1. Introduction

A thermal system named S2001 (in recognition of the term project group in the ME190 Spring 2001 class: Eddie McCloud, Chris Noeth, Sylvio Cardoso, Truc Pham, Peter Verbiest, and Demetrius Williams, who designed and fabricated the thermal system) consists of a temperature sensor, a power resistor (heater) on an aluminum plate, and a fan (among other components). This thermal unit can be interfaced with a micro-controller (such as a New Micros NMI board, Handy Board, or Basic Stamp) or DAQ (e.g., National Instruments DAQ) board. This system is designed to provide ME students with a vehicle to practice and implement, real-life PI/PID/Modified PI/On-Off temperature control.

The objective of this experimental study is to learn how to implement digital PID algorithms and observe closed loop system response with and without a process disturbance, and to derive the best PID gains for the temperature control of the small aluminum plate. The students are encouraged to study the components and system characteristic of the S2001 unit, and to learn how they can be interfaced to a microcontroller or a DAQ board for feedback control.

In the Section 2, the S2001 thermal system is presented. In Section 3, a very brief introduction of LabVIEW programming for implementing a process control system with a DAQ board is presented. Introduction of PID feedback control system is presented in Section 4. Section 5 contains the experiment assignment.

2. Thermal System and Schematics

The S2001 thermal system is shown in Figure 1 and Figure 2. The thermal system consists of the items listed in Table 1.

![Figure 1. S2001 Thermal system experiment board.](image-url)
The temperate of the plate is read using an LM35 temperature sensor. The LM35 is a Precision Centigrade Temperature Sensor manufactured by National Semiconductor. It is an integrated circuit that
outputs 10 mV for every degree Celsius. As seen in Figure 3, the output of the LM 35 is amplified by a non-inverting op-amp (LF 356) with a gain of 3.32. This gain was used so that the expected temperature extremes of 0 – 150 °C, would yield an output of 0-5 V. This way, the full span of the micro controller A/D (or DAQ board) can be used. With this setup, the controller should be able to resolve about 0.5 °C.

![Figure 3. Temperature sensor schematic.](image)

The fan is controlled by sending either a digital or analog signal (2.5 V is enough) to the base of the BJT (bipolar junction transistor). When the base conducts, a 5 V source gives rise to a current which flows thru the collector-emitter junction and drives the fan. The schematic for the fan is shown in Figure 4. The fan is used to introduce a disturbance for cooling the plate.

![Figure 4. Fan control circuit.](image)

The third main circuit is for driving the heater or power resistor (see Figure 5). The circuit incorporates two jumpers. The first jumper selects whether a simulated microcontroller control signal or actual micro controller control signal is to be used. The simulated control signal basically allows one to control the voltage going into the amplifier by tuning the trim pot. The voltage can be regulated from 0 to 5 volts. This was included so that the heater drive circuitry could be tested without the need of a microcontroller. So, if there is a suspicion that the heater drive circuit is not functioning, use the simulated signal via the trim pot to test the circuit.

The second jumper was included to accommodate both the NMI and Basic Stamp microcontrollers, which can output 2.5 volts and 5 volts maximum, respectively. In the up position, the first op-amp is bypassed (used for Basic Stamp). In the down position, the first op-amp amplifies the signal by a factor of two before the second amplifier (this is used with NMI). In either case, the maximum voltage across the resistor is about 5 V. The power transistor controls the high current that the heater requires.

LEDs and decoupling capacitors were also included over the power and ground lines to signify successful power hookup and filter out high frequency noise from the power supplies, respectively (see Fig.6). The
The overall system schematic is shown in Fig. 7. A controlled system with a set point of 50 °C will take approximately one minute to reach the steady state value.

