Temperature Control System

Cyber-Physical System Project

Leonardo Arrighi

January 27, 2022

1. Problem statement

The temperature control system is an example of a controller which is largely and daily used. This project presents the study of an *Arduino temperature micro-controller*, Temperature Control Lab (TCLab), which is described in (*Park et al., 2020*)[2]. The aim of this project is to develop a temperature controller: given a desirable temperature, the system must reach it. In order to reach the objective, a Proportional Integral Derivative (PID) controller has been developed. In addition, an Extended Kalman Filter has been studied to emulate the real sensor. We also stated some desirable properties of the system using Signal Temporal Logic. We have verified that these properties are satisfied by the proposed control strategy. Finally we propose the falsification of one of these properties by varying the parameters of the PID controller.

2. Plant Model

The plant consists of a printed circuit board (PCB) shield connected to an Arduino micro-controller, two transistors that acts as heaters and two temperature thermistor sensors (figure 1). A quick description of the plant and its functioning can be found in Appendix A, while more details can be found in *Park et al., 2020*[2]. The model is developed assuming the following ipothesis: the heaters have a uniformly distributed temperature; the sensors have a negligible thermal mass[1] and negligible surface areas; temperature changes are driven by heat conduction from the heaters. The model consists of a lumped parameter model which describes the dynamic input power to each actuator and the temperature sensed by each sensor; it is represented by equations 1-2-3-4:

$$m\,c_p\,\frac{dT_{H_1}(t)}{dt} = U\,A\,(T_{amb}-T_{H_1}(t))+\epsilon\,\sigma\,A\,(T_{amb}^4-T_{H_1}^4(t))+Q_{C12}(t)+Q_{R12}(t)+\alpha_1 Q_1(t), \tag{1}$$

$$m\,c_p\,\frac{dT_{H_2}(t)}{dt} = U\,A\,(T_{amb}-T_{H_2}(t))+\epsilon\,\sigma\,A\,(T_{amb}^4-T_{H_2}^4(t))-Q_{C12}(t)-Q_{R12}(t)+\alpha_2 Q_2(t), \tag{2}$$

[1] Thermal mass is the ability of a material to absorb, store and release heat

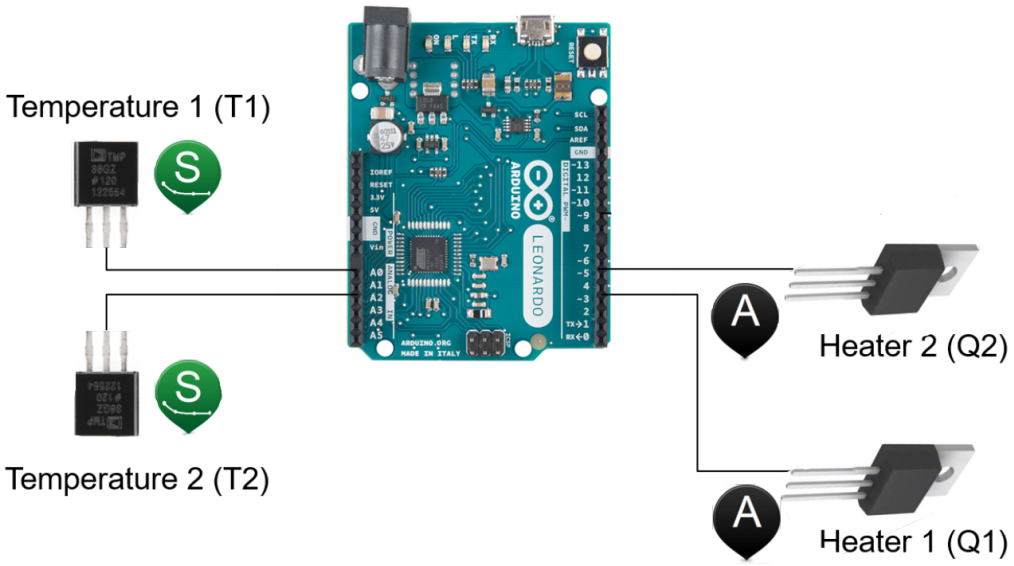

Figure 1: Temperature sensors (S) and heater transistors (A) with connections to an Arduino Leonardo.

