Ray Mack, W5IFS

17060 Conway Springs, Austin, TX 78717: **w5ifs@arrl.net**

SDR Simplified: Demystifying PID Control Loops

The real story behind how a Proportional-Integral-Differential (PID) control loop works. Ray gives insights into how to build one and tune it for reasonable operation.

An Introduction to Closed Loop Control

We use closed loop control all the time in Amateur Radio. Regulated power supplies, phase locked loops, and oven control of crystal oscillator temperature are a few examples. All of these systems follow the block diagram shown in Figure 1. The goal is to control a physical property and maintain it at some value. Figure 2 shows the classic plot of the three types of step response of a controlled system can exhibit. The most stable (but not necessarily most desirable) response is over-damped response. This is the same response you get with a step in voltage on a simple RC circuit. It approaches the target voltage slowly until the difference between desired and actual is essentially zero. A faster response is the critically damped response. This is the response you get in an RLC circuit with extremely low Q. The resistance is so large that oscillations cannot get started. The fastest response possible is with an under-damped system. This is the same response you get in an RLC system with some moderate value of Q. The trade off in this system is significant cycles of oscillation above and below the desired value early in the response but with a rapid convergence to the desired value. The other trade-off is that the initial overshoot can be substantial.

Before the advent of computer circuit analysis, it was easiest to design circuits in the frequency domain using Laplace transforms to turn capacitors and inductors into poles and zeros in the frequency analysis. By moving poles and zeros around,

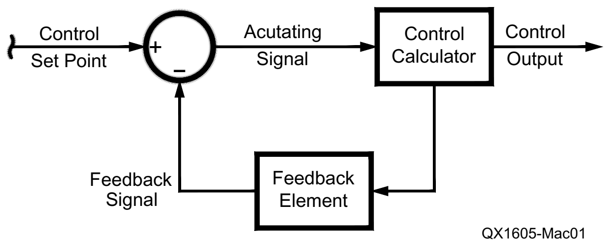


Figure 1 – Block diagram of a typical control feedback system.

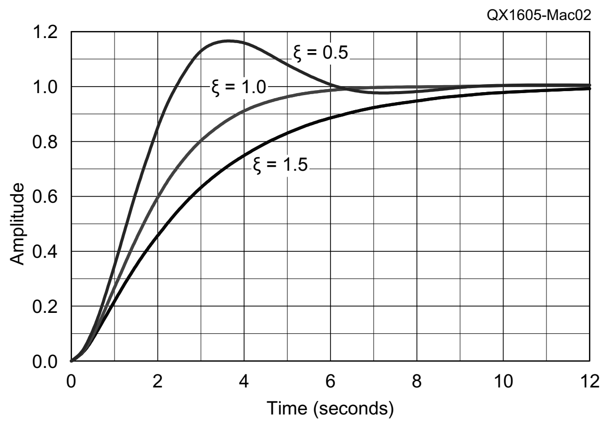


Figure 2 – The response curves of the classic control loop responses. ξ=0.5 shows the under damped response, ξ=1.0 shows a critically damped response, and ξ=1.5 shows an over damped responses.

you could modify the circuit response from over-damped, to critically-damped, or even to under-damped. The design process of creating a system with poles and zeroes is pretty daunting — but engineers still work this way when the situation fits.

The Full PID Loop

As in all electronics systems, you can evaluate operation in either the frequency domain or the time domain. You can do a lot of mathematical manipulations to convert an analog circuit with several capacitors — and occasionally an inductor or two — into a basic formula that has a single *integral* term, a single *proportional* term, and a single *derivative* term — a proportional-integral-derivative or PID loop. Its form looks like this:

$$\begin{aligned} Control\ Voltage = a\int error(t)dt \\ + b \times error(t) \\ + c\frac{d[error(t)]}{dt} \end{aligned} \tag{1}$$

Don't tune out if this equation means absolutely nothing to you, it is really quite simple as we will see.

