

Designing Control Systems

By now, you have encountered several control systems in BME354 lab such as the baby incubator. The baby incubator is an excellent example because it demonstrates all the key features of a simple control system: an output (controlling the heater), an input (the temperature reading), and some sort of feedback algorithm to relate output to input.

The baby monitor is a simple feedback-based control system in that the feedback is based only on the binary state of the output (above or below the set temperature), producing an output which oscillates about the set point. Control systems are present in many devices that you use daily, some of which are much more complex. Air conditioner systems use control systems to set the correct temperature of a room. An elevator uses a control system to smoothly bring the cabin to level with the destination floor. Quadcopters use advanced control systems to stabilize properly and perform aerial maneuvers and are also used in research applications to test novel control algorithms. In the future, control systems will be used in applications like driverless cars. These are only a few examples-there are even more nuanced applications of control system you probably wouldn't think of, but they are everywhere.

How does a control system work?

Control theory is a vast and deep subject which you could take many courses in, so here we will briefly cover how a control system works. There are many implementations of control systems-the one you implemented in the baby monitor lab is a primitive example. One of the standard mechanisms used is called **Proportional-Integral-Derivative**, or PID control. PID control, illustrated in the control diagram shown below, is an extremely popular class of control systems. In fact, PID is the most common controller used by industry.

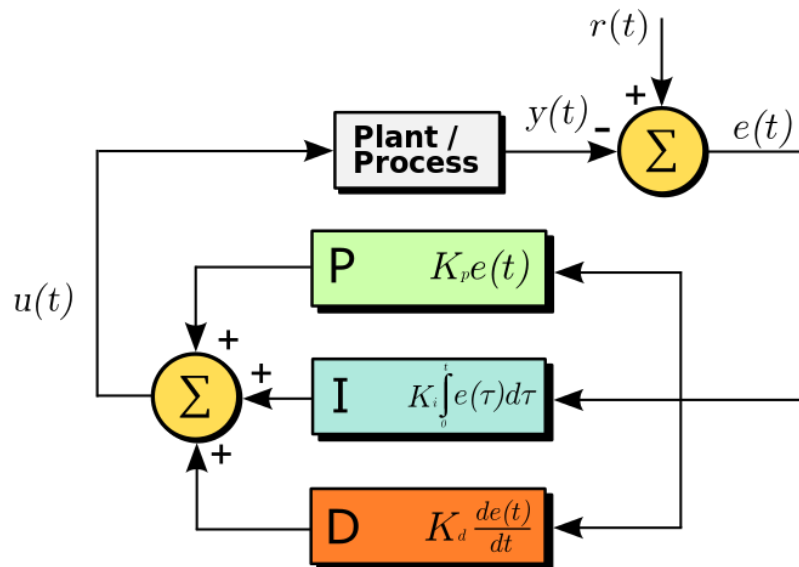


Figure 1: PID Control Diagram

Your feedback loop is based on three terms: the error between the desired value and the current actual value (proportional control), the total amount of error that has occurred up until now (integral control), and the rate of change of the error (derivative control). Each has a weighting scalar that you can change to

control how much each of them affects the output of the system. Using these parameters, you can control how quickly the system will adapt to reach the desired set point, how much the system overshoots, and whether the system oscillates. There is a lot of math that goes on behind the scenes that you will learn if you take a control systems course, but won't be covered here.

How to design a controller for a reflow oven?

In this project, aside from user interface, the primary task will be to design the control system. Although Arduino has made this process a lot easier for you through built-in libraries, there are still some design decisions you will have to make.

- What are the relevant temperature resolutions? How accurately can your system measure temperature? Is the resolution closer to 1°C or 0.1°C ? How does this affect your control system? Is it even necessary for our application to get 0.1°C resolution?
- What are the relevant time scales? How quickly do you need your input and output to change? How fast do you need to sample the temperature? This part of the design process is akin to deciding how fast you need to sample a biosignal. Just like a biosignal (which you have seen is not very well band-limited), the time scales / relevant frequencies in this system may not be clear cut.
 - What phenomena would affect this time scale? This is a good time to note that diffusive heat transport has $t_D \sim L^2$. If you have taken / are taking BME 307 then you recognize this as a time scale for chemical diffusion.
 - What would convective transport (i.e. using a fan in the reflow oven) do to the time scales that you have to worry about?
 - For the scale of the reflow oven, this means that temperature still fluctuates on a relatively slow time scale (s to 10s of seconds).
- Is a constant sampling rate the best way to handle your measurements? Could you use a higher precision technique during periods of change? Think about how oversampling / averaging could help improve accuracy.

Arduino PID Tutorials

Luckily, there are built-in libraries for Arduino that you can use. Watch the following videos to get started using them.

- PID Controller Intro: <http://www.youtube.com/watch?v=UR0hOmjaHp0>
- PID Control Examples: <http://www.youtube.com/watch?v=XfAt6hNV8XM>