where Q_{C12} and Q_{R12} are, respectively, the convective heat and the radiative heat transfer between the two heaters, explained by the following equations:

$$Q_{C12}(t) = U\, A_s\, (T_{H_2}(t) - T_{H_1}(t)),$$

$$Q_{R12}(t) = \epsilon\, \sigma\, A\, (T^4_{H_2}(t) - T^4_{H_1}(t)).$$

Furthermore, equations 3-4 represent the dynamic sensor temperature response expression:

$$\tau_c \frac{dT_{C_1}(t)}{dt} = T_{H_1}(t) - T_{C_1}(t), \tag{3}$$

$$\tau_c \frac{dT_{C_2}(t)}{dt} = T_{H_2}(t) - T_{C_2}(t). \tag{4}$$

The meaning of the variables is explained in table 4 in Appendix B.

2.1. Simulation result

As we can notice in equations 1-2-3-4, the model depends on the ambient temperature (T_{amb}) which is set to $23°C$. The plot in figure 2 is obtained by manually modulating the power percentages of the two heaters (Q_1 and Q_2), which are then provided as inputs to the model. The initial temperature of the heaters (T_0) is set to $23°C$. We have plotted the temperature of the heaters (T_{H_1} and T_{H_2}) and the temperature of the sensors (T_{C_1} and T_{C_2}), comparing them with the percentage of power provided to the heaters.

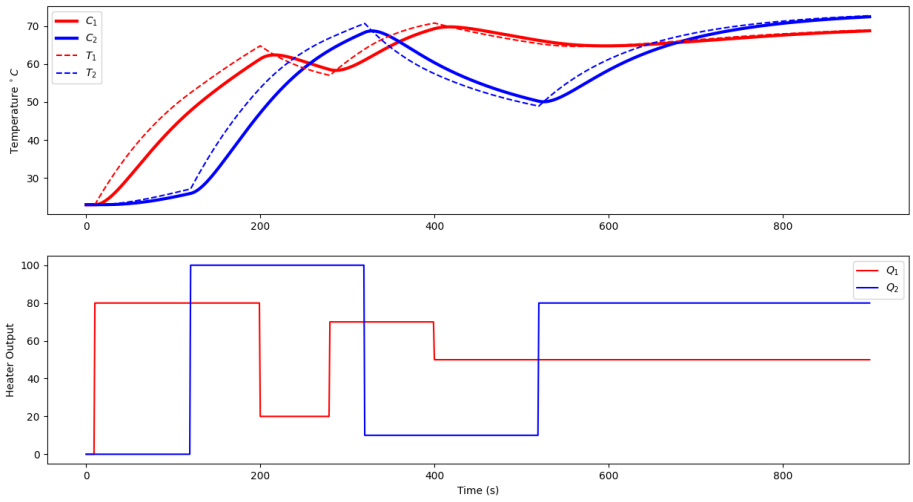

Figure 2: Plot of sensors and heaters temperature compared with the percentage of power provided to the heaters. Initial temperatures are set to $23°C$.

3. PID Control

As control strategy we have implemented a Proportional-Integral-Derivative (PID) controller for each heater, separately: the influence of each heater on the other is included in the model. In order to keep the same notation, we recall that the standard form of the PID control function is:

$$u(t) = K_c \left(e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau + \tau_d \frac{de(t)}{dt} \right), \tag{5}$$

where $e(t) = ref(t) - T_{C_i}$ is the error calculated as the difference between the reference temperature we aim to obtain and the temperature sensed by the i sensor.

