KEITHLEY

Keithley Instruments, Inc.
28775 Aurora Road
Cleveland, Ohio 44139
(440) 248-0400

Models 2510 and 2510-AT

Introduction

This addendum to the Model 2510 Instruction Manual is being provided in order to supply you with the latest information in the least possible time. This information concerns the Model 2510-AT, which includes PID temperature control loop autotuning. This addendum also contains information on a change in the maximum allowed PID derivative loop constant (Kd, Ki, and Kp) values, which applies to both the Models 2510 and 2510-AT.

PID autotune

The Model 2510-AT autotune algorithm provides the user with a tool to help in tuning the Model 2510-AT Temperature PID loop. It is intended to give a set of PID tuning parameters that will give close to the optimum system performance, but it may not result in ideal tuning parameters. These PID parameters can be improved upon by iterative adjustments, which are covered later in this addendum.

Autotune operation

The Model 2510-AT PID autotune algorithm obtains its information from the system by forcing a step function in voltage across the TEC and then observes the system response to that step function. The Lag and Tau times of this response waveform are then extracted and applied to a modified version of the Ziegler-Nichols tuning equations.

Response options

There are two different options for the autotune function: minimum settling time and minimum overshoot. The discussion below compares these two options based on short and long Lag and Tau (τ) times.

Short Lag and Tau time example

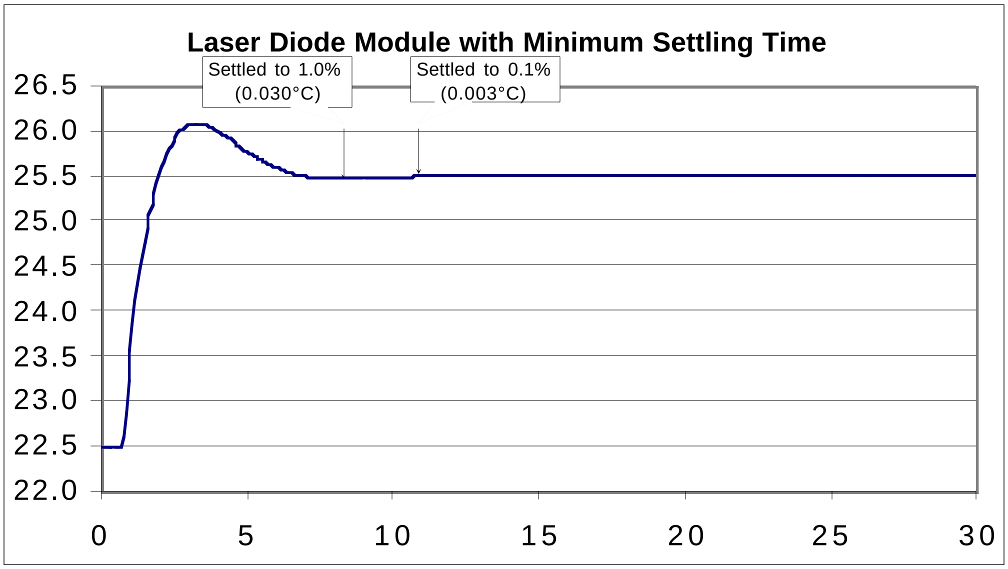
Typical response characteristics for the two response options are shown in Figure 1. In these examples, the Laser Diode Module has a Lag time of 0.77 seconds and a Tau of 7.70 seconds and is subjected to a +3°C step in temperature setpoint. For the minimum settling time plot, the temperature overshoot occurred at 3.274 seconds (with a peak value of 26.09°C). It settles to within ± 0.1% of the final temperature (0.003°C) in 11.14 seconds. In the minimum overshoot case, the temperature overshoot occurs at 4.96 seconds (with a peak value of 25.67°C). It settles to within 0.1% of the final temperature (25.503°C) at 27.32 seconds. (See Table 1.)