The S2001 thermal system requires a triple-output supply that can deliver at least 1.5 Amps. It is a good idea to short the "ground" and "common" on the power supply. Each wire from the S2001 thermal system is clearly labeled, so that there is little confusion as to where the wires should be connected. Table 2 summarizes the connections that should be made to implement the system. Further detail on the circuit interface for the Labview DAQ terminal box is provided in the Appendix. Also, make sure that the jumpers on the heater drive circuit are set according to the microcontroller being used (see Table 3). For the Labview interface we follow the NMI Jumper Settings. If the "Heater" wire is not grounded or connected to the microcontroller output, the circuit may pick up noise or do what’s often referred to as "humming". This phenomenon is due to electromagnetic noise that is picked up from the surroundings.

Figure 5. Heater driver circuit.

Figure 6. LED power indicators and decoupling capacitors.
Figure 7. Overall system diagram.

Table 2. S2001 Power and Signal Connections

<table>
<thead>
<tr>
<th>Label</th>
<th>Wire Color</th>
<th>Directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>-12 V</td>
<td>Orange</td>
<td>goes to -12V power supply, shares common ground with other supplies</td>
</tr>
<tr>
<td>+12 V</td>
<td>Red</td>
<td>goes to +12V power supply, shares common ground with other supplies</td>
</tr>
<tr>
<td>+5 V</td>
<td>Blue</td>
<td>goes to +5V power supply, shares common ground with other supplies</td>
</tr>
<tr>
<td>Common</td>
<td>Black</td>
<td>goes to common/ground on power supply</td>
</tr>
<tr>
<td>Fan</td>
<td>Gray</td>
<td>goes to micro controller digital or D/A output</td>
</tr>
<tr>
<td>Heater</td>
<td>Purple</td>
<td>goes to micro controller D/A output (or digital output if using On/Off Control)</td>
</tr>
<tr>
<td>Temp Out</td>
<td>Blue</td>
<td>goes to micro controller A/D input (or A/D circuit for conditioning)</td>
</tr>
<tr>
<td>µP GND</td>
<td>Black</td>
<td>goes to a common ground on micro controller</td>
</tr>
</tbody>
</table>

Table 3. Heater Driver Jumper Settings

<table>
<thead>
<tr>
<th>Source</th>
<th>Jumper 1</th>
<th>Jumper 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMI µC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Stamp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trimpot Simulated Input</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


3. LabVIEW for Data Acquisition and Process Control

What is LabVIEW?

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where data determine execution.

In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains the code. If organized properly, the block diagram resembles a flowchart. The best way to learn LabVIEW programming is to request a LabVIEW Evaluation Package (free from National Instruments ni.com/labview).

LabVIEW is integrated fully for communication with hardware such as plug-in data acquisition devices. The lab computers have a DAQ board, with input and output capability for both analog and digital signals, that is controllable by the LabVIEW program. Using LabVIEW, you can create 32-bit compiled applications that give you the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions.

LabVIEW programs are called Virtual Instruments, or VI’s for short, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and displays that information or moves it to other computers.

A VI contains the following three components:

- Front panel – which serves as the user interface.
- Block diagram - which contains the graphical source code of the VI that defines its functionality.
- Icon and connector pane – which identifies the VI so that you can use the VI in another VI. A VI within another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages.

Tool palettes: LabVIEW Tool palettes give you the options you need to create and edit the front panel and block diagram. The Tools palette is available on the front panel and the block diagram. Use the tools to operate and modify front panel and block diagram objects. See the LabVIEW manual or books for more options.

The Front panel and Block diagram for temperature control of the aluminum plate are included in the Appendix. As you can see in the Front panel of our application, the user can set the desired plate temperature. The PID gains can be adjusted for the best response. The disturbance Fan ON/Off button is included to evaluate the disturbance rejection performance of the control system. The Stop button stops the running of the control system.