3.1. First-Order Plus Dead-time Model

In addition to the physics-based model, a first-order plus dead-time (FOPDT) model is fit to step response data. The model is developed in order to obtain initial tuning parameters for the PID controller, which can be calculated using the parameters of the FOPDT model itself. For the simple purpose of the FOPDT model, heater 1 (Q_1) is varied along the course of the simulation, while heater 2 (Q_2) always remains off. The model is fit on the data that have been calculated by providing the heater inputs to the previously discussed model (equations 1-2-3-4). In the figures 3 and 4 we compare the plot obtained by carrying out the simulation of the model providing heater inputs, and the FOPDT model fit on the data calculated in the mentioned simulation. The FOPDT model is a differential equation:

$$\tau_p \frac{dT_{C_1}(t)}{dt} = -T_{C_1}(t) + K_p \, Q_1(t - \theta_p), \tag{6}$$

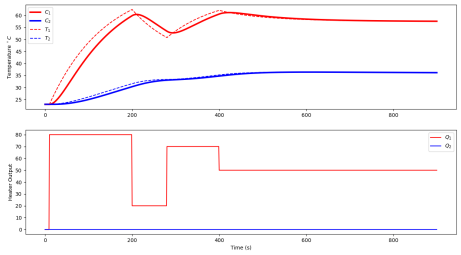

Figure 3: Simulation carried out providing heater inputs. Heater 2 is off.

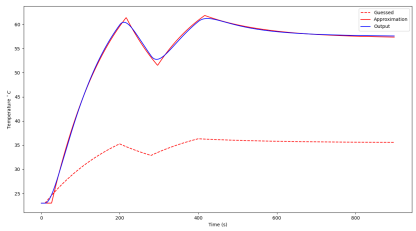

Figure 4: FOPDT model fit on the data calculated in the simulation.

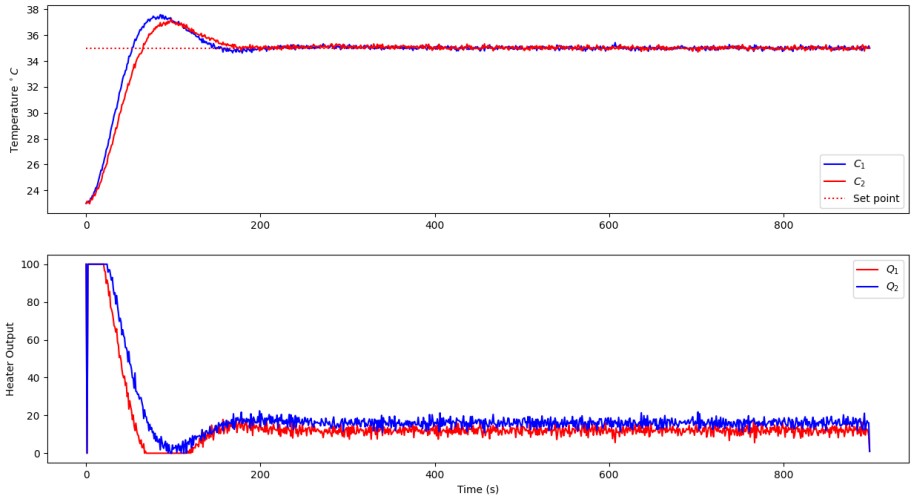

Figure 5: PID control, assuming as reference temperature $35°C$.

where τ_p, K_p and θ_p are the three parameters called, respectively, *gain*, *time constant* and *delay time*. As mentioned, the parameters are calculated and their values are: $\tau_p = 156.4326\,s$, $K_p = 0.6826\,°C/\%$ and $\theta_p = 16.7952\,s$.

3.2. PID parameters

Once we have calculated the parameters of the FOPDT model, we can use them to process the initial parameters to be entered in the PID controller, as shown in [3]. The final parameters tuned in PID control are: $K_c = 10.0438$, $\tau_i = 164.8303$, $\tau_d = 7.9698$

Performance	Value Heater 1	Value Heater 2
Overshoot	2.583	2.230
Rise Time	49.0	58.0
Steady State Error	0.002	0.024
Settling Time	144.0	161.0

Table 1: Measuring control performances. Reference temperature is $35°C$

3.3. PID result

In the figure 5 we can see one of the simulations tested. As we can see, we have chosen $35°C$ as reference temperature. The controller works correctly: after a first interval in which both heaters are on and operating at 100% of their power, the temperature is kept constantly around the reference temperature simply by maintaining the power of the two heaters to about 15%.

