Hot Take

Topic of Interest: PID controller for the coffee heating plate from Prelim 1

Abstract: Our goal is to create a new model for the coffee warmer from prelim 1, in which we have two settings: one that prioritizes rise time and the other that minimizes steady state error. We were inspired by how bad the system was in prelim 1 (36% too high). We will be exploring feedback control law, block diagrams, and ODEs, as well as steady state behaviour.

Students/Roles:

Student	**Task/Role**
Sean	**Parameter estimation, ODEs, Laplace**
Cameron	**Block diagrams, ODEs, Laplace**
Hovik	**Matlab scripts and plots. Expect to get a predicted plot for the temperature of the coffee with PID, PD or PI control, and a plot for the control effort.**
Sara	**Block diagrams and part of ODEs for open loop**

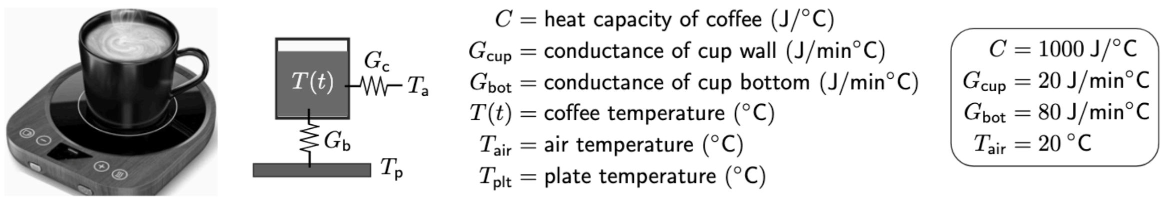
List of MAE 3260 concepts or skills used in this group work:

Block Diagram (dependent on ODE), Matlab Script and Plots (depends on ODEs and Block Diagrams), Information about temperature dissipation and heat transfer, in order to make the ODEs. Open-loop systems: Parameter estimation Steady state behavior, Step or frequency response. Active control: Feedback control law, Command following, Disturbance rejection.

Intro

The problem that we are addressing is improving the Coffee warmer from Prelim 1. We were inspired by the poor performance of the system and wanted to improve it. The closed loop proportional controller from Prelim 1 was 21.5º C below the reference temperature of 60ºC [1]. Basically cold coffee compared to the reference. Because the original design problem is that Professor Cambel drinks his coffee slowly, the primary system design goal is to ensure a very small steady state error, and secondary design goal is to decrease the rise time of the system.

We did this by designing a PI and PID controller based on the ODE from prelim 1.



C	= heat capacity of coffee (J/°C)
G_{cup}	= conductance of cup wall (J/min°C)
G_{bot}	= conductance of cup bottom (J/min°C)
$T(t)$	= coffee temperature (°C)
T_{air}	= air temperature (°C)
T_{plt}	= plate temperature (°C)

$C = 1000$ J/°C
$G_{cup} = 20$ J/min°C
$G_{bot} = 80$ J/min°C
$T_{air} = 20$ °C

Figure 1. System setup and parameters from prelim 1

The open loop model of this system can be written in 1st order form as follows:

$$C\frac{dT}{dt} = G_{bot}(T_{plt} - T) + G_{cup}(T_{air} - T)$$

A block diagram of the proportional closed loop model of this system is shown below where the hot plate temperature is the controller input. Using this model we created a PID control system.

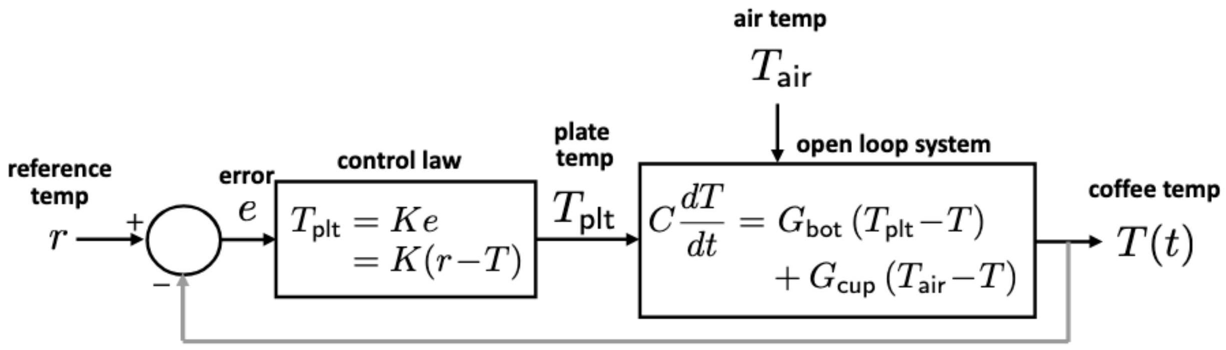


Figure 2. Block diagram of the closed loop system from Prelim 1

PID Control Law Derivation

We wanted to try both a PI and PID controller because the primary goal is to reduce steady state error so having the I term is really important, and then to improve rise time incorporating the D term. For PI control we set the K_D term to zero in the PID controller.