Table 1
Response time comparison example 1 (laser diode Lag Time 0.77sec, Tau Time 7.70sec)

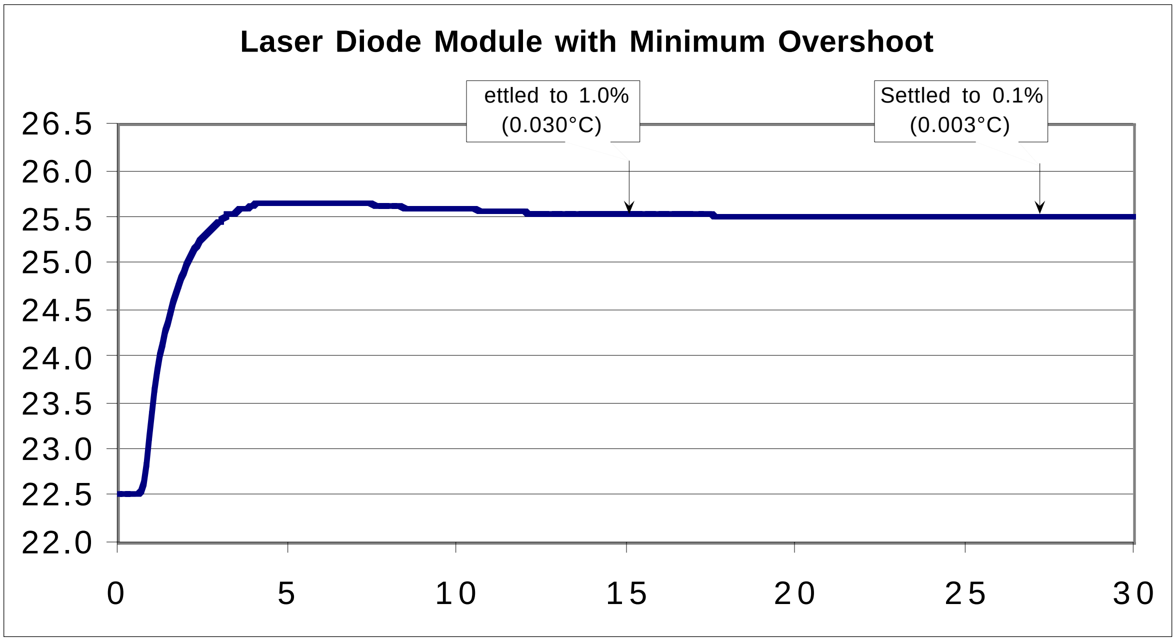
Condition	Minimum Settling Time	Minimum Overshoot
Max Overshoot:	26.09°C @ 3.274 Sec after Step	25.67°C @ 4.96 Sec after Step
Settle to 0.1%:	25.497°C @ 11.14 Sec after Step	25.503°C @ 27.32 Sec after Step

Figure 1

Response comparison example 1 (short Lag and Tau times)



