Tuning the Explore 100 Reflow Oven Controller

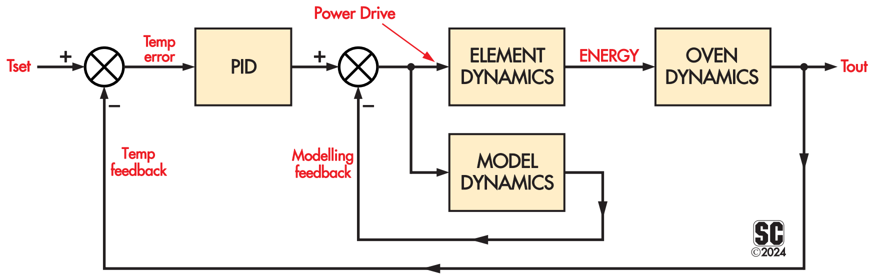
My controller generates a 'tick' interrupt once every 500 milliseconds, which sets (flag = true). The main loop waits on this flag, sets the flag back to false ready for the next interrupt, sums 32 readings of the thermocouple amplifier output voltage connected to pin 77, scales this to a temperature reading, and then takes appropriate control action. The whole process takes about 21ms, so the program loops checking for the flag for the ~479ms remaining until the next tick.

Various screen pages allow the operator to set up the State parameters, and PID & Compensation parameters for this control action.

A significant difficulty with controlling these ovens, is that of thermal overshoot – the temperature may take maybe 15-20 seconds to even start to move when full power is applied. Then, when power is removed the oven temperature will continue to rise, for up to another 2 minutes. Clearly, a simple PID system will lack the data to improve things much – as well the behaviour is extremely non-linear, as the oven temperature may increase at a rate of say 1.2°/second, but fall at only 0.1 deg/sec, or less – you can't take energy out of the oven…!

Looking at the physical system, the energy from applied power first goes into the elements, heating them up with the inertia of thermal mass. Heat then transfers into the oven internals by convection from the elements – represented by a delayed response time-constant. The situation becomes more complex once the elements have become red-hot, as a point is reached at which the energy transfer rate from the elements via radiation as well as convection into the oven interior, becomes equal to the power input. The red-hot element will continue to heat the oven for some considerable time, even once the power supply is removed.

I've come up with a compensation system, that helps largely reduce the temperature overshoot. This works using an "internal feedback" model, to simulate what is going on, as shown in the following diagram.



This system accumulates a 'count' of the energy provided to the elements, by adding an 'Energy Base' number, progressively decreased by a 'decay' factor to compensate for non-linearity, to the accumulator every ½ second while power is applied. This accumulated figure, Esum, is then subtracted from the PID figure each cycle.

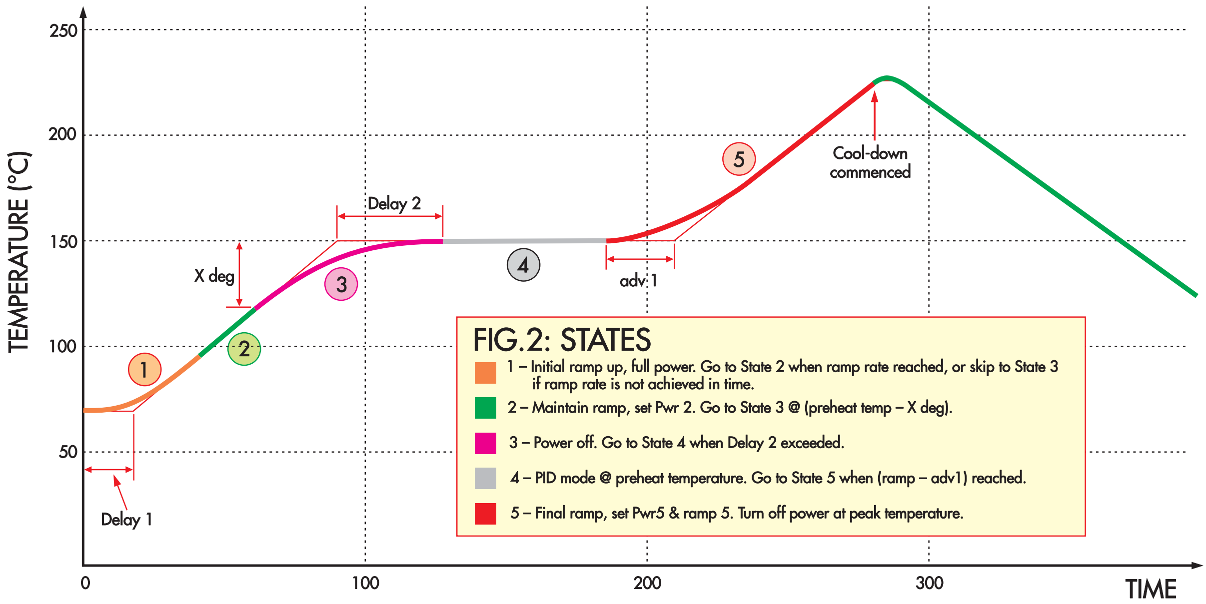
The gain of this can be set from a screen, such that once enough energy is estimated to have been supplied to reach the required temperature, the (PID – Esum) figure will go to zero, causing power to be switched off. At this point the oven temperature keeps rising, with the power being held OFF.