Figure 3 shows an op-amp circuit that implements a hardware system with the performance of Eq. (1), and implements the control equation block from Figure 1. The top op-amp implements the integral term with the *a* parameter set by the ratio of C1/R1. The middle op-amp is a standard inverting amplifier where *b* is set by the ratio of R3/R2. The bottom op-amp implements the derivative term where *c* is set by the ratio of R4/C2. Op-amps work by converting the feedback current into a voltage. In these amplifiers, the input current is equal to the feedback current. The top op-amp performs an integrator function because the output will charge or discharge the capacitor C1 as long as the error voltage is not equal to zero. The voltage across a capacitor is the integral of the current through the capacitor. When the error voltage goes to zero the voltage on the capacitor will stay at some value. Likewise, the bottom op-amp implements a derivative function because the current through a capacitor (C2) is equal to the derivative of the voltage across it. If the input voltage does not change, then no current flows and the output of the op-amp is zero. The final amplifier is a summing amplifier with a gain of -1, so the whole system implements Eq. (1). In a real circuit, it may be possible to combine one or more of the circuits around an op-amp to make the circuit simpler, but it helps our illustration to see each term implemented individually.

The Proportional Part

It is possible to implement only a subset of a PID loop. I have used many loops that needed only the proportional and integral terms because the system was so slow that the differential term added nothing to performance. Of course, the simplest PID loop uses only the proportional term (*a* and *c* of Eq. (1) are zero). The problem for purely proportional control is that it requires that we know the exact transfer function to describe the difference between the set point and the feedback from the driving function. If the system drifts or external factors alter the transfer function, the system will have some small amount of error that is set by the proportional term gain. The error can be made small by increasing the proportional term gain, but the system is likely to overshoot the set point during transients. For this reason, control loops almost never operate in just proportional mode.

The Integral Part of PID

In many systems, we want the error to be as close to zero as possible when the system is in control. The integral portion of the PID equation provides that feature. Oddly, I have never seen this explanation in any control text book! In fact, when the system is in control, *only* the integral portion drives the output; the proportional term is exactly zero. The integral term ramps up to the required drive voltage slowly over time — where "slowly" is relative to your system operation — so that the error becomes zero. This allows the proportional term to operate more quickly to bring the system back into control if something knocks the system out of control. In general, we try to have the proportional term do its work about 10 times faster than the integral portion. The integral term supplies an adaptive feature to the PID loop that will compensate for external and internal drift or error.

The Differential Part of PID

Some systems need to respond very quickly to either a step change of the set point or an external push from stability. The differential term provides that quick but short-duration "kick" to push the system close to equilibrium. In general, we design the integral term to be slowest, the proportional

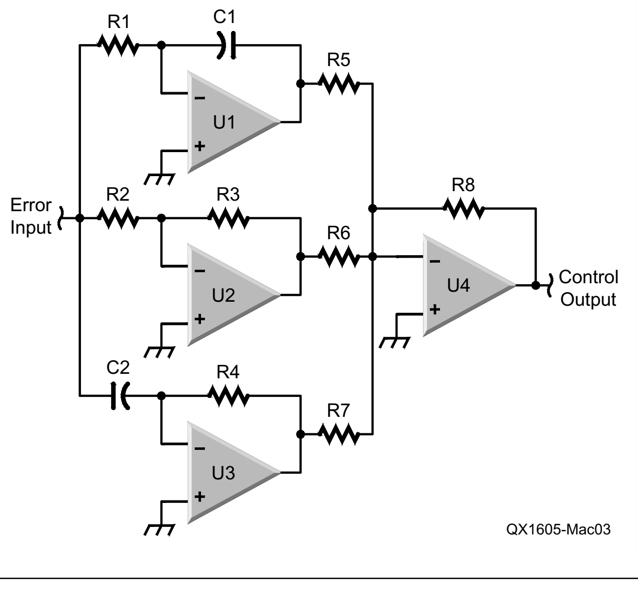


Figure 3 – A representative op-amp circuit that implements the PID control.