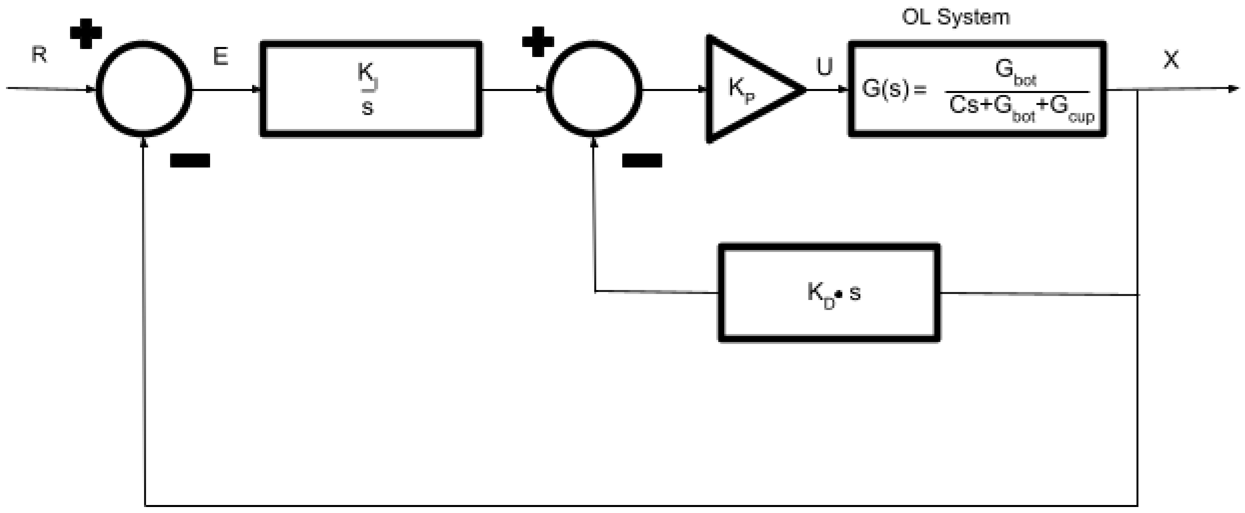


Figure 3. Block diagram of our PID coffee heater system

To implement the PID controller, we first define the control law in terms of the error:

$$e(t) = r(t) - Y(t)$$

where r(t) is the reference temperature and Y(t) is the coffee temperature. Taking the Laplace transform gives:

$$\mathcal{L}[e(t)] \Rightarrow E(s) = R(s) - Y(s)$$

The control input T_{plt} is then expressed as a function of the error:

$$T_{plt}(t) = K_p e(t) + K_i \int_0^t (e(\tau)d\tau) - K_d \frac{d\,e(t)}{dt}$$

Applying the Laplace transform to the control law yields:

$$\mathcal{L}[T_{plt}(t)] \Rightarrow T_{plt}(s) = (E(s))K_p(\frac{K_i}{s} - K_d s) = (R(s) - Y(s))K_p(\frac{K_i}{s} - K_d s)$$

This representation expresses the heater plate temperature in terms of the reference input and the coffee temperature, which can then be combined with the system transfer function to analyze the closed-loop behavior.

Open-Loop System Modeling and Laplace-Domain Representation

To find the transfer function we take the Laplace ($\mathcal{L}$) of the open loop system, then rearrange so that the output temperature Y(s) is in terms of the two disturbances and their transfer functions.

