PID loop

The PID algorithm is a negative-feedback control algorithm that tries to stabilize a system by making use of a proportional counteracting drive. In the case of PID, the simple proportional method of Watt's steam engine governor (see p.154) is complicated by the inclusion of derivative and integrative factors. Simple proportional methods lack accuracy due to their own counteracting nature. Like in ON-OFF methods (see Figure 78a), if the applied power is high enough to yield good transients, the proportional method will overshoot (and then overreact in counteraction), yielding a usually stable, but not accurate, oscillating pattern (see Figure 78b). To avoid oscillation, the applied power (gain) must be exceedingly low, producing a system that stabilizes way below the desired temperature.

The inclusion of a derivative term into the control system (see Equation 5b), produces a damping effect, since it gives the system a rough knowledge of its ongoing tendency that allows it to smooth the feedback response (see Figure 78c). Proportional-derivative (PD) systems tend to stabilize better and quicker than proportional ones if noise level is not too high, but they still produce a steady-state error when oscillation disappears, stabilizing below the desired temperature. In addition, too much damping will invariably result in a hard slowing of transient times. To overcome both P and PD state-state error problems, an integrative term can be introduced (see Equation 5c). In essence, the integrative term provides the system with a long-term *memory*, and its net effect is to change the actuator power until the time-averaged value of the temperature error is zero (see Figure 78d). Unfortunately, integrative terms can also introduce unwanted overshooting when fast transient times are required, an issue that will be dealt with later on (see p.165).

$$(a) \qquad P_{OUT} = P \cdot (T_S - T_R) \qquad (b) \qquad PD_{OUT} = W \cdot \left[P \cdot (T_S - T_R) + D \cdot \frac{d}{dt}(T_S - T_R) \right]$$

$$(c) \qquad PID_{OUT} = W \cdot \left[P \cdot (T_S - T_R) + D \cdot \frac{d}{dt}(T_S - T_R) + I \cdot \int (T_S - T_R) dt \right]$$

Equation 5 - Formulae for the parallel proportional, proportional-derivative and proportional-integrative-derivative algorithms. *W* stands for the global gain of the system, whilst *P*, *I* and *D* are the adjustable PID gain parameters. T_S and T_R correspond, respectively, to the set temperature and the real temperature.

The program PID loop is a simple routine that calculates the PID parameters at each round, yielding a drive voltage to be applied to the actuator. The PID routine is called iteratively from a higher-level loop, embedded in the PCR-cycle or other high-level steps, and its iterative parameters (current integral cue, actual error, etc.) are passed through (or reset, at convenience) by the higher-level loop.

Data acquisition board interface

At the core of the control algorithm lies the lowest-level routine, which controls the DAC interface and conditions read data for PID operation. Basically, the DAC interface unit scans the DAC buffer at regular, specified time intervals and calculates a mean value for read parameters using a prefixed number of reads. Concerning output, the DAC interface unit generates the drive output voltage specified by the PID through one of the DAC analog outputs and produces the switching signal to shift the external circuitry multiplexer via a digital line. After calculating the mean value of signal reads, the unit reverses all external amplification steps, introduces user-defined calibration offsets and calls interpolated look-at-table functions to convert non-linear thermocouple and Pt100 voltage readings into temperatures. In the case of the multiplexed driver power consumption variables (voltage and intensity), the unit uses a buffer and its knowledge of the set reading (voltage or intensity) to infer power consumption every two cycles. Finally, the unit is called by a somewhat higher-level routine, which calculates the error between actual temperature read and set temperature for PID use.

Control optimization

PCR control

As mentioned before, the PID method is a nice solution to negative-feedback typical steady-state error problems, but the introduction of an integrative term poses some problems by itself. In the particular case of temperature control for PCR amplification on chips, there are some facts that need to be taken into account. For instance, to gain advantage of the intrinsic properties of downscaling PCR into chips, very fast transition times are required (see p.97). If high temperature gradients are required (like 55 °C to 95 °C in a typical PCR protocol), the integrative factor will not decrease fast enough and will generate large and long overshooting effects (see Figure 79).

