 ## Store Log
137 def store_log(self, log):
138 pretty_print('MONGO', 'Storing Log')
139 try:

```

```

140 log['time'] = datetime.now()
141 hive = self.mongo_db[sample['hive_id']]
142 log_id = hive.insert(log)
143 pretty_print('MONGO', 'Log ID: %s' % str(log_id))
144 return str(log_id)
145 except Exception as error:
146 pretty_print('ERROR', str(error))
147 ### Send Response
148 def send_response(self, status, sample_id):
149 pretty_print('ZMQ', 'Sending Response to Hive')
150 try:
151 response = {
152 'id' : sample_id,
153 'status' : status,
154 'type' : 'response',
155 'time' : datetime.strftime(datetime.now(), self.TIME_FORMAT),
156 }
157 dump = json.dumps(response)
158 self.socket.send(dump)
159 pretty_print('ZMQ', str(response))
160 except Exception as error:
161 pretty_print('ERROR', str(error))
162 ##Periodic Functions
163 # Listen for Next Sample

```

```

164 def listen(self):
165     pretty_print('CHERRYPY', 'Listening for nodes')
166     message = self.receive_message()
167     if message['type'] == 'sample':
168         sample_id = self.store_sample(message)
169         if sample_id:
170             status = 'ok'
171         else:
172             status = 'bad'
173         self.send_response(status, sample_id)
174     ## Backup
175     def backup(self):
176         pretty_print('CHERRYPY', 'Backing up data')
177     ## Check Database
178     def check(self):
179         pretty_print('CHERRYPY', 'Checking database')
180     ##Handler Functions
181     # Render Index
182     @cherry.py.expose
183     def index(self):
184         html = open('static/index.html').read()
185         return html
186     ## Handle Posts
187     #This function is basically the API

```

```

188 @cherry.py.expose
189 def default(self, *args, **kwargs):
190     try:
191         kwargs['type']
192         if kwargs['type'] == 'log':
193             self.store_sample(kwargs)
194         elif kwargs['type'] == 'graph':
195             results = self.query_db(int(kwargs['range_select']), 'sample')
196             self.dump_json(results, self.SAMPLES_FILE)
197         elif kwargs['type'] == 'save':
198             results = self.query_db(int(kwargs['range_select']), 'sample')
199             self.dump_csv(results, self.CSV_FILE)
200         else:
201             pass
202         except Exception as err:
203             pretty_print('ERROR', str(err))
204         return None
205     # Main
206     if __name__ == '__main__':
207         aggregator = HiveAggregator(CONFIG_FILE)
208         cherry.py.server.socket_host = aggregator.CHERRYPY_ADDR
209         cherry.py.server.socket_port = aggregator.CHERRYPY_PORT
210         currdir = os.path.dirname(os.path.abspath(__file__))
211         conf = {

```

```
212  '/': {'tools.staticdir.on':True,
213       'tools.staticdir.dir':os.path.join(currdir,'static')},
214  '/data': {'tools.staticdir.on':True,
215            'tools.staticdir.dir':os.path.join(currdir,'data')},
216  }
217  cherrypy.quickstart(agggregator, '/', config=conf)
```