The open-loop system is described by the ODE:

$$C\frac{dT}{dt} = G_{bot}(T_{plt} - T) + G_{cup}(T_{air} - T) = G_{bot}T_{plt} + G_{cup}T_{air} - T(G_{cup} + G_{bot})$$

Taking the Laplace transform:

$$\mathcal{L}[\, C\frac{dT}{dt} = G_{bot}T_{plt} + G_{cup}T_{air} - T(G_{cup} + G_{bot})\,]$$

And then solving for Y(s):

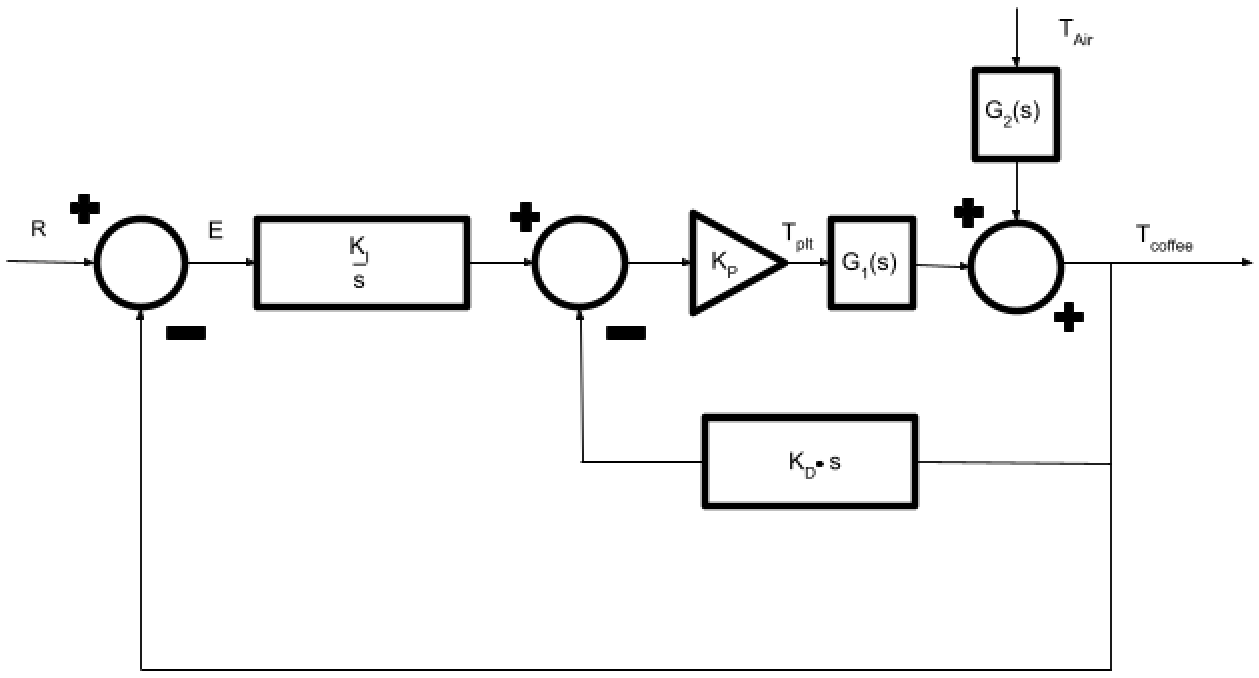
$$Cs\,Y(s) = G_{bot}T_{plt}(s) + G_{cup}T_{air}(s) - Y(s)(G_{cup} + G_{bot})$$

$$Cs\,Y(s) + Y(s)(G_{cup} + G_{bot}) = G_{bot}T_{plt}(s) + G_{cup}T_{air}(s)$$

$$Y(s)\,(Cs + (G_{cup} + G_{bot})) = G_{bot}T_{plt}(s) + G_{cup}T_{air}(s)$$

$$Y(s) = \frac{G_{bot}}{Cs+(G_{cup}+G_{bot})}T_{plt}(s) + \frac{G_{cup}}{Cs+(G_{cup}+G_{bot})}T_{air}(s)$$

$$Y(s) = G_1(s)T_{plt}(s) + G_2(s)T_{air}(s)$$



This results in a double transfer function system with disturbances T_{plt} and T_{air}. To get around the issue of solving a multi input single output system the T_{air} term can be approximated as a constant. Because T_{air} is constant we can approximate its Laplace transform as:

$$\mathcal{L}[T_{air}(t)] \Rightarrow T_{air}(s) = \frac{T_{air}}{s}$$

Substituting this into the system equation simplifies the analysis of the steady-state effect of the disturbance.

$$Y(s) = \frac{G_{bot}}{Cs+(G_{cup}+G_{bot})}T_{plt}(s) + \frac{G_{cup}T_{air}}{Cs^2+s(G_{cup}+G_{bot})}$$