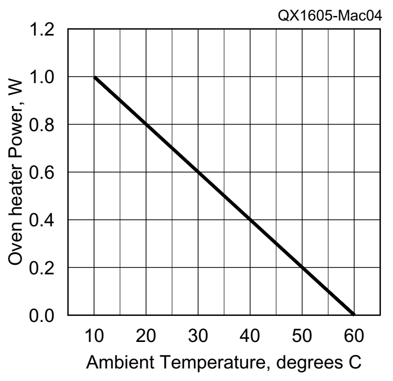


Figure 4 – A plot of power *vs.* ambient temperature for the OCXO control loop.

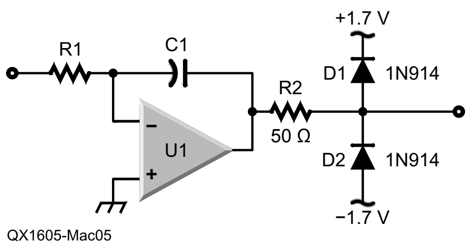


Figure 5 – A schematic showing a clamp of the integral function in an analog implementation.

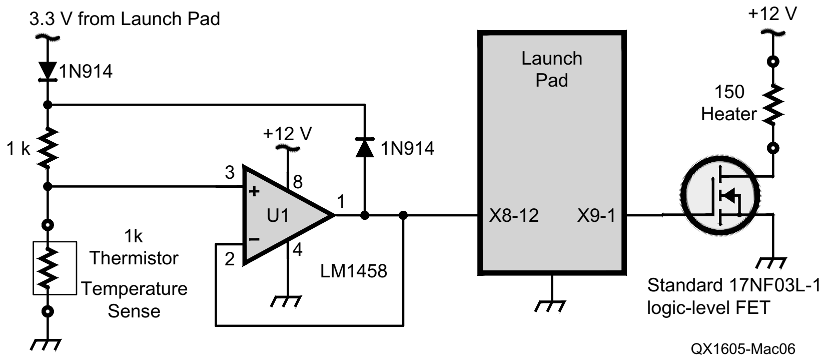


Figure 6 – The schematic of an example sampled PID control loop.

term to be ten times faster, and the differential term to be ten times faster than proportional term. The differential term is most useful in a system where a large, quick (but short), drive will move the system almost immediately closer to the control point. An example is a motor driven system like anti-lock brakes where putting a short 10× overdrive pulse of current through the motor will quickly move the actuator. In order for the differential term to be useful, the drive system must have significant overdrive capability compared to the amount of drive normally needed by the proportional and integral terms for slower control changes.

A (More or Less) Real Example

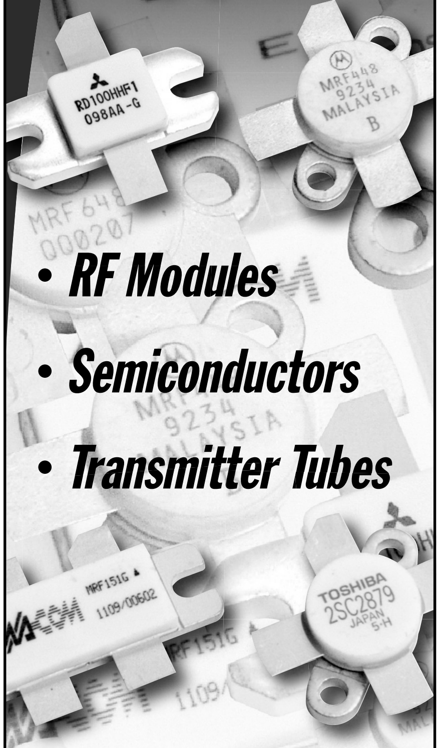
It helps to understand how things work if we can see how to implement a real system. Let's see how we would implement a heater for an oven controlled crystal oscillator (OCXO) that is designed to keep the crystal at 60° C ± 0.1° C. The first piece of information is the plot of the amount of heat in watts that is required to maintain the crystal at a given temperature relative to a constant ambient temperature (Figure 4).

Next we need to implement the feedback system to control the temperature. While we could implement the system with op-amps, this type of system is more easily implemented with a small microcontroller that has an on-board ADC for input and a PWM port for output. Using a computer allows us to create the control loop as a sampled system that directly implements the PID equation. Sampled systems do not actually implement an integrator or differentiator. Instead they approximate those functions as a sum of samples and difference between samples, but the results are essentially identical to integration and differentiation if we sample fast enough.

There is a feature of the integral function for both the op-amp and also the sampled systems that may not be obvious. The integral can grow to be either a very large positive or very large negative number if the error exists for any length of time. The op-amp integrator has a physical limitation: the output cannot exceed the positive rail or the negative rail. Once the output reaches the rail value, it is limited or saturated. We will look at tuning the integral term later, but for now we look at a way to limit the integral saturation to some value less than the rail. In an op-amp, we can use diodes to clamp the output to a value less than the rail as shown in Figure 5. In a sampled computer system, we can implement the "clamp" in software. Clamping the integral term is necessary especially in slow systems such as our heater example.

From MILLIWATTS
To KILOWATTS℠
More Watts per Dollar℠

In Stock Now!
Semiconductors for Manufacturing and Servicing Communications Equipment



- ***RF Modules***
- ***Semiconductors***
- ***Transmitter Tubes***

Se Habla Español • We Export

Phone:	**760-744-0700**
Toll-Free:	**800-737-2787**
(Orders only)	**800-RF PARTS**
Website:	**www.rfparts.com**
Fax:	**760-744-1943**
	888-744-1943
Email:	**rfp@rfparts.com**

VISA MasterCard AMERICAN EXPRESS Cards

RF PARTS™
COMPANY
From Milliwatts to Kilowatts℠

Listing 1

```
  /****************************
  The control loop function that implements
  PID control of an OCXO
  ****************************/
  #define PROPORTIONAL_PARAMETER  4
  #define INTEGRAL_PARAMETER 2
  #define DIFFERENTIAL_PARAMETER 0
  // differential value turned off by setting to zero.
  // Set to positive value to implement differential action
  #define INTEGRAL_RAIL  100
  #define NEG_INTEGRAL_RAIL -100
  // maximum or minimum value that the integral term can
  // attain. This limits overshoot for very large excursions.
void PID_control_loop(int target_ADC_reading)
{
int error, ADC_value;
int integral_accumulator, integral_term;
int differential_term, last_error;
int proportional_term;
int timer_value;
int PWM_value;

  integral_accumulator = 0;
  while (1) // an infinite control loop
  {
    // sit here and burn cycles until the next sample time
    while (TimerLoadGet(TIMER0_BASE, TIMER_A) != 0)
    {} // empty loop
    // restart the timer
    TimerLoadSet(TIMER0_BASE, TIMER_A, 10000);
    ADC_value = read_ADC(); // a helper function that reads the ADC0 pin.
    error = target_ADC_reading - ADC_value;
    proportional_term = PROPORTIONAL_PARAMETER * error;
    integral_accumulator += INTEGRAL_PARAMETER * error;
    if (integral_accumulator > INTEGRAL_RAIL)
        integral_accumulator = INTEGRAL_RAIL;
    else if (integral_accumulator <  NEG_INTEGRAL_RAIL)
        integral_accumulator = NEG_INTEGRAL_RAIL;
    integral_term = integral_accumulator;
    differential_term = DIFFERENTIAL_PARAMETER * (last_error - error);
    last_error = error;
    PWM_value = proportional_term + integral_term + differential_term;
    if (PWM_value > 400)
        PWM_value = 400;
    else if (PWM_value < 0)
       PWM_value = 0;
    PWMPulseWidthSet(PWM_BASE, PWM_OUT_0, PWM_value);
  }
                                          }
```

Implementing and Tuning

I implemented the example for this explanation using a Texas Instruments Tiva 129 Launchpad because it is an inexpensive and capable system that has the peripherals needed for the control loop: an ADC, a PWM, and a timer. Additionally, you can set the *Code Composer Studio* to a mode that displays debug print in the console, so you do not need a serial connection to the target while debugging. *Code Composer* also comes with a set of functions that mirror the internal ROM functions for controlling the peripherals. These functions really make writing software much easier!

Figure 6 shows a schematic of the example heater control that could be used for an OCXO. My experimental setup uses the 150 Ω resistor to provide up to 1 W of heat with a 12.6 V power supply. The resistor is placed on one side of a cube of aluminum 0.5 inch per side and the thermistor is placed opposite the heating resistor. The two resistors and the aluminum are enclosed in plank foam shipping material to insulate the assembly from ambient. The next step is to determine the set point. I chose an Ametherm 1 k Ω NTC thermistor with response curve B. From the table of relative resistance *vs.* temperature, we get 318 Ω at 60 °C. This means the feedback voltage will be 1.21 V when the system is in control. Note that we need to limit the input voltage to a value less than 3.3 V using the bipolar transistor. The target ADC value is (1.21/3.30) × 4096 or 1638. The example PWM setup from the Tiva 129 data sheet uses a 10 MHz clock to set the PWM frequency to 25 kHz and gives a range of 0 – 400 for the PWM value. The *Launchpad* uses a 25 MHz crystal, so the actual frequency is 62.5 kHz but still with 0 – 400 PWM range.

Listing 1 shows the sequence of software commands that implements the PID loop. It is quite simple. Step one, wait for 10 ms timer to elapse. This sets the loop to operate with a constant 10 ms sample period. Step two reads the ADC and compares against the target to determine the error value. The next step calculates each of the three parts of the PID equation and generates the control value. The last step applies the control value to the PWM hardware. The full set of software including the hardware configuration is available on the *QEX*files web page, **www.arrl.org/qexfiles**, as well as from **dsp-radio-resources.info**.

Tuning the loop is a lot easier if you have a laptop to watch the output while you put the assembly in your refrigerator and freezer. Start with the assembly sitting on your desk, which is pretty close to 22 °C. Place the assembly in the foam insulation and close it. Start the system running and watch the error value in the console of *Code Composer Studio*. The error will start out positive and approach zero. You will know how well the system is working by whether the error change slows a lot when close to zero or goes right past zero to become negative. This process can take quite a while depending on the thermal mass of your system. You can adjust the integral parameter larger or smaller to set one of the classic responses. The proportional parameter also affects the response. Setting the proportional parameter too large can cause the system to oscillate because it overcompensates for small errors. In most systems a critically damped response provides the best compromise. That means that the integral term is modest as is the proportional term. You could use an upside down "microduster" or other cold source to give a cold spike to the system to investigate how adding a differential term can rapidly bring the system back into control after a spike of hot or cold.

Errata

Correcting the Formula for Return Loss in the ANSI Standard

Edward Wetherhold, W3NQN, IEEE Life Member, reports that a correction was approved to the equation for *Return Loss* in the ANSI/EIA-364-108-2000 Standard. The correction places a minus sign before the original and incorrect equation to make the resultant and corrected *Return Loss* (dB) to be positive. The corrected equation is,

$$Return\ Loss = -20\log_{10}|\Gamma| = -20\log_{10}|s11|$$

where $\Gamma = s11$ is the voltage reflection coefficient.

From MILLIWATTS
To KILOWATTS
More Watts per Dollar

Transmitting & Audio Tubes

Eimac® RF Taylor® TUBES

COMMUNICATIONS
BROADCAST
INDUSTRY
AMATEUR

Immediate Shipment from Stock

3CPX800A7	4CX1000A	810
3CPX1500A7	4CX1500B	811A
3CX400A7	4CX3500A	812A
3CX800A7	4CX5000A	833A
3CX1200A7	4CX7500A	833C
3CX1200D7	4CX10000A	845
3CX1200Z7	4CX15000A	6146B
3CX1500A7	4CX20000B	3-500ZG
3CX3000A7	4CX20000C	3-1000Z
3CX6000A7	4CX20000D	4-400A
3CX10000A7	4X150A	4-1000A
3CX15000A7	572B	4PR400A
3CX20000A7	805	4PR1000A
4CX250B	807	*...and more!*

Se Habla Español • We Export

Phone:	760-744-0700
Toll-Free: (Orders only)	800-737-2787 RF PARTS
Website:	www.rfparts.com
Fax:	760-744-1943 888-744-1943
Email:	rfp@rfparts.com

VISA MasterCard AMERICAN EXPRESS Cards

RF PARTS™ COMPANY