4. Extended Kalman Filter

In order to simulate real sensors we have add random normal noises to the states of the system and we have used an Extended Kalman Filter. The state observer has been implemented linearizing the dynamic of the system at each step of the simulation.

5. Performances of the controller

We have also implemented some functions to measure control performance. In particular we have introduced[2]: *rise time*, *overshoot*, *settling time* and *steady state error*. In table 1 we can observe the results obtained assuming $35°C$ as reference temperature.

6. Requirements

In this section we analyze some properties that our control system should satisfy and formalize them in Signal Temporal Logic (STL). The properties are requested for both heaters independently:

- Oscillations in the warm up phase: we require the system that the temperature of the heater warms up without high temperature jumps. We want to avoid large thermal changes that could affect the materials and the behaviour of the heater. To guarantee this, the oscillations

[2]Definitions were given during the course, so they are omitted here.

Requirement	Value heater 1	Value heater 2
ϕ_1	0.675	0.630
ϕ_2	0.100	0.168
ϕ_3	0.577	0.697
ϕ_4	62.417	62.770

Table 2: Robustness computed using as reference temperature: $35°C$

$(o(t) = |T_{C_i}(t) - T_{C_i}(t-1)|)$, during the warm up phase $(0 - 300\,s)$[3], must be less than $1°C$. In STL:

$$\phi_1 = \mathbf{G}_{[0,300]}(o(t) < 1.0).$$

- Oscillations in the maintenance phase: once a certain temperature is reached, the temperature oscillations $(o(t) = |T_{C_i}(t) - T_{C_i}(t-1)|)$ should be minimal, in order to ensure that the system maintains a certain stability. This interval comes after the previously discussed warm up phase $(300 - N\,s)$. In STL:

$$\phi_2 = \mathbf{FG}_{[300,N]}(o(t) < 0.5).$$

- Proximity: we require the system that the temperature reached is close to the reference temperature and that this property is maintained for the entire duration of the simulation. In STL:

$$\phi_3 = \mathbf{FG}_{[300,N]}(p(t) < 1.0),$$

where $p(t) = ref(t) - T_{C_i}(t)$.

- The mandatory property is that the temperature of the radiators never reaches $100°C$. Reaching this value would cause irreparable damage to the heaters and to the PCB. In STL:

$$\phi_4 = \mathbf{G}_{[1,N]}(T_{C_i}(t) < 100.0),$$

In order to verify these properties, we have simulated the model providing 9 values as reference temperature: $30°C$, $35°C$, $40°C$, $45°C$, $50°C$, $55°C$, $60°C$, $65°C$, $70°C$. The robustness is then computed for the PID control. Since the results are always positive we can conclude that the requirements are actually met. We can observe an example in table 2.

[3]The size of the warm up phase depends on the reference temperature that our heaters must reach. Since there is not a precise value, we have opted for an approximate interval which in some cases is greater than the phase

Reference value	Value heater 1	Value heater 2
$30°C$	68.251	68.594
$35°C$	62.200	62.683
$40°C$	56.023	56.492
$45°C$	49.790	50.129
$50°C$	43.301	43.432
$55°C$	36.839	36.819
$60°C$	30.065	29.894
$65°C$	23.192	23.340
$70°C$	16.297	17.774

Table 3: Robustness computed using as reference temperature: $35°C$

7. Falsification

Since property ϕ_4 is certainly the most important and mandatory, in this section we evaluate its falsification using the PID controller. We have ran the controller simulation for 10 iterations, each time varying the noises passed as input to the Extended Kalman Filter and the reference temperature. The final output is the minimum value of robustness calculated during the simulation. Since, once again, robustness always takes positive values, we can conclude that the property can not be falsified. In table 3 we can see an example.

8. Conclusion

In this work we have developed a temperature control system for two small heaters. The system can easily be adapted to larger complexes such as buildings. The peculiarity lies in controlling each of the two heaters separately, rather than both at the same time. We developed a PID controller, simulated the sensors with an EKF and finally we verified that the controller met some indispensable requirements. The work can be expanded by evaluating other requirements and proposing different strategies. The simulations were proposed using different values as the reference temperature. All results can be found in the github repository [1].

References

[1] Cps project. https://github.com/LeonardoArrighi/CPS_Project, 2022.

[2] Junho Park, R. Abraham Martin, Jeffrey D. Kelly, and John D. Hedengren. Benchmark temperature microcontroller for process dynamics and control. *Computers Chemical Engineering*, 135:106736, 2020.

[3] Anindo Roy and Kamran Iqbal. Pid controller tuning for the first-order-plus-dead-time process model via hermite-biehler theorem. *ISA Transactions*, 44(3):363–378, 2005.

Appendix

A. Plant description

The sensors T1 and T2 send temperature readings to the Arduino using a voltage signal (unit of measure: milliVolts mV) (values are in range $0 - 3300\ mV$). These signals are digitalized in the Arduino with a 10-bit Analog-to-Digital Converter (ADC), resulting in 1024 Discrete Levels (DL). Hence the signal is converted again to milliVolts: $mV = \frac{3300}{1024} DL$, and then to a temperature:

$$T = 0.1\ mV - 50.$$

Sensor accuracy is equal to $\pm 1°C$ at $25°C$. It increases, proportionally to the temperature, until to $\pm 2°C$ at $150°C$, or at $-40°C$. If the temperature sensor exceeds $100°C$, the Arduino firmware turns off the heaters to protect the equipment and the user.
The two heaters Q1 and Q2 are transistors (less powerful than traditional heaters). They are controlled through Pulse Width Modulation (PWM). The two transistors can be set to different power, between 0% and 100%, for a certain period of time. The maximum power output for the heater, with $5\ V$ power supply is more or less $1\ W$. A portion of the power supplied to the heaters is dissipated over the power cable, which has a thickness of $20\ AWG$.

B. Description of the constants

We can notice that the two heaters have the same mass (m), equal capacity (of the steel) (C_p), same heat transfer coefficient (U), same emissivity[4] (ϵ) and same Stefan Boltzmann constant (σ). Moreover, α_i represents the amount of power dissipated by the heater; as we can notice in the table 4, α_1 and α_2 are not identical: the two heaters do not perform equally. The time constant τ_c is a lumped parameter[5].

[4]The emissivity is the effectiveness in emitting energy as thermal radiation

[5]It comes from a discretized version of Fick's Law of heat transfer, with $\tau_c = m_s\, c_{ps}\, \Delta x / k_c\, A_{cond}$, where m_s is the mass of the sensor, c_{ps} is the heat capacity of the sensor, k_c is the thermal conductivity of the thermal epoxy, and Δx is the width of the thermal epoxy. It is estimated from the data and set in the model to the value of $23.3\ s$

Quantity	Value
Ambient temperature (T_{amb})	$296.15\ K$ ($23^\circ\ C$)
Heater output (Q_1)	0 to $1\ W$
Heater output (Q_2)	0 to $0.75\ W$
Heater factor (α_1)	$0.01\ W/(\%\ heater)$
Heater factor (α_2)	0.0075 W/($\%\ heater$)
Heat capacity (C_p)	$500\ \frac{J}{kg\,K}$
Suface Area Not Between Heaters (A)	$1.0 \times 10^{-3}\ m^2$
Suface Area Between Heaters (A_s)	$2.0 \times 10^{-4}\ m^2$
Mass (m)	$4 \times 10^{-3}\ kg$
Heat Transfer Coefficient (U)	$4.4\ \frac{W}{m^2\,K}$
Heat Transfer Coefficient Between Heaters (U_s)	$24.0\ \frac{W}{m^2\,K}$
Emissivity (ϵ)	0.9
Stefan Boltzmann Constant (σ)	$5.7 \times 10^{-8}\ \frac{W}{m^2\,K^4}$
Time constant (τ_c)	$23.3\ s$

Table 4: Values