State Offset Transformation and Simplified Transfer Function

To further simplify we can make a substitution, and because there are infinitely many state spaces we can make linear substitutions and the system is still fully represented. We define a linear offset relative to ambient temperature [2]:

$$x = T - T_{air} \qquad x_R = T_{plt} - T_{air}$$

The ODE in terms of the offset becomes:

$$C\frac{dx}{dt} = G_{bot}(x_R - x) + G_{cup}(x_R - x)$$

Taking the Laplace transform:

$$CsX(s) + X(s)(G_{bot} + G_{cup}) = G_{bot}X_R(s)$$

And then solving for the transfer function in terms of X(s) and X_R(s)

$$X(s) = \frac{G_{bot}}{Cs+G_{bot}+G_{cup}}X_R(s)$$

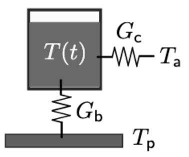
$$G(s) = \frac{G_{bot}}{Cs+G_{bot}+G_{cup}}$$

Finally, to recover the original temperatures, the outputs and reference inputs are offset by T_{air}.

With the simplified transfer function derived, we can now implement and analyze the closed-loop system using both PI and PID controllers. This model allows us to simulate the temperature response of the coffee warmer, evaluate steady-state error, and examine the effect of the derivative term on rise time and system stability. The following sections present the MATLAB implementation and results, demonstrating how the redesigned controller improves upon the original Prelim 1 system

Graphs and Results

MATLAB was used to implement our coffee warmer model and simulate performance and efficiency of our PI and PID controllers. The system parameters and controller gains were defined according to the ODE and transfer function derived earlier, and the closed-loop system was analyzed using step-response simulations. The simulations provide insight into the temperature response, steady-state error, and control effort of the system.



$$\begin{aligned} C &= \text{heat capacity of coffee (J/°C)} \\ G_{cup} &= \text{conductance of cup wall (J/min°C)} \\ G_{bot} &= \text{conductance of cup bottom (J/min°C)} \\ T(t) &= \text{coffee temperature (°C)} \\ T_{air} &= \text{air temperature (°C)} \\ T_{plt} &= \text{plate temperature (°C)} \end{aligned}$$

$$\boxed{\begin{aligned} C &= 1000\ \text{J/°C} \\ G_{cup} &= 20\ \text{J/min°C} \\ G_{bot} &= 80\ \text{J/min°C} \\ T_{air} &= 20\ \text{°C} \end{aligned}}$$

The controller's goal was to achieve a steady state error of zero, a 10-90 rise time of 10 minutes, to not exceed the max control effort of 100ºC. Using the same parameters from Prelim 1 except for the maximum control effort: C = 1000 J/ºC, G_{Cup} = 20 J/minºC, G_{Bot} = 20 J/minºC, T_{Air} = 20ºC, $T_{Plt\ Max}$ = 300ºC.

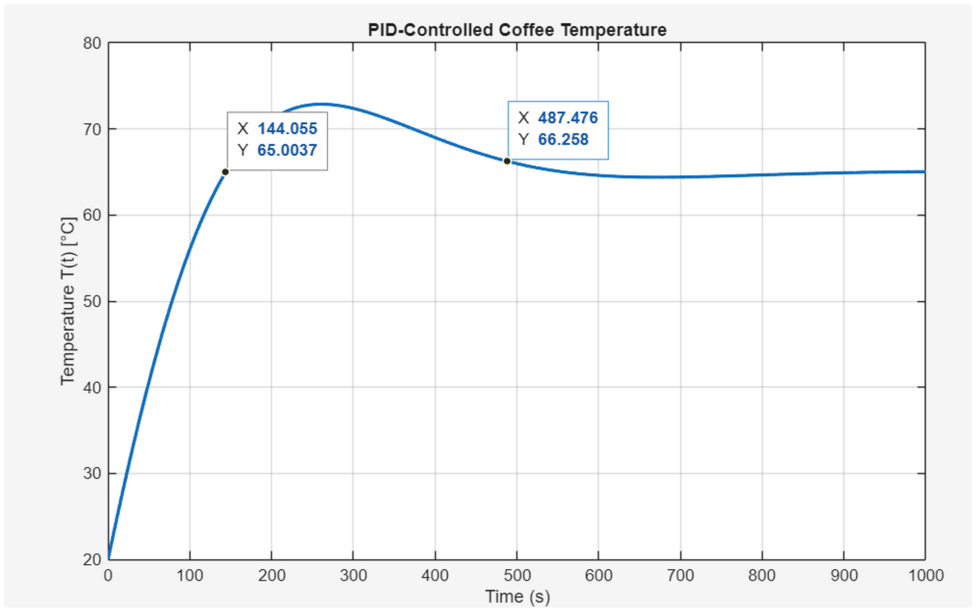


Figure 4: PID Temperature Graph

Figure 4 shows the coffee temperature as a function of time under PID control. The temperature rises from approximately room temperature to about 66°C, which was our reference. Compared to the original proportional controller from Prelim 1, this demonstrates a significant improvement in achieving the desired final temperature.