However, in this holding mode the Esum value is also decayed at a rate to ensure sufficient time elapses for the temperature to 'settle' before power may be re-applied – this figure is 'looked up' from a data table and is scaled by the 'Dec scaling' factor. A larger scaling factor results in a more rapid decrease of the accumulated value holding off the re-application of power.

This works very well, with final temperature overshoot often less than 2° or so and undershoot about 1° about the set-point when in the set-point temperature controlling mode, once tuned.

Following some experiments, I came to realise that PID control will not work well for much of the Reflow mode of operation – and much of the work I had done amounted to little more than open-loop control. In the end, to simplify adjustments by the operator I have implemented what could be best described as a "linear State engine" – which only uses PID control in State 4.

During REFLOW, this new system progresses through 5 'STATES' – see the following:



Referring to this diagram, the thin red line represents a profile 'template' – however this is not used to determine set-point temperatures. The term 'Delay1' is pre-set on the 'Profile' GUI screen, by the operator. Initially, full power is applied to the oven (State 1). The aim of this setting is for the actual measured temperature to become tangent to the initial ramp – if it cuts through the ramp, the delay1 figure should be increased – however, anything within a couple of seconds is ok. This State terminates once the temperature ramp rate reaches 1.2°C/second, matching the template.

State 2 aims to maintain this slope – if your oven is more powerful than mine, you can edit the value of the 'pwr2' figure (in the code) to a figure less than 100%. This is again not critical. If your oven cannot reach the slope, you could increase delay1 – State 2 will be skipped if necessary.

When the temperature gets within 'Xdeg' of the preheat temperature, the oven switches off, but the temperature continues to rise. The aim of adjusting Xdeg, is to try to have the temperature run near-tangent to the preheat temperature (constant temperature).

After a time 'delay2' has elapsed from the ramp/flat intersection, the system will switch into State 4. In this state, PID is active at the preheat set-point. At the point 'Adv1' ahead of the final up ramp, the system switches into full power (or pwr5, if your oven is powerful). The controller will switch off power shortly before the peak temperature setting is reached.

A typical, tuned, reflow curve is shown below. Of course, it is still necessary to open the oven door about 20mm or so once the peak temperature is reached (the controller beeps to ask for the door to be opened). There is a little skill in getting this right – I find I need to open the door about 25mm – for maybe 5-10 seconds – then open it fully. It is recommended that the cooling curve should not exceed 6 deg/sec – the target curve has been set at 0.8 deg/sec, because after the initial drop, the oven is battling to achieve this. The door should be opened carefully to avoid vibration to the oven – you could also switch OFF the convection fan – because at this time the solder is all still molten.

Please note: The initial flat section of the curve is set by 'Delay1' – to match the first ramp to the oven's capability. Whilst this might appear 'contrived' it isn't – as the remainder of the target curve is adjusted as necessary to suit the length of the preheat zone, etc.

Also note: To provide consistency, I always firstly preheat the loaded oven to 70-75°C (and let this settle before pressing the 'Reflow' button). This is deemed ok, as the SMD parts are considered safe at 85°C. The software first measures this initial preheat temperature and uses this to adjust the target curve to suit. Obviously, if the oven is still heating strongly when the 'Reflow' button is pressed, the initial transition into the first ramp will happen more quickly, so it will overshoot the curve (although, this is of no dire issue, as Stage 3 is entered a set temperature before the flat preheat region is reached, to allow the ramp to flatten).

My oven is small (20L but will reflow a 230 x 230 mm board ok) and rated at 1500 watts (although I 'over-boost' this a little). My theory is that a smaller oven has less surface and mass to heat, however I would expect an 1800-watt version to respond more quickly.

Within the software, you will find 2 x power settings (Pwr2 and pwr5, expressed as a percentage). For my oven, all of these are set to 100% - however for a more powerful oven you may need to reduce the ramp maintenance figure Pwr2, and even Pwr5 if the final ramp rate is exceeded significantly. Neither Phil's nor my oven can reach the final 1.2°C/second ramp rate setting – in the interests of aesthetics I have therefore provided a GUI control (under 'Profile'), to allow this ramp to be adjusted down a little – the recommended maximum heating rate is 3°C/second, but the minimum is simply to ensure the period the temperature remains above 217° is not too long. The maximum recommended cooling rate is 6°C/second – but we are unable to approach this, except perhaps if the door is opened to quickly initially.

If you have the inclination, this alternative controller is easy to set up, and worth a try.