The block diagram shows that the voltage of the signal corresponding to the temperature sensor is read from the specified analog input channel of the device. This digitized input voltage is converted to the temperature in °F according to the calibration equation. The feedback temperature is then compared with the desired temperature, and an error signal is then formed. The PID subVI then calculates the control action required to reduce the error. The PID (or modified PID) gains are specified from the Front panel. The Delta T (DT) is needed for derivative control action (see the section on PID Controller). The control action from the PID subVI must be constrained within the range of 0 to 5 V. This constrained control signal is converted from digital form to an analog signal. The analog output channel of the given device will output the PID control signal. The FanON/Off digital action is used to send a signal to the digital output channel to produce an ON/OFF disturbance to the thermal process. The control system Stop Block can be used to terminate the control system action. Other options such as data recording with Excel is also included. See the Appendix for connection of the S2001 thermal unit to the DAQ terminal unit.
4. PID Feedback Control

Four themes thread through all of dynamic systems and control system theory: the search for a cause and effect relationship; the view that the current output of a system is a result of its past input; the use of a system's own output to regulate itself; and the definition of relevant interactions among a system's component parts (causality, dynamics, feedback and system decomposition). The objective of this experiment is to learn how to implement digital PID control algorithms, and fine-tune the PID gains for the desired closed loop performance with or without process disturbance (feedback control).

Fig. 8 shows the canonical configuration of an engineering feedback control system. The set point is one of the inputs to the closed loop system. It sets the value desired for the system's output, the controlled variable. The reference input (set-point) in our case stays constant for the desired temperature of the plate. Since the control object may be subject to disturbing effects and load changes from its environment, the disturbance in Fig. 8 is also considered as an input to the closed loop system. The output (the value of the controlled variable c(t)) is fed back via a feedback element (a measurement instrument). Its output, the feedback signal, is then checked against the set point and the difference (error) actuates the controller, which, in turn, applies a controlling input (the manipulated variable) to the control object.

Figure 8. A standard control system block diagram.

The step input response of a single loop feedback control system gives crucial information about the system's control performance. Depending on the purpose of the control system, the reference input can be a step or the disturbance can be a step. Four crucial parts of the response are marked as regions A, B, C, and D in Figure 9. When equilibrium at an initial state of zero is broken by a step application, the response starts after some delay or lag, as shown in region A. In part (b) of the figure, the response is caused by disturbance (V(t) in Figure 8). In region B the controlled variable tends to overshoot above or below the set point. The magnitude of the maximum overshoot is a measure of performance in region B.

Figure 9. Step input response (in the controlled variable) of a control system.

An indication of the speed of response may be also obtained for a step change in the reference input by observing the time at which the peak occurs. We can avoid overshoot by letting the rise level off, as
indicated by the dashed line in Figure 9a. and thus trade off reduced speed of response. The dominant period and decay of the loop oscillation are observed in region C. A control system may have no residual error or, as shown in region D, it may have an offset (a steady-state error). The settling time is the time at which the response becomes completely bounded by a tolerance band as shown.

**Linear Control Law**

A common form of the controller $G_c(s)$ that relates the manipulated variable $M(s)$ (or $m(t)$) to the error signal $E(s)$ (or $e(t)$) is called a **linear control law**. There are at least four basic linear control laws:

**Proportional Action (P-action)**

$$G_c(s) = K_P, \quad \text{or in time domain form}$$

$$m(t) = K_P e(t)$$

**Integral Action (I-action)**

$$G_c(s) = \frac{K_I}{s}, \quad \text{or}$$

$$m(t) = K_I \int e(t) \, dt$$

**Proportional plus Integral Action (PI-action)**

$$G_c(s) = K_P \left( 1 + \frac{1}{T_i s} \right) = K_P + \frac{K_I}{s}, \quad \text{or}$$

$$m(t) = K_P e(t) + K_I \int e(t) \, dt$$

**Proportional plus Integral plus Derivative Action (PID – action)**

$$G_c(s) = K_P \left( 1 + \frac{1}{T_i s} + \frac{1}{T_d s} \right) = K_P + \frac{K_I}{s} + \frac{K_D}{s}, \quad \text{or}$$

$$m(t) = K_P e(t) + K_I \int e(t) \, dt + K_D \frac{d}{dt} e(t)$$

Where $K_p$, $K_i$, and $K_d$ are gains and $T_i$ and $T_d$ are I-action and D-action time constants, respectively. These are normally adjustable controller parameters.

**Digital PID Control Algorithms**

Continuous Time PID Algorithm is given as

$$m(t) = K_p e(t) + K_i \int e(t) \, dt + K_d \frac{d}{dt} e(t)$$

Discrete Time PID Algorithms

Error: $e(k) = e(k DT)$, Control: $m(k) = m(k DT)$, Reference: $r(k) = r(k DT)$

Where $k$ is an integer, $k = 1, 2, 3, \ldots$ DT is the signal sampling time period.

Discrete time PID control action can be implemented by Velocity algorithm and Position algorithm.
Velocity Algorithm

\[ m(k) = m(k-1) + K_p (e(k)-e(k-1)) + K_i \left( \frac{e(k) + e(k-1)}{2} \right) DT + K_d \left( \frac{e(k)-2e(k-1)+e(k-2)}{DT} \right) \]

Position Algorithm

\[ m(k) = K_p e(k) + K_i I(k) + K_d \left( \frac{e(k)-e(k-1)}{DT} \right) \]

Here \( I(k) = I(k-1) + e(k) DT \). \( I(0) = 0 \).

Modified PID Controller

Position Algorithm

\[ m(k) = -K_p c(k) + K_i I(k) - K_d \left( \frac{c(k)-c(k-1)}{DT} \right) \]

Here \( I(k) = I(k-1) + e(k) DT \).

PID Parameter Tuning

The parameters in the PID controller can be selected by various methods. This topic is normally covered in a textbook on Controls. However, these methods require a dynamic model of the process, which is not always readily available. Ziegler-Nichols tuning is a method for picking the PID parameter based on fairly simple experiments on the process and thus bypasses the need to determine a complete dynamic model. Ziegler-Nichols and other PID Tuning methods are well covered in any standard control textbook.
5. Laboratory Assignment

PID Parameter Tuning in the Laboratory Experiment

The PI controller gains may be adjusted along with the following steps:

1. With no integral control action, find the proportional gain $K_p$ that gives rapid return to a steady state using a trial-and-error procedure.

2. Add integral action a little at a time, starting with a small $K_i$ value. Between each addition, create a small load change by changing the set point up and down a little. Also introduce a step disturbance when the process reaches steady state for each run. Leave the best proportional gain “as-is”.

3. When you see that integral is creating instability, reduce the integral gain a little. Leave the proportional gain “as-is”.

With PID Action

4. Leave $K_p$ and the best $K_i$ from the steps 1 - 4; add a small derivative gain, $K_d$. Observe how the performance of the control system changes by changing the set point and disturbance as stated above. If no noticeable performance improvement is observed, then leave the best PI controller gains for the closed loop system.

5. Derivative action added to proportional control action is particularly good for processes which are frequently started. The derivative action may greatly reduce the initial overshoot of the controlled variable, which produces stability sooner than proportional action can do alone. Derivative action is a stabilizing action when used in proper amounts. By increasing stability, derivative action permits use of a larger proportional gain. Obviously the best controller for ordinary process will be a PID controller.

Student Experimental Assignment

1. Study the S2001 thermal system components and the system. Study the signal levels.

2. Study the how the S2001 thermal unit can be interfaced to the terminal unit associated with the DAQ board.

3. Study the LabVIEW Front panel and block diagram for PID control. Study how to modify this program for On/Off , or modified PI control.

4. Fine-tune the PID parameters for the best control with the step set-point change and the disturbance on and off. Observe overshoot, steady state offset, disturbance rejection, and system noise.

5. Try to record the response to an Excel file for plotting the controlled system response.

6. Your comments and suggestions on the experiment are invited.

J.C. Wang