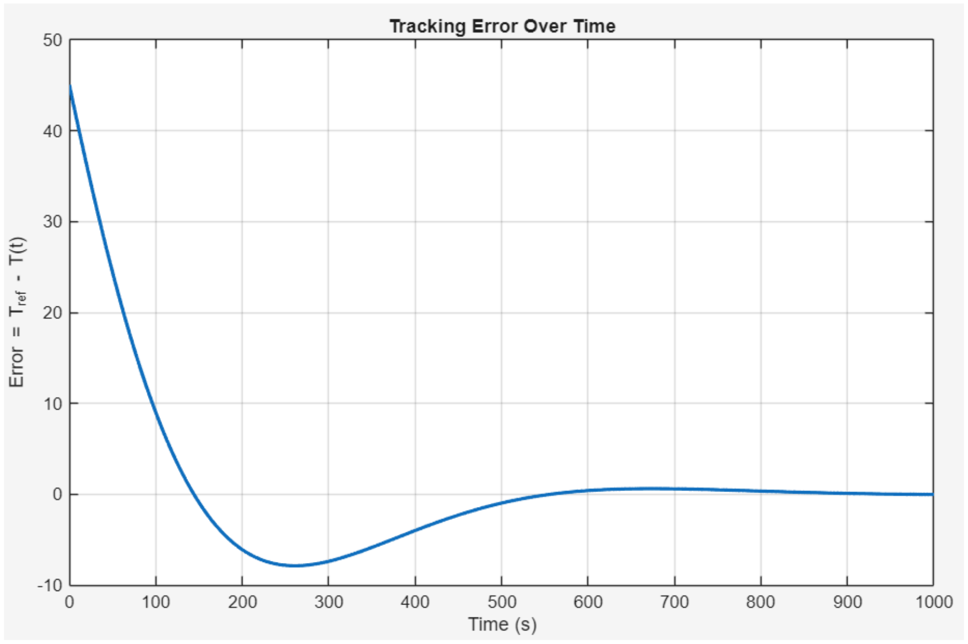


Figure 5: PID e_{ss} graph

Figure 5 highlights the steady-state error of the system. The error converges to nearly zero, meeting the primary design goal of minimizing steady-state error. The inclusion of the integral term in the controller is primarily responsible for this improvement, while the derivative term contributes to stabilizing the response and reducing overshoot.