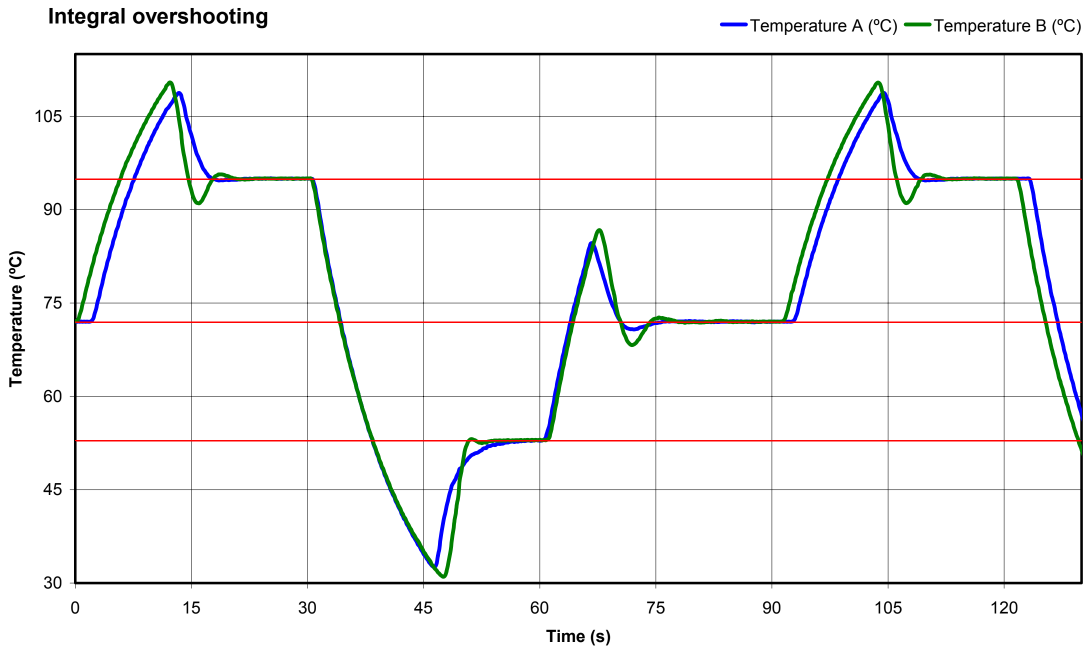

Figure 79 - Overshooting due to increased integral terms (A and B)

This might not be a problem in situations where temperature is not critical parameter, but this is clearly not the case of PCR. For example, a 4 °C overshoot when reaching 95 °C will cause deterioration of both the template DNA and the polymerase enzyme, and might even lead to boiling, breaking havoc onto PCR. Similar overshoots in extension and annealing times will drive down specificity and drop the chances of reducing stationary times. Hence, single PID control is not suited to deliver fast transient times and quickly stabilizing stationaries like the ones depicted in Figure 92, p.176.

Mixed PD-PID control

To overcome the problem of PID overshooting in fast transients, the obvious solution is to discard integrative terms, but this will leave the system with classical steady-state errors that are also not suited for PCR processes, in which lower than ±0.5 °C errors are recommended. Thus, the solution to PID control for low thermal mass PCR lies in the composite use of separate PD and PID controls. In the present control software, two different adjustable hysteresis levels are introduced. The first one determines switching between PD and PID control, whilst the second one decides whether the system has achieved a stationary phase and starts the downtime counter for that particular PCR step. Hence, fast transients are achieved by a boosted PD algorithm, which is turned off after reaching the first hysteresis (see Figure 80). Thereafter, a full PID control algorithm takes over, with an initial integral cue manually adjusted by the user, and stabilizes the system with ±0.1 °C accuracy. In this way, fast transient times (above 5 °C/s) can be achieved without overshooting nor steady-state error. After some tinkering with the system, optimal cycling behaviors can be obtained for a particular set of cycling temperatures, as depicted in Figure 92, p.176.

