Sous Vide Project

John Lucero & Katie Ayala

PHYS 129L

Abstract

The SousVide.py program creates a user interface to a Proportional Integral Derivative (PID)/phase angle controlled heating element. This hardware/software combinations allows a user to precisely control the temperature of a water bath.

1 Introduction

Sous Vide is a method of precision cooking in which food is vacuum-sealed and placed in a water bath for longer than normal cooking times at an accurately regulated temperature. Our project goal is to build a functioning Sous Vide device that maintains water at a constant, user specified temperature. This project was chosen because it presents unique challenges that stem from precisely heating and maintaining water temperature. Temperature regulation and the controlled application of power to resistive loads are of particular interest in the engineering of devices used for various physics related tasks.

2 Software

Necessary software for the SousVide.py program includes the RPi.GPIO module, the w1thermsensor module, the Adafruit Python SSD1306 library, the Python PID module, the Python Imaging Library, and the NumPy library. This software must be installed for the program to operate.

The RPi.GPIO module is installed by default in Raspbian. To make sure that it is at the latest version:

```
$ sudo apt-get update
$ sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

To install the w1thermsensor module:

```
$ sudo apt-get install python3-w1thermsensor
```

To install the Adafruit Python SSD1306 library:

```
$ git clone https://github.com/adafruit/Adafruit_Python_SSD1306.git
$ cd Adafruit_Python_SSD1306
$ sudo python setup.py install
```

To install the Python PID module:

```
$ git clone git://github.com/sandves/pid.git
```

After download, the file containing the PID module must be placed in the same directory as the main program.

To install the Python Imaging Library (PIL):

```
$ sudo apt-get install python-imaging
```

To install NumPy:

```
$ python -m pip install --user numpy
```

3 External Hardware

Part	Description	Qty	Source
2W 1.5k ohm resistor		1	Mouser
2W 15K ohm resistor		2	Mouser
2W 2.4k ohm resistor		1	Mouser
2W 180 ohm resistor		1	Mouser
1/4W 1k ohm resistor		1	Santa Barbara Electronics
1/4W 220 ohm resistor		1	Santa Barbara Electronics
0.01uF capacitor	filtering capacitor	1	Mouser
H11AA1	zero cross detection optocoupler	1	Mouser
MOC3021	TRIAC firing optocoupler	1	Mouser
Q6015L5	15A TRIAC	1	Mouser
2N5551	NPN transistor	1	Mouser
634-20ABPE	TRIAC heat sink	1	Mouser
709-IRM10-5	5V 2A 10W power supply	1	Mouser
DS18B20	temperature sensor	1	Adafruit
Pi Zero		1	Adafruit
Proto Bonnet		1	Adafruit
12mm ESUPPORT button	pack of 5	1	Amazon
Tolako 5V Relay		1	Amazon
EWP-3502HT6V	high temp pump, food grade	1	Amazon
1500W Heat Element		1	Amazon
0.96” OLED 128x64	SSD1306 driver	1	Amazon

4 Program

The program uses an OLED display and 4 buttons for all user interaction, as shown in Figure 1. From left to right, the buttons are: up, down, enter, and a back button (which was discovered not to be necessary for this project).

Figure 1: User interface display and push buttons.

During use, the program transitions between two phases. The first phase ensures the device is safely set up and ready for use. The second, or operational phase, heats the water to a user specified set point between 100-212°F. Due to time sensitivity and the number of operations, threading is employed in the second phase of the SousVide.py program.

The initial phase presents the user with safety questions to ensure the heating element is submerged and the pump is primed. If the response to a question is *no*, the user is asked to take appropriate action and the question is repeated until the response is *yes*. The second phase of the program is prevented from executing until *yes* is selected for both questions, signaling that it is safe to operate both the heater and pump.

Once the user indicates the device is safe for use, the transition into the operation phase takes place. In this phase, display operations, temperature readings, and AC control are done on individual threads. The main thread iterates through a while loop to perform PID calculations and determine a delay referenced by the AC control thread.

On the screen, the user is prompted to select a set temperature while simultaneously displaying the current water temperature by the display thread. The user is able to scroll through the temperature range with the up and down buttons; a selection is made with the enter button. If the user scrolls to a temperature different from the set temperature and does not press the enter button, the set temperature will be displayed again and will not be changed.

Temperature readings are taken in a different thread to minimize the main while loop iteration time. These readings are used by the main while loop and the display thread. The main while loop provides the set temperature and temperature reading to the PID.

4.1 PID

The PID module used in our program acts as a PID controller. It continuously calculates an error value $e(t)$ as the difference between the set temperature and the measured temperature and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively). The controller attempts to minimize the error over time by adjustment of a control variable $u(t)$. The overall control function can be expressed mathematically as

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt}$$

where K_p, K_i, and K_d, all non-negative, denote coefficients for the proportional, integral, and derivative terms and $e = Setpoint - Input$. The PID's output is used to determine a time delay in milliseconds for use by the AC control thread, which is responsible for controlling the power applied to the heating element.

4.2 AC Control

Phase angle control is applied to the AC input voltage to limit the power flow to the heating element. It works by modulating a thyristor (in this case, a triac) into and out of conduction at a predetermined phase of the applied waveform. The phase of the AC input modulation cycle at which the triac's gating signal is triggered is calculated based on the PID's output. In other words, the PID determines an appropriate delay time between the zero-crossing and the triggering of the triac gating signal. The delay time, or delayed firing angle (firing time each half-cycle), is continuously calculated and applied.

Once a gate pulse is sent to the triac, current begins to flow to the heating element. This current latches the triac to the on state until the current through it becomes zero at the zero-crossing. Voltage is supplied to the heating element for a fraction of the cycle, determined by how long the triac conducts during the half cycle.

The PID/phase angle controlled heating element is necessary to heat the water as quickly as possible to the set temperature without overshooting it, as well as maintain a consistent water temperature for long periods of time.

5 Results

Our fully assembled Sous Vide device properly submerged in a container of water is shown in Figure 2 and Figure 3. The device is capable of maintaining a constant water temperature of +/- 1°F. This was achieved by finely tuning the PID control terms, K_p, K_i, and K_d, through testing.

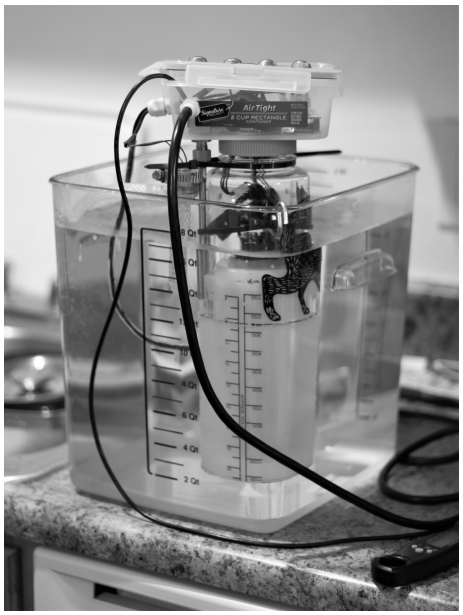

Figure 2: Completed Sous Vide device.

Figure 3: Sous Vide device in use.

Approximate values for the control terms K_p, K_i, and K_d for our program were initially entered, but were refined by introducing a temperature setpoint change and observing the system response. A balance of the control terms was achieved to produce an optimal control function (PID output), and to, therefore, optimize the performance of our device. The control term values derived are $K_p = 0.17$, $K_i = 0.25$, and $K_d = 0$.

In testing the device, the time it took to heat water from 70°F to 130°F was approximately 30 minutes, which is comparable to many mid-range commercially available units, though the temperature maintenance of our device was not able to match that of such units (+/- 0.1°F). However, this can be improved if the PID control terms are more carefully adjusted to eliminate overshoot and ringing. Other improvements would include a higher water circulation rate, which could be achieved by using a pump with a higher flow rate to help reduce the temperature gradient across the water while heating.

Instructions For Use

To use the assembled Sous Vide device using the SousVide.py program, first fill a suitable container (capable of holding 18L of water and withstanding temperatures of 212°F) with 16L of water. Hang the Sous Vide device over the side of the container with a hook so that the heating element is completely submerged, being careful not to expose the electronic elements to water. Then, plug the power supply cable into a standard 120V wall outlet. The SousVide.py program should run automatically on start up. Next, prime the pump. Answer the safety questions displayed on the screen appropriately. If the answer to either question is *no*, follow the prompts on screen, making sure that there is water in the container and that the pump is primed. Once both questions are answered *yes*, the Sous Vide device will start. Select the desired set temperature and press enter. If at any point you wish to change the set temperature, scroll to the desired temperature and press enter. If enter is not pressed, the temperature will revert to the previous set temperature. Finally, wait for the water to reach the set temperature.

6 Credits

Hardware Design: John Lucero
Hardware Build: John Lucero
User Interface: Katie Ayala & John Lucero
PID and phase angle control software components: John Lucero
Operational Test: John Lucero & Katie Ayala
Report: Katie Ayala & John Lucero