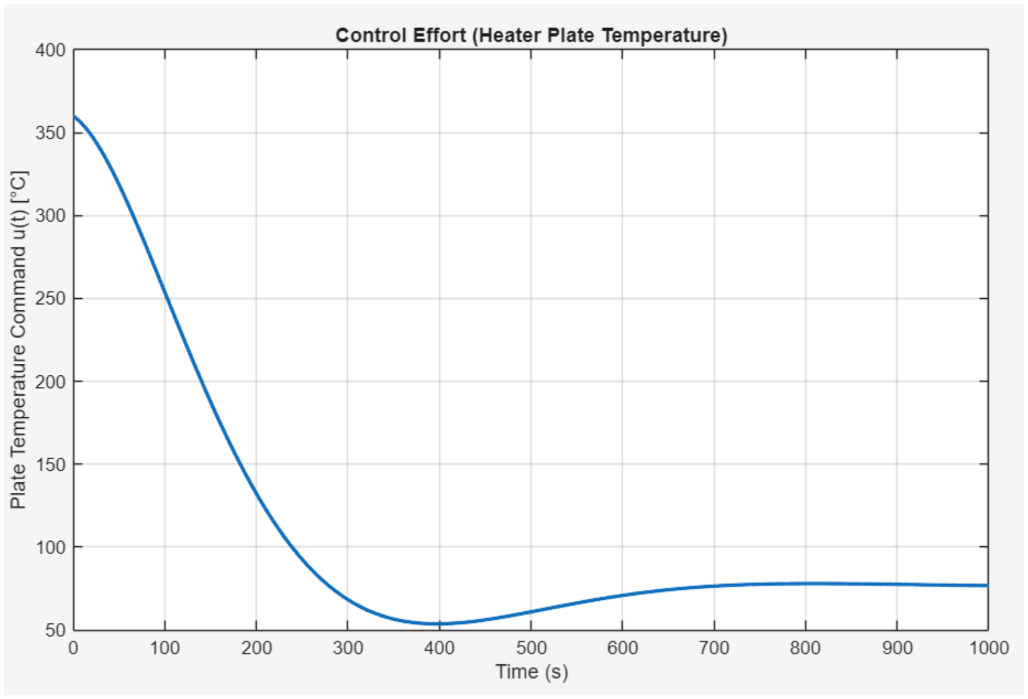


Figure 6: Control Effort Graph

Figure 6 shows the control input applied by the heater plate. The controller applies a large initial effort to rapidly increase the coffee temperature and then gradually reduces the input as the system approaches equilibrium. The graph indicates no saturation or excessive oscillations, confirming that the controller operates efficiently and as intended.

Assuming that the coffee is at 20 degrees celsius (air temperature) at the beginning, and our intended final temperature is 65 degrees celsius, our PI control system achieves a rise time of about 144 seconds, and a settling time of 487.5 seconds. Maximum control effort (the temperature of the hot plate) is observed at the very beginning, just above 350 degrees celsius. At the steady state, the control effort remains at around 77 degrees celsius. Our steady state error is 0. Overall, the PI control achieves our main tasks, which are lower rise time and settling time, as well as 0 steady state error.

References:

[1] M. Campbell, *MAE3260 Prelim 1,* 2025

[2] W. J. Palm, *System Dynamics,* 4th ed., New York, NY: McGraw Hill, 2021.

Matlab script

```
function [t, T, e, u] = coffee_pid(C, Gbot_min, Gcup_min, Kp, Ki, Kd, ...
                                  T_init, Tref, Tair, tf)
% PID-controlled coffee heater system
% Gbot_min, Gcup_min are given in J/min*C
   % Convert J/min*C --> J/s*C
   Gbot = Gbot_min / 60;
   Gcup = Gcup_min / 60;
   tspan = [0 tf];
   % State vector:
   % x(1) = T  (coffee temperature)
   % x(2) = I  (integral of error)
   % x(3) = e_prev (for derivative)
   x0 = [T_init; 0; 0];

     function dx = dynamics(t, x)
         T = x(1);
         I = x(2);
         e_prev = x(3);
         e = Tref - T;
         d_e = (e - e_prev);
         inner_sum = Ki*I + e + Kd * d_e;
         u = Kp * inner_sum;
         dT = ( Gbot*(u - T) + Gcup*(Tair - T) ) / C;
         dI = e;
         d_e_prev = d_e;
         dx = [dT; dI; d_e_prev];
     end
     [t, X] = ode45(@dynamics, tspan, x0);

       T = X(:,1);
       I = X(:,2);
       e_prev = X(:,3);
       e = Tref - T;
       % Compute derivative of error using stored state
       d_e = e - e_prev;
       % Compute control effort u(t) for plotting
       u = Kp * ( Ki*I + e + Kd * d_e );
end
clc; clear; close all;
C     = 1000;     % J/C
G_bot = 80;      % J/min*C
G_cup = 20;      % J/min*C
Kp = 8;
Ki = 0.009;
Kd = 0;
T_init = 20;   % initial coffee temperature
T_ref  = 65;   % reference
T_air  = 20;   % ambient
tf = 1000;      % 16.6 minutes
[t, T, e, u] = coffee_pid(C, G_bot, G_cup, Kp, Ki, Kd, ...
                      T_init, T_ref, T_air, tf);

% Coffee Temperature
figure;
plot(t, T, 'LineWidth', 2);
grid on;
xlabel('Time (s)');
ylabel('Temperature T(t) [°C]');
title('PID-Controlled Coffee Temperature');

% Error
figure;
plot(t, e, 'LineWidth', 2);
grid on;
xlabel('Time (s)');
ylabel('Error = T_{ref} - T(t)');
title('Tracking Error Over Time');
% Control Effort
figure;
plot(t, u, 'LineWidth', 2);
grid on;
xlabel('Time (s)');
ylabel('Plate Temperature Command u(t) [°C]');
title('Control Effort (Heater Plate Temperature)');
```