A. For a 3°C step: Settled to ±1.0% (±0.030°C) in 8.54 Seconds
Settled to ±0.1% (±0.003°C) in 11.14 Seconds



B. For a 3°C step: Settled to ±1.0% (±0.030°C) in 15.32 Seconds
Settled to ±0.1% (±0.003°C) in 27.32 Seconds

Long Lag and Tau time example

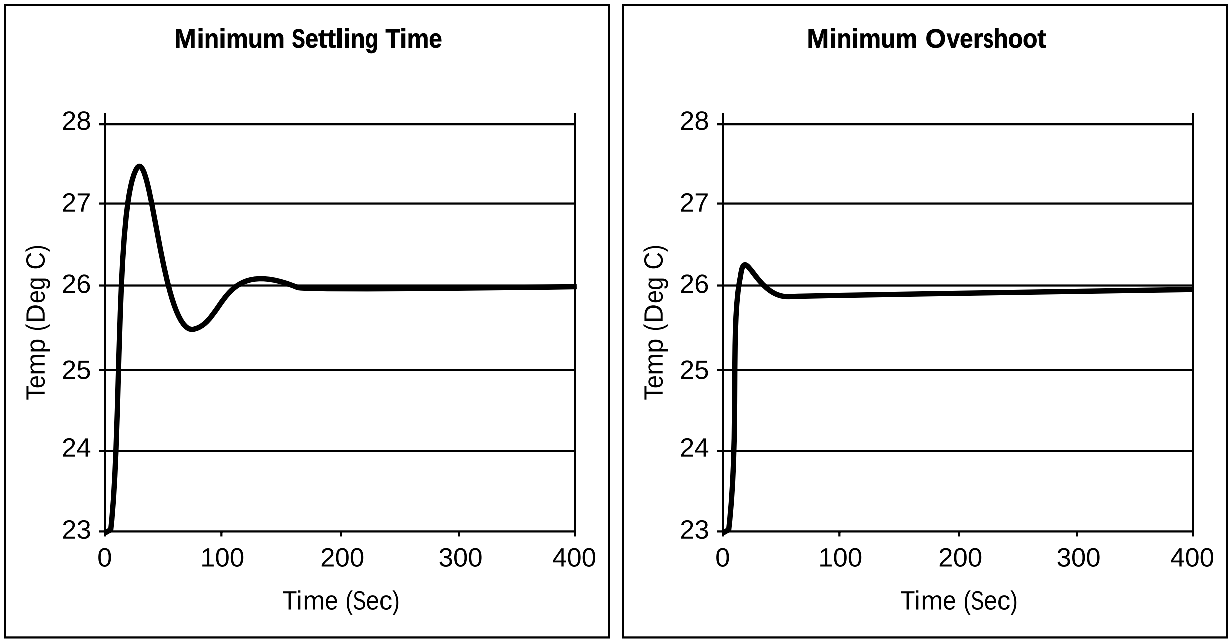
In order to demonstrate the benefits of response tuning, refer to the example of a system with very long Lag and Tau times (shown in Figure 2). In this example, the load has a Lag time of 11.0 seconds and a Tau of 107.0 seconds and is subjected to a +3°C step in temperature setpoint. For the minimum settling time plot, the temperature overshoot occurs at 26.495 seconds (with a peak value of 27.450°C). It settles to within ± 0.1% of the final temperature (±.026°C) in 149.1 seconds. In the minimum overshoot case, the temperature overshoot occurs at 20.847 seconds (with a peak value of 26.226°C). It settles to within 0.1% of the final temperature (25.974°C) at 521.914 seconds (See Table 2).

Table 2
Response time comparison example 2 (Lag Time 11.0sec, Tau Time 107.0sec)

Condition	Minimum Settling Time	Minimum Overshoot
Max Overshoot:	27.450°C @ 26.495 Sec after Step	26.226°C @ 20.847 Sec after Step
Settle to 0.1%:	26.025°C @ 149.100 Sec after Step	25.974°C @ 521.914 Sec after Step

Clearly the minimum settling time option settles much more quickly to 0.1% than does the minimum overshoot version.

Figure 2
Response comparison example 2 (long Lag and Tau times)



Autotune limitations

The Model 2510-AT autotune algorithm assumes that the system response to a step function in the TEC voltage is an exponential temperature rise of the form:

$$T_{system} = T_{initial}$$

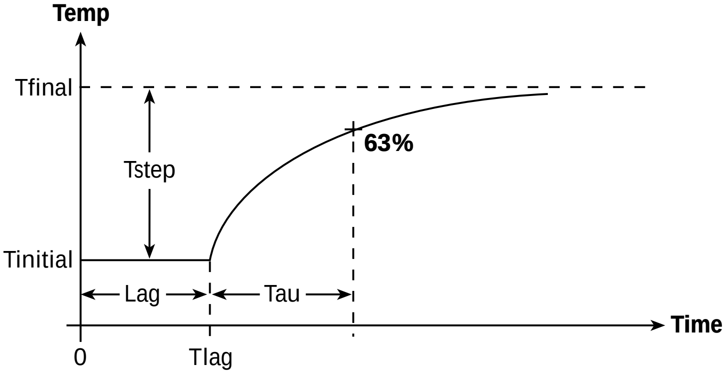
(for $t < t_{Lag}$)

$$T_{system} = T_{initial} + T_{step}\left(1 - e^{-\left(\frac{(t - t_{lag})}{\tau}\right)}\right)$$

(for $t \geq t_{Lag}$)

This relationship is depicted graphically in Figure 3.

