Faculty of Enigeering

Cairo University

MODIFICATIONS TO THE TEXTBOOK PID

Presented for ELC 4046 Lab 25

Presented to:

Dr. Ahmed Ismail

T.A: Mohamed nade

TEAM MEMBERS (4th Year Electronics and Electrical Communication Engineers)	
مجدي أحمد عباس عبد الحميد الابرق	Sec: 3 / I.D: 9210899 / BN: 32
محمد إبراهيم محمد على	Sec: 3 / I.D: 9210906 / BN: 34

LABORATORY EXERCISE 25
(MODIFICATION TO THE TEXTBOOK PID)

OBJECTIVE: To demonstrate the following modifications:

Derivative on measurement

Internal filter

Interacting vs noninteracting controller

Proportional on measurement

Setpoint softening

Bump less transfer

PREREQUISITE: Completion of Laboratory Exercise 8 PID Controller Characteristics

1. RUNNING THE PROGRAM

Run **PC-ControLAB**.

Slow down the simulation speed via **View | Scroll Rate** and select a mid-value.

Check the top line of the display. Be sure that you are using the GENERIC process model and the FEEDBACK control strategy.

From the Menu Bar, select **Control | Control Options**.

For Control Algorithm, select **PID, Non-Interacting**.

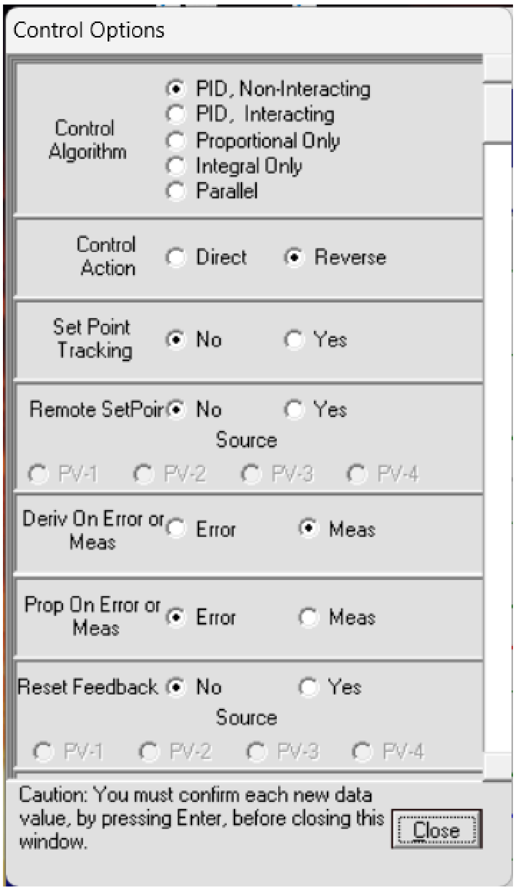


Figure 1: Selecting PID Non-Interacting controller

2. DERIVATIVE ON MEASUREMENT

In **Control | Control Options**, for Derivative on Error or Meas, select **Error**.

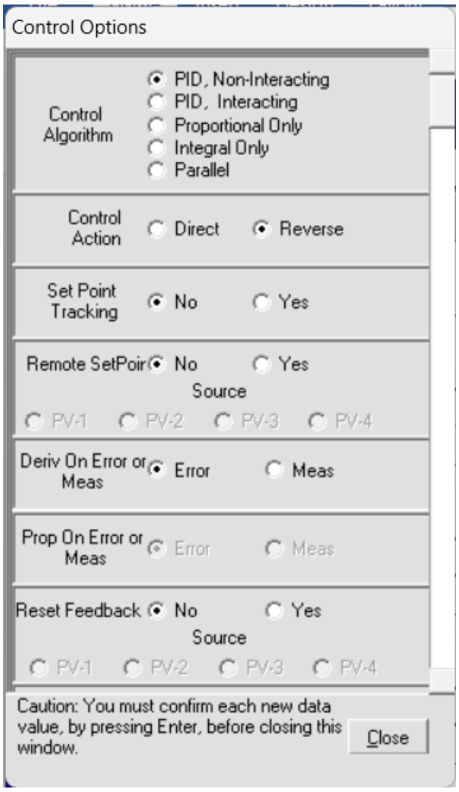


Figure 2: Derivative on error

In the **Tune | Options** tab, set the following:

Reset Action	OFF
Derivative Gain	100.0

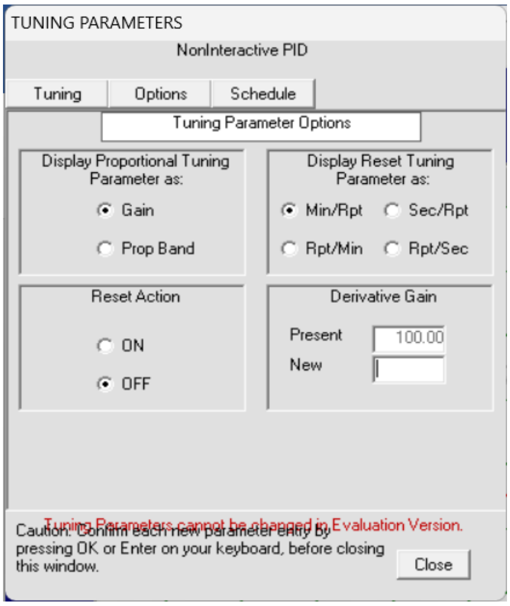


Figure 3: Derivative gain = 100 and reset action is off

In the **Tuning** tab, enter the following tuning values:

Gain: 2

Reset, Min/Repeat: Not relevant (it's OFF)

Derivative, Min: 0.5

Note that the simulation starts with an initial controller output of 35%.

Put the controller in AUTO and change the setpoint to 325.

- A setpoint step change will cause the output to spike to maximum or minimum as the derivative of a step is a delta, as shown in fig. (5) the output instantly spikes to maximum value (100%) due to increase in SP (reverse acting controller) the spike effect is instant and then the fades away and what's left is the proportional kick calculated as following:
- $m = 35\% + \frac{325-275}{500} * 100 * 2 = 55\%$ (as shown in fig. 5)

