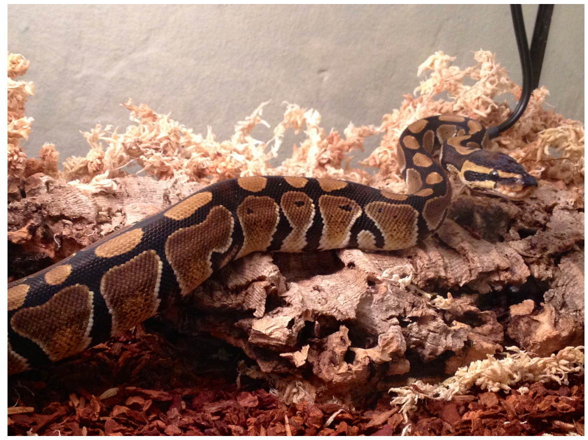
HappyPython.py Final Report

Hayley Proctor | David Rothblum | Kyle Walden | Isaac Williams

I. Abstract

Using Calvin the Friendly Ball Python and his terrarium environment as our system, we can monitor and actuate his living conditions to ensure that they are both hospitable and efficient. By successfully implementing temperature and humidity controls in Calvin's terrarium, we are able to provide him with a more comfortable existence, while reducing the costs associated with maintaining his environment. Housing a cold-blooded reptile adds a layer of complexity to our system, as Calvin must be provided with two temperature levels in his environment to allow him to equilibrate his internal temperature.

Calvin's environment must have to two zones with temperatures between 87-92 degrees Fahrenheit and 77-82 degrees Fahrenheit respectively, and maintain humidity levels above 55%. Temperature can be monitored and actuated by using both a thermostatic model and model predictive control methods. Implementing a cyber-physical system to monitor and actuate these environmental conditions, while providing real-time data and feedback will allow



Picture 1: Calvin in his "natural" environment

Calvin's environment to be kept both ideal and consistent.

II. Introduction

a. Motivation and Background

Initially, we considered actuating a much larger environment, such as a large room. However, by using a snake terrarium—an environment that we have unfettered access to—we have a much smaller and more controlled environment with which to initially create and test our system. In order to transfer this system of sensors and actuators to a separate environment, we would only need to expand our system, collect new data, and set ideal conditions. Because of this, a terrarium-based cyber-physical system is an ideal control system, which can then be extrapolated to larger environments. Comfort is a large concern for environment control in residential and commercial buildings, and the methods of maintaining a snake's comfort in a terrarium provide a similar challenge, albeit at a smaller and more precise scale. In particular, the act of maintaining two separate temperatures in a small connected space without any boundaries may be valuable in future efforts at room comfort.

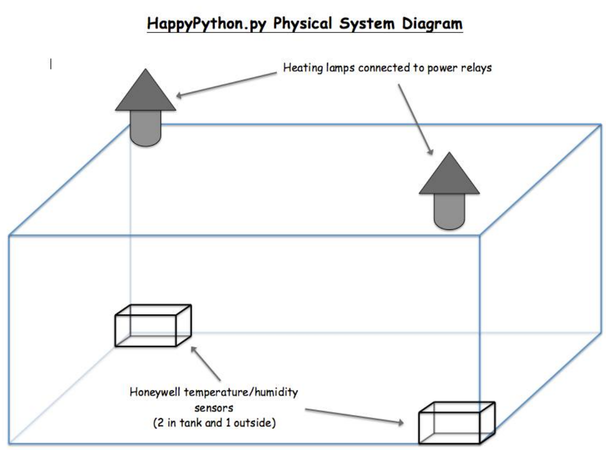


Diagram 1: Terrarium Sensor Setup

For this specific system, the main challenges are creating models for the temperature and the humidity within the terrarium. Therefore, we collected data to observe how the actuating devices affect the environment and implement two temperature and humidity readings in the terrarium: one for each zone of the tank. In addition to these sensors, an ambient sensor is used to record data and act as a factor in our model predictive control to account for the heat transfer that would naturally occur. In order to keep a snake happy, there must be a "hot" (87-92°F) and "cold" (77-82°F) side of the tank, and the humidity should be no lower than 55% at anytime.

Employing the completed model from our data, the system checks temperature and humidity levels, and actuates the heat lamps accordingly. When the temperature varies from ideal ranges, the system will actuate the heat lamps on or off to restore it to optimal conditions. One of the main challenges concerning humidity is determining the extent of the effects from temperature. The humidity levels in Calvin's tank are consistently monitored, and, should the system go out of balance, a pop-up notification window is generated on the website to notify the owner of the problem in their system. It is a gentle reminder because humidity issues are typically less pressing than temperature in determining comfort.

b. Relevant Literature

"Care Sheet for Ball Pythons". World of Ball Pythons.[1] — A reference for suitable ball python conditions

DeWitt, Zoltan, and Matthew Roeschke. "Optimal Refrigeration Control for Soda Vending Machines." *UC Berkeley* May 2015. — A report that provides a foundation for the

---

[1] http://www.worldofballpythons.com/python-regius/care-sheet/

modeling principles applied to our terrarium environment.

c. Focus of Study

Maintaining appropriate environmental conditions can be of paramount importance in a wide range of scenarios, hence the focus of our cyber-physical system. Specific to our infrastructure, temperature monitoring is very necessary when examining energy use. As noted, Calvin must exist in very specific temperature ranges, and it is easy to imagine an overuse of energy if said ranges are not consistently monitored and controlled. If the heat lamps used are on for the minimum time necessary to maintain our desired ranges, we can certainly expect to minimize the energy used to keep Calvin comfortable while maximizing total comfort.

Another aspect worth considering as we attempt to maintain a comfortable environment for Calvin is the ambient temperature surrounding the terrarium, as outside temperatures will influence the heating needs of the system. By monitoring and determining the impact of the ambient temperature, we can examine yet another influence on the cost of Calvin's precise environmental needs.

The monitoring and control techniques implemented on Calvin's terrarium can be scaled and modified to apply to most, if not all, imaginable environments. As such, energy consumption issues related to the maintenance of comfortable environments could be mitigated to some degree.

IV. Technical Description

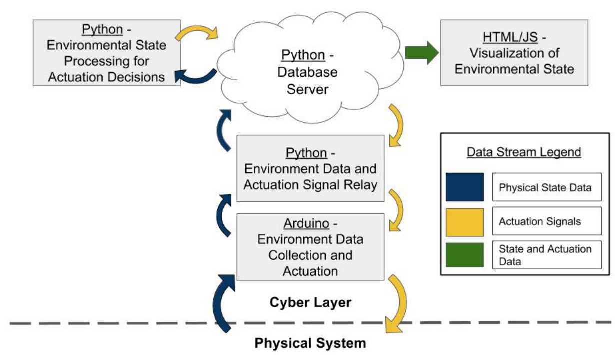


Diagram 2: CPS Architecture

a. Cyber System Code

1. Arduino

SensorNodeCode.ino — An Arduino file uploaded to the Arduino Mini boards in each sensor node collected temperature and humidity data from the sensors and sent it unmodified through their attached Xbees set as end devices.

Coordinator.ino — An Arduino file uploaded to the central Arduino Uno, which used an Xbee set as a coordinator to collect each piece of data sent through the end device Xbees and send this data as packets to ListenAndSendProject.py. Furthermore, this file received actuation data to control the power relays connected to the heat lamps.

2. Python

ListenAndProcessProject.py — Responsible for collecting and processing data from the database server, determining the subsequent system state, and sending actuation signals back to the server.

ListenAndSendProject.py — Responsible for collecting data from the Arduino Uno, transmitting data to the database server to be processed, and relaying the actuation signals from the server to the Arduino Uno.

Nanoserver.py — Creates the SQLite database and hosts a local website for the data to be posted to and taken from. This is the base for our website's functionality.

3. Matlab

HappyPython.m — Performs multivariate regression on the collected data. Half of the collected data is used to compute appropriate regression coefficients, and the other half of the data is used to confirm the accuracy of said regression coefficients. Regression is then performed on the entire dataset to compute coefficients that are even more representative. These coefficients provide the ability to predict the future temperature in either the warm or cold zone, based on the current temperatures and light states.

MPC.m — A Matlab function file written to interact with Python and perform a model predictive control on the system. The file is intended to pull a one timestep worth of data from the system, and then determine the ideal next step for each light, while considering n future timesteps. Issues arose in the indexing and constraining of the function, and ultimately, the function ended up not being implemented.

Actuate.m — A Matlab function file written to interact and perform a slightly simpler model predictive control on the system, over just two future timesteps. Again, issues arose here, and the function ended up not being implemented.

4. HTML

Nanodashboard.html — Includes all the text, pictures, and data that should be loaded based on where the user clicks on the website. Also responsible for the humidity popup window. and calculating if Calvin is happy with his environment and visualizing it for the user.

Dashboard.js — Sets up basic layout for the website by calling on bootstrap. Makes sidebar functionality.

b. Mathematical Description of Data Analysis

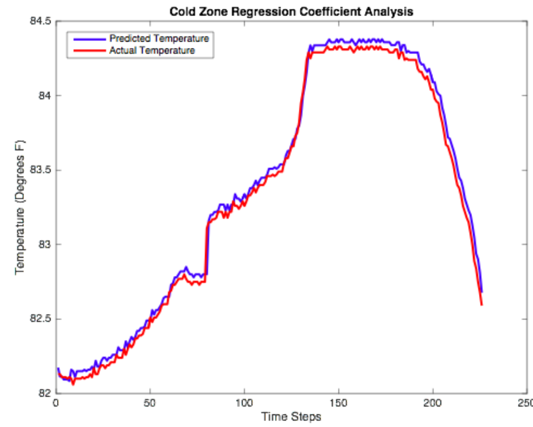
After collecting appropriate amounts of data across a wide range of states, various methods of analysis were implemented. Initially, a thermostatic model was employed. This model compared the collected data to upper and lower bounds to determine how the system should develop. If temperatures existed outside the set bounds, actuation signals prompted the heat lamps to turn on or off to influence the temperature towards the appropriate range. If the temperatures existed within the set bounds, the system maintained the state that existed previously.

Additionally, multivariate regression was performed on the collected data, to establish a predictive model for the temperature of the warm and cold zones, as a function of all three recorded temperatures and the states of both heat lamps. The calculated regression coefficients were implemented to predict the development of the temperatures in a different dataset, and proved to provide a representative model of the system. As you can see from the coefficients below, the temperature in each side of the tank is dependent on its respective temperature from the previous step; additionally, the heat in each side is almost entirely generated from the hot lamp. The small size of the terrarium is responsible for this limitation. In a larger environment, this regression analysis would have been less trivial.

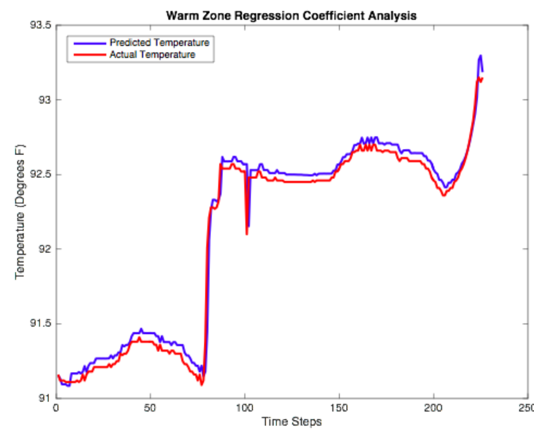
$$T_w(k+1) = \theta_1 * T_w(k) + \theta_2 * T_c(k) + \theta_3 * T_a(k) + \theta_4 * L_w(k) + \theta_5 * L_c(k)$$

$$T_c(k+1) = \theta_6 * T_w(k) + \theta_7 * T_c(k) + \theta_8 * T_a(k) + \theta_9 * L_w(k) + \theta_{10} * L_c(k)$$

$$\theta = [0.9897 - 0.0045 \;\; 0.0180 \;\; - 0.1099 \;\; - 0.0478 \;\; -0.0015 \;\; 1.0027 \;\; - 0.0007 \;\; - 0.0186 \;\; - 0.0298\,]$$



Graph 1: Cool Zone Temperature Predictions from Regression Coefficients Compared to Actual Temperature Development



Graph 2: Warm Zone Temperature Predictions from Regression Coefficients Compared to Actual Temperature Development

Model predictive control was posited as a useful form of analysis and control, and while it is mathematically more intensive, an attempt at implementing such control was made. Using the established regression model, along with the set bounds, a mixed-integer quadratic programming optimization problem was created to represent the system and its states. If implemented successfully, the control process determined the actuation signals based on the optimization of not only the next timestep, but of n subsequent timesteps.

c. Visualization Overview

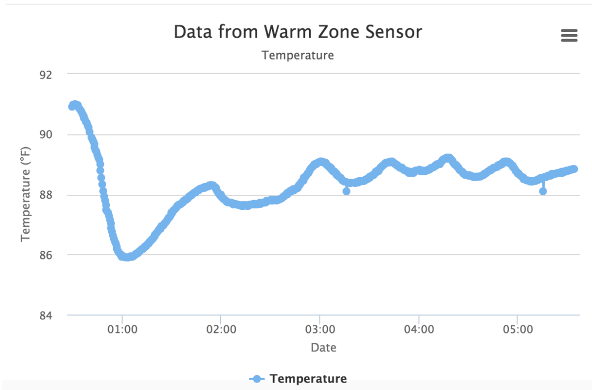
The website is the main center of data for this project; the website allows the user to interact with the cyber-physical system and interpret the data that has been collected. The links on the sidebar are Project, Status, Bio, Reports, Plots, and Export. Initially, when the page is accessed, the user will be directed to the Project tab; at this time, if the humidity in the tank is too low, the user will see a popup window that advises the user to manually apply a few sprays of water to Calvin's tank. The Project main page displays a copy of our poster with project overview information. The Status page shows the user whether Calvin is happy or sad with easy-to-interpret pictures; if three of the four necessary conditions are met in the terrarium, we assume that Calvin is happy. The Bio tab is a for-fun page with pictures of ball pythons in hats; we built this to practice implementing HTML visualization. A link to our Project Proposal and this Final Report are available on the Reports tab. The Plots tab visualizes the data that is being live-posted to the DBNanoServer. This provides the user with a plot to interpret the temperature evolution. The Export tab allows the user to extract this data for other purposes-like our Matlab regression.

Table 1: Bill of Materials

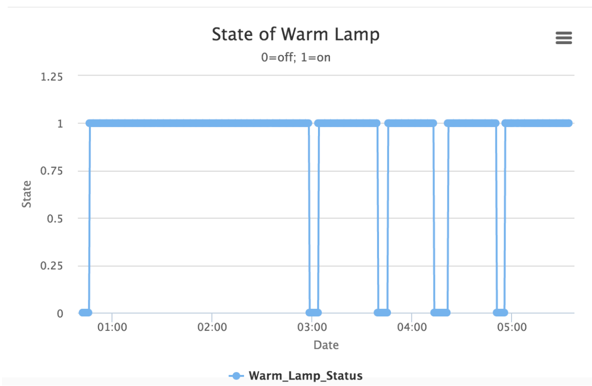
Quantity	Material	Source
3	Power Relay	CE 186 Supplied
3	Honeywell Sensor Node	CE 186 Supplied
3	Arduino Mini	CE 186 Supplied
1	Arduino Uno	CE 186 Supplied
4	Xbee Radios	CE 186 Supplied
2	Ceramic Heat Lamps	
1	Ball Python named Calvin	

V. Discussion

Heating indoor environments is one of the most energy intensive processes for buildings today. As a result, finding more reliable and efficient methods of maximizing comfort is in the interest of reducing electricity consumption without sacrificing quality of living conditions. Our project sought to use a small-scale and well-controlled environment as a method of modeling the behavior of a large-scale room. It also added the challenge of creating two separate zones within the same environment without boundaries, which is analogous to individualizing room comfort. A model like this could potentially be used for creating individualized cubicle environments in an office space, maintaining separate climates among rooms or floors in a house, or acting as more accurate climate control in a car.



Graph 3: Warm Zone Temperature Data from Thermostatic Model Implementation



Graph 4: Warm Zone Light State during Thermostatic Model Implementation

Thermostatic controls are a sufficient base for attaining these goals in many scenarios; in fact, during testing of our terrarium system, the system actuated the temperature zones with great success (see Graphs 1 and 2, above). With sufficient sensors for all zones and heating sources, bounds can be established and maintained. Our small-scale model demonstrates that fluctuations in conditions do not preclude the existence of predictable behavior in an environment, and, in any system, these fluctuations can be controlled to provide a comfortable environment solely based on thermostatic bounds.

There is, however, room for further refinement though regression modeling and model predictive control. Regression modeling affords the user the ability to accurately predict how their environment will evolve, as the methods used on our physical system can transfer to any imaginable set of data collected in an environment. Model predictive control provides an even “smarter” method of analyzing and actuating an environment, as it optimizes each subsequent actuation on a set number of future states, which allows a system to evolve more efficiently. Accordingly, energy usage is minimized, while the system is encouraged to stay as close as possible to the desired temperatures. This is common practice in “smart” thermostats, and it can play an important role in creating a more efficient and accurate model than thermostatic control.

Finally, the ability to present the data on a website with a logical interface and clear graphs presenting trends in the system helps to ease the use of the system. A glance at the graphs allows simple understanding of trends in the system and what adjustments need to be made to tailor the environment to the needs of the user.

Picture 2: The 5 members of our group at the end-of-semester symposium. From left to right: Kyle, Isaac, Calvin, Hayley, and Dave.

VI. Summary

Our project aimed to create the ideal living conditions in a terrarium for a ball python. This required monitoring and adjustments of temperature and humidity; temperature was especially important as ball pythons require access to two temperature zones to allow them to actuate their internal temperatures. Maintaining two temperature profiles without any boundaries between the two zones is a concept that is applicable to many other large-scale environments, such as cars and office buildings. Furthermore, using forms of thermostatic control, regression modeling, and model predictive control allows energy efficiency to be addressed without sacrificing comfort.

We were able to successfully implement this system inside Calvin's terrarium. This project acts as a proof-of-concept for other larger systems to which this system may be applied. By applying this system to larger environments, large amounts of energy could be saved, while still maintaining desired comfort levels.