Figure 3
System response to step function



In this case T_{step} is positive, but it could also be negative. Temperature responses of a different mathematical model will cause the autotune function to fail (or give unpredictable results). Once the Lag and Tau times are extracted from the sampled data taken from the voltage step response, the tuning constants can be calculated. Inherent limitations that will cause the tuning function to fail include:

- A system Tau time of less than 1 second or greater than 470 seconds
- A system Lag time greater than 0.6 × Tau
- Ambient temperature outside the range protection limits (programmed by the user)
- Ambient temperature movements during the autotune sequence of greater than ±1°C
- Noisy temperature measurements
- Reaching any temperature, current, or voltage limits during the final autotune voltage step test

Practical autotune considerations

There are several practical considerations to take into account when using autotune. Each of these is outlined below.

TEC module gain

The gain of a TEC module varies with operating temperature and increases with increasing temperature. This variation can be as much as 7:1 over the operational temperature range of the device. The principle reason for this change in operational efficiency is that the I^2R heating generated by the normal operation of the Peltier junction adds to the efficiency of the junction when pumping heat into the load, but it subtracts from the efficiency of the junction when cooling the load. This change in Peltier gain will cause the overall system characteristics (and hence the tuning constants which will control it) to change appreciably over the entire operating temperature range of the system. It is for this reason that you should tune the PID loop at the actual system operating temperature. Ideally, if the temperature changes by more than 10°C, you should create a new set of tuning constants for proper control.

Large temperature steps

If the temperature step is very large, tune the PID loop at the highest temperature that will occur during normal operation. This tuning method will ensure stability of the PID loop at the higher operating temperatures. If you use these same tuning constants at the lower temperatures, response at the lower temperatures will be slower (due to the lower gain of the TEC). If the loop is tuned at the lower temperatures (to give a better response), and these same tuning constants are used at the higher temperatures, the PID loop may become unstable at the higher operating temperatures (due to the higher gain of the TEC at these higher temperatures). Averaging the constants from both extremes is not advisable as poor performance will result at both operating points — more overshoot and ringing at high temperatures; slower response at lower temperatures.

PID fine tuning

If the autotune algorithm does not yield the desired response, use the PID loop constants created by the autotune algorithm as a starting point, and modify them as needed to obtain the desired performance. The three PID loops constants perform the following functions:

- **Kp (Proportional gain constant)** — This constant “pushes” the system to its final value. Lower numbers create a slower response while larger numbers help increase the response speed. Values too large may cause the system to oscillate and/or become unstable.
- **Ki (Integral gain constant)** — This constant is responsible for how fast the system settles to its final value, as well as how much overshoot occurs. Low values create a long settling tail with minimal overshoot, while large values settle much more quickly but with larger overshoot. Values too large may cause the system to become unstable (and most likely oscillate).
- **Kd (Derivative gain constant)** — This constant helps control the slew rate / dynamics of the output response waveform. Small values allow the temperature to move as quickly as possible, while large values control how rapidly the system responds. Values that are too large may cause the system to become unstable.

Using autotune commands

Autotune command summary

Table 3 summarizes Model 2510-AT autotune commands. Note that all of these commands are part of the :SOURce[1]:TEMPerature subsystem, which is fully described in the Section 10 of the Model 2510 User's Manual.

Table 3
Autotune commands

Command	Description
:SOURce[1]	Path to SOURce[1] commands.
:TEMPerature	Temperature commands.
:ATUNe	Auto Tune commands.
:LCONstants	Temperature PID loop constants.
:MSETtle	Use minimum settling time criteria.
:GAIN?	Query temperature gain constant.
:DERivative?	Query temperature derivative constant.
:INTegral?	Query temperature integral constant.
:TRANsfer	Accept and install MSET constants as defaults.
:MOVershoot	Use minimum overshoot criteria.
:GAIN?	Query temperature gain constant.
:DERivative?	Query temperature derivative constant.
:INTegral?	Query temperature integral constant.
:TRANsfer	Accept and install MOV constants as defaults.
:TAU?	Query tau value.
:LAG?	Query lag value.
:STARt <n>	*Set start temperature value.
:STARt?	Query start temperature value.
:STOP <n>	*Set stop temperature value.
:STOP?	Query stop temperature value.
:INITiate	Initiate autotune procedure.
:SYSTAU	Program rate for SHOR, MED, or LONG Tau loads.
:SYSTAU?	Query Systau value.

*Temperature value must be between the upper and lower temperature protection limits.

Basic autotune procedure

The general procedure below outlines the basic steps for using the autotune commands to tune the PID loop. Keep in mind that this procedure is intended only as a starting point, and some experimentation may be required to obtain the desired results.

1. Send the SOUR1:TEMP:ATUN:STAR command to program the start temperature value. For example, the following command would set the start temperature to 45°:

   ```
   SOUR1:TEMP:ATUN:STAR 45
   ```

2. Send the SOUR1:TEMP:ATUN:STOP command to program the stop temperature value. For example, the following command would set the stop temperature to 50°:

   ```
   SOUR1:TEMP:ATUN:STOP 50
   ```

3. Set the maximum temperature protection limit; for example:

   SOUR1:TEMP:PROT:HIGH:LEV 50

4. Set the minimum temperature protection limit; for example:

   SOUR1:TEMP:PROT:LOW:LEV 0

5. Set the maximum TEC voltage limit; for example:

   :SOUR1:VOLT:PROT:LEV 2.8

6. Set the maximum TEC current limit, for example:

   :SENS:CURR:PROT:LEV 1.2

7. If the system TAU is known to be greater than 100 seconds, send the SOUR:TEMP:ATUN:SYSTAU MED command to allow for the slower system response. If the system TAU is known to be greater than 200 seconds, send the SOUR:TEMP:ATUN:SYSTAU LONG command to allow for the very slow system response.

8. Initiate the autotune process by sending this command:

   SOUR1:TEMP:ATUN:INIT

9. Wait until the autotune process is complete. The unit will display messages on the front panel to indicate autotune progress. To abort the autotune procedure, you must cycle power.

10. Once the autotune procedure is complete, you can use the MSETtle or MOVershoot GAIN, DERivative, and INTegral queries to request the autotune gain, derivative, and integral constants for minimum settling or minimum overshoot as desired.

    If you are satisfied with the results, you can send the following commands to transfer these tuning constants to the appropriate registers for usage:

    - :SOUR:TEMP:ATUN:LCON:MSET:TRANS — Accept and install MSET constants as defaults.
    - :SOUR:TEMP:ATUN:LCON:MOV:TRANS — Accept and install MOV constants as defaults.

    NOTE To retain these tuning constants after cycling power, be sure to save them as power-on defaults by using the SAVESETUP selection in the main MENU.

11. If the results are not exactly what you ideally want, use the following manual PID loop constant commands to fine tune the system using the autotune values as a starting point:

    - :SOUR:TEMP:LCON:GAIN <gain> — Set manual PID gain constant.
    - :SOUR:TEMP:LCON:DER <derivative> — Set manual PID derivative constant.
    - :SOUR:TEMP:LCON:INT <integral> — Set manual PID integral constant.

    Also, you can use the TAU? and LAG? queries to request the corresponding Tau and Lag constant values.

Maximum PID constants values

The maximum value for all of the PID constants (kp, ki, and kd) has been changed to 100000. These constants are programmed from the front panel PID menu selections in the temperature, voltage, current, and resistance configuration menus (Tables 3-1 to 3-4 in Section 3 of the manual). See Section 10 of the manual for details.