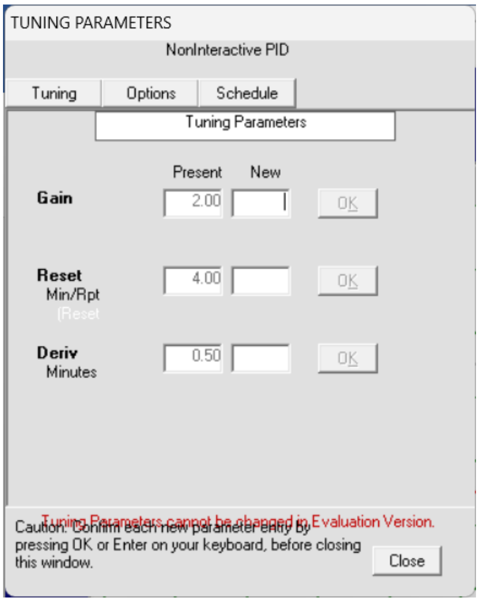


Figure 4: Tuning values

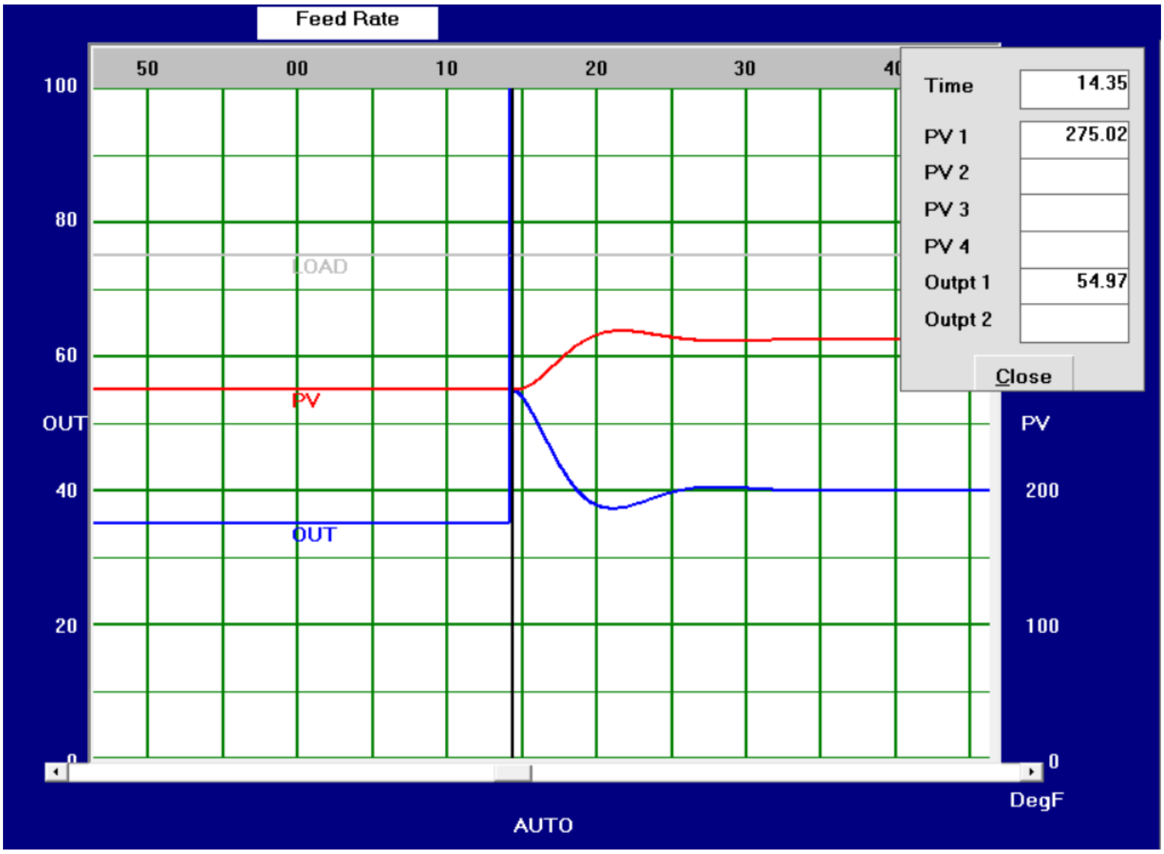


Figure 5: Response after changing setpoint to 325

Observe: Even with this moderate setpoint change, the controller output "spikes" all the way to 100%, then very quickly drops back to a normal operating range. This is called the derivative spikes. In a real-world application, you cannot send such a severe shock to the process.

Put the controller in MAN. Select **Control | Control Options**, and select **MEAS** for Derivative on Error or Meas. Put the controller back to AUTO and decrease the setpoint by 50 Deg F,

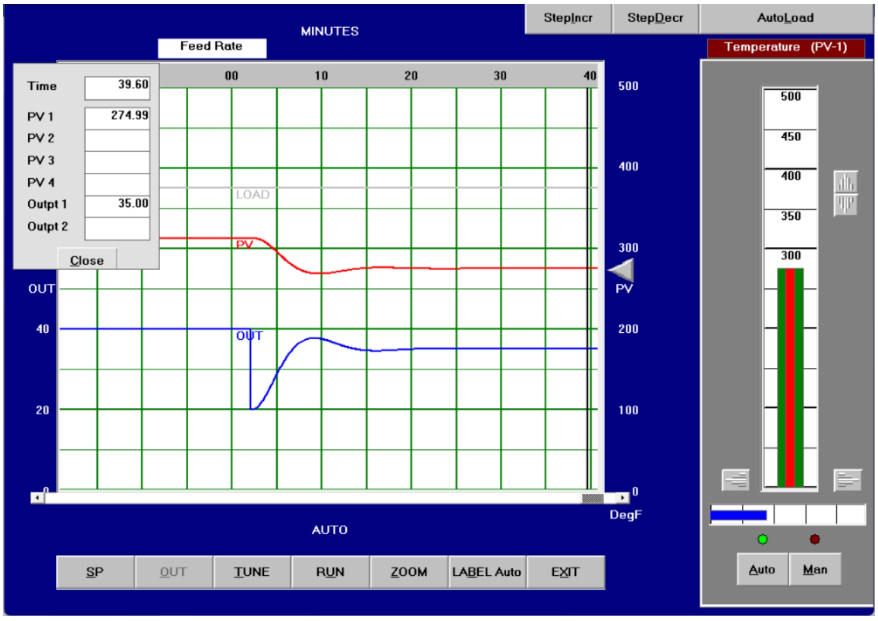


Figure 6: Response of derivative on measurement after decreasing setpoint by 50 Deg F

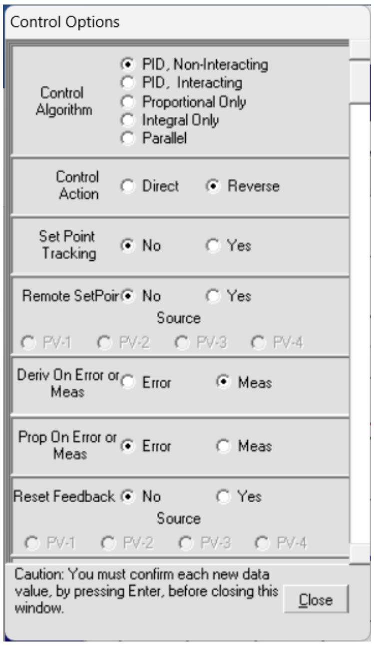


Figure 7: Selecting