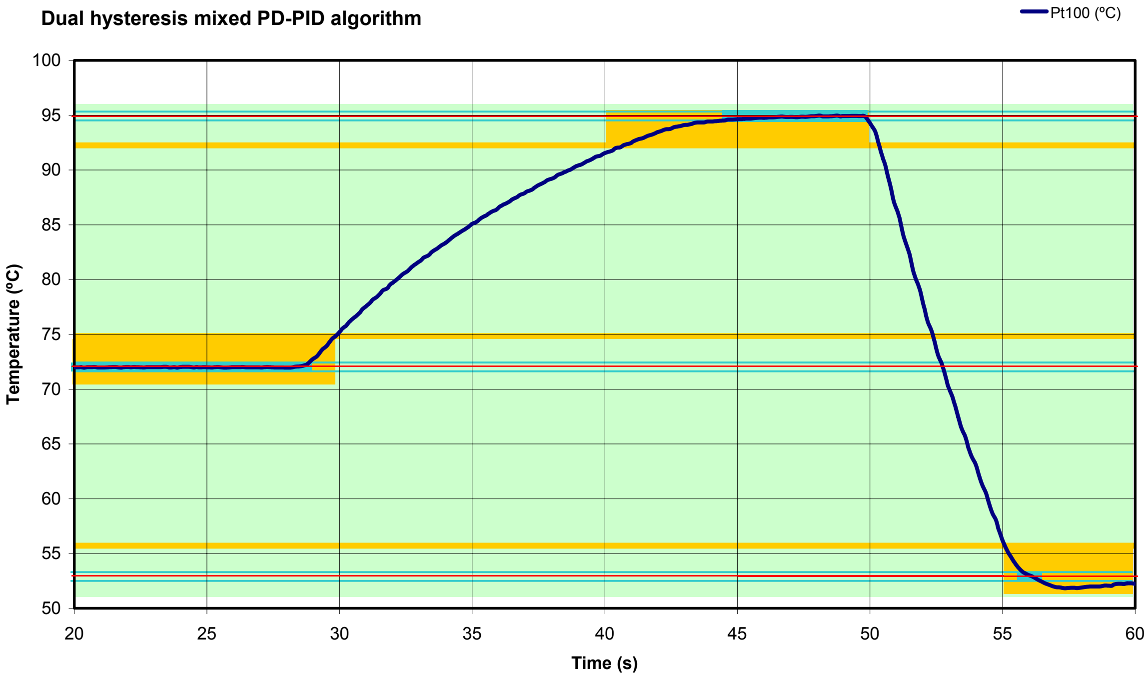

Figure 80 - PD-PID dual mode of operation for fast transient times and stable stationaries. Green zones denote simple PD operation, while the orange line illustrates the method switching hysteresis. Thus, orange zones correspond to PID operation. Blue lines and zones represent countdown hysteresis and zones. If temperature migrates off the countdown hysteresis before achieving the desired stationary time (like in the 53 °C zone), the counter resets itself. Red lines mark set temperatures.

Program interface

The program interface consists in a main window (see Figure 81), where input variables (such as PID parameters, cycling times, number and temperatures) can be modified at will by the user. The main window also displays thermocouple (set to ambient reading in Figure 81) and Pt100 temperature readings through a graphic interface, and power consumption and drive variables in numerical fields. The user can also specify a path and file name to record PCR cycling performance in Ms-Excel compatible data-sheet format. The main operating buttons are *Start* (to start and stop the program operation) and *Preheat*, to start preheating in hot-start PCR. Cycle and step information are also constantly displayed at the right of the graphical output, as well as internal DAC parameters (scan rate and number of reads to produce the mean read value) to allow checking for hardware errors.

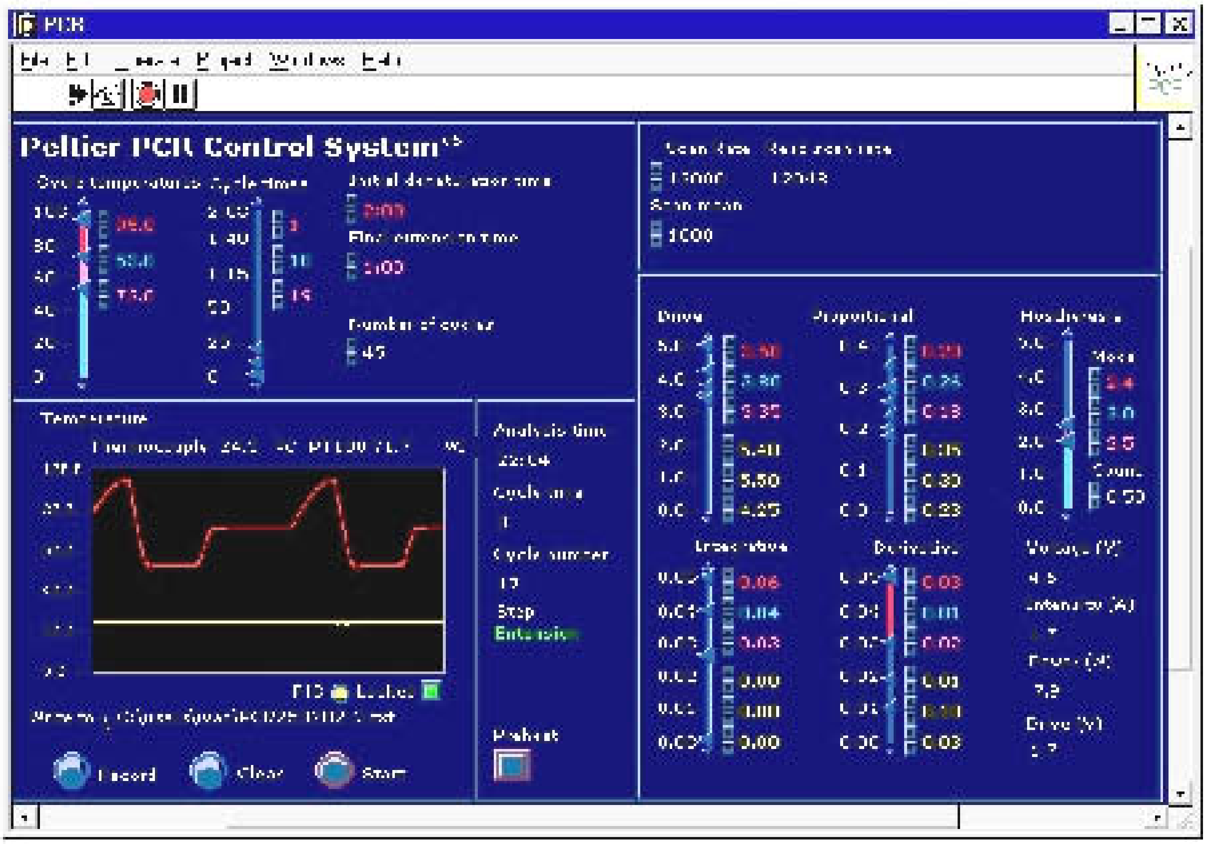

Figure 81 - Labview control-software interface screen.

4.4.6. SYSTEM SETUP

Assembly

After development of the control circuitry and the control software, a prototype fast external thermocycler was assembled (Figure 82) to test its efficiency and assess its main problems.

Setup

The assembled system is a particular adaptation of that proposed by *Shoffner's* team [Cheng1996b]. As already mentioned, a Labview program (*National Instruments*) running on a Pentium-I processor (*Intel*) takes full control of the temperature control and PCR cycling issues through a 12bit PCI-1200 data acquisition board (*National Instruments*). The DAC controls the behavior of the system by receiving input from a Pt100 and a type K thermocouple through conditioning circuitry powered by a standard dual ±15 V-1 A power supply (*Promax*), and stimulating a driver circuitry powered by the ±12 V-10 A FE-17 power supply (*Cebek*). The system actuator, a Peltier cell, is mounted with 340 Heat-sink compound (Dow Corning) onto a undrilled tee-slot heat dissipater (*IMI Marston*) and connected to the power driver, while the Pt100 is clamped onto the Peltier cell surface and the thermocouple is left unbound for room temperature sensing. All signal lines are ground-shielded with a metallic mesh, while conditioning circuitry is encased into grounded metallic boxes to avoid RF interference. Ground paths are distributed radially to all the circuit boards from the ±12 V-10 A power supply to avoid ground loops.

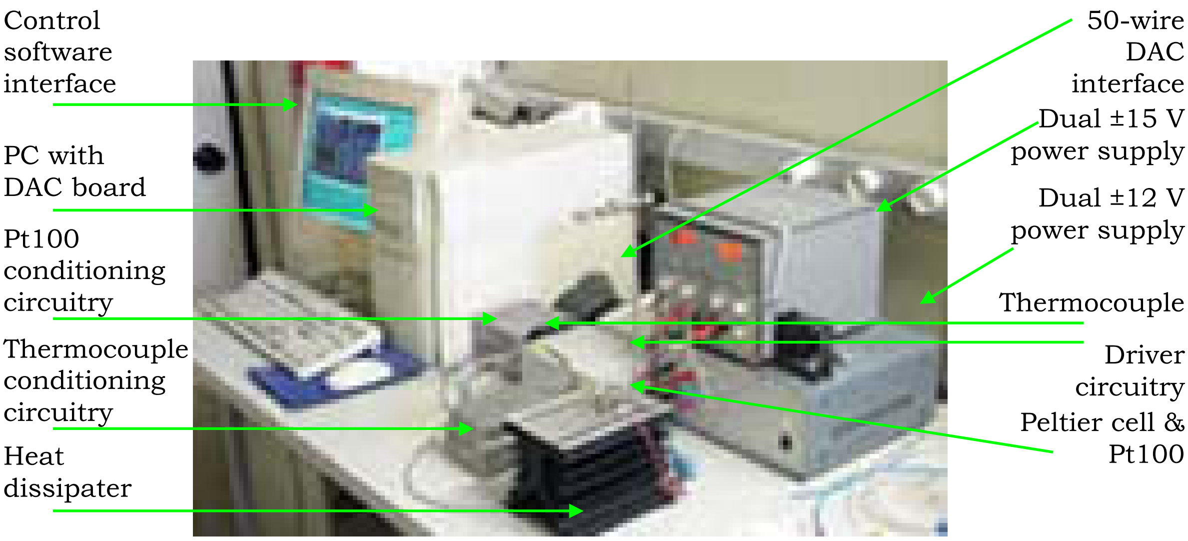

Figure 82 - External fast thermocycler set-up.

Characterization

After much tinkering with noise and RF interference problems and some board redesigns, the Pt100 and thermocouple sensors were calibrated by immersion in a de-ionized water bath with commercial thermometers (see *Materials and Methods*, p.310) and the system was tested for functional characterization. Figure 83 is a graphical display of a typical PCR cycling operation using the above-depicted assemblage.

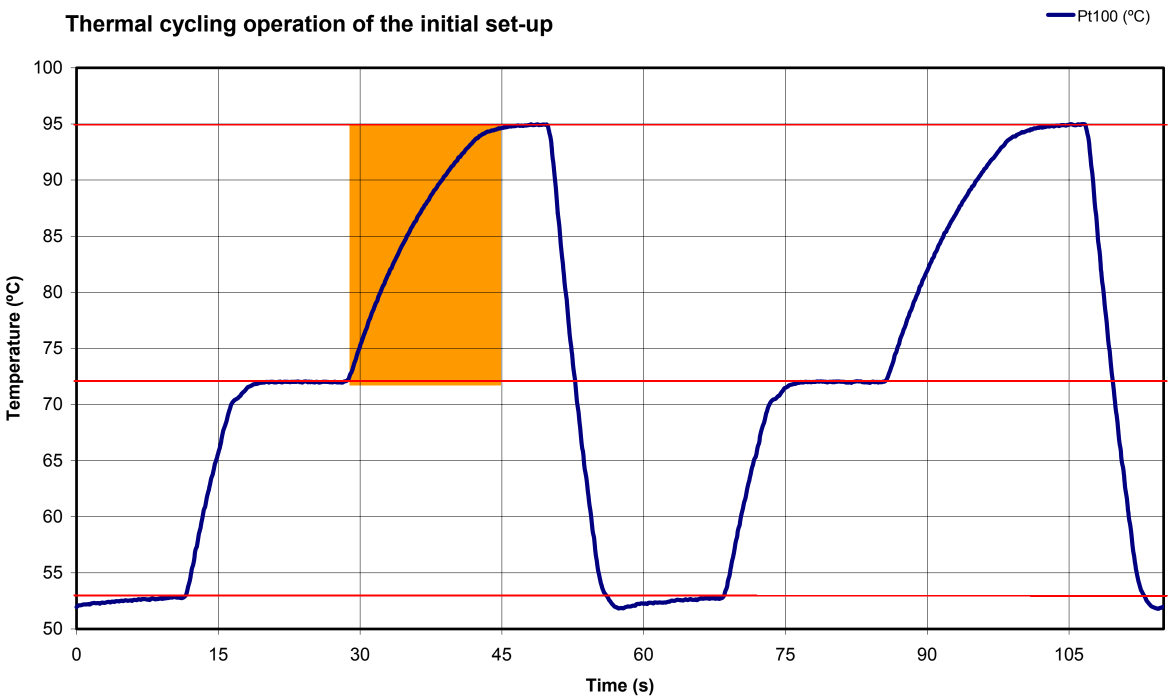

Figure 83 - Cycling operation of the initial setup. Set temperatures are marked in red, while the slow transient time (>15 s) near the Peltier cell operational range is background lighted in orange.

Peltier cell operational range

From Figure 83 it can be readily deduced that, although the system operated quite efficiently, with neat transients and fast stabilization times provided by the dual PD-PID control, there was an apparent problem with the Peltier range of operation, which could not deliver fast transient times for 50-60 to 72 °C and 72 to 95 °C transitions at room temperature, since it was running quite close to the limit of its operational range (68 °C differentials, see *Materials and Methods*, p.307). Nonetheless, although this posed a problem for fast thermocycling, it should not be much of a hindrance in conventional PCR operation, which has similar transient times.

Hence, after some unsuccessful PCR amplification assays, and after repeatedly recalibrating the system, another source of possible temperature inaccuracy was detected. It must be noted here that temperature accuracy was a foremost requisite for the fast external thermocycler, because problems concerning temperature control had to be discarded beforehand in order to permit independent coping with other inhibitory or limiting factors stemming from the peculiar circumstances of PCR-chip amplification (different surface-to-volume ratio, presence of inhibitors, etc.).

Temperature accuracy

Peltier cell surface temperature gradient

Monitoring the temperature of the Peltier cell surface with an external Pt100 sensor connected to an Integra Series 2700 Multi-meter Data Acquisition System (*Keithley*), it was seen that the Peltier cell surface displayed the temperature distribution shown in Figure 84, which could probably interfere with PCR operation, since temperature gradients of up to nearly 3 °C were observed across the Peltier cell surface.

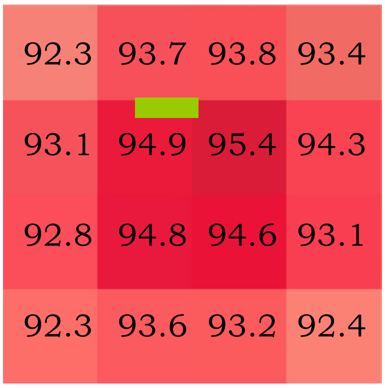

Figure 84 - Temperature distribution at the Peltier cell surface for a set temperature of 95.0 °C, as measured with an external Pt100 sensor. The green rectangle represents the position of the PID-control Pt100 sensor during the measurements.

The physical explanation of the temperature gradient observed at the surface of the Peltier cell lies in the inner workings of such a device (see *Materials and Methods*, p.305). As already mentioned, a Peltier cell is a composite parallel array of independently paired Peltier junctions. Hence, the flow of a precise amount of electric current through each Peltier element unequivocally determines a temperature differential between the two Peltier junctions. The temperature gradient on the Peltier cell surface originates from the fact that the underlying dissipater will not present a uniform temperature distribution at the cold junctions (being slightly hotter at the center of the Peltier cell than at its extremes) and, hence, this non-uniform distribution will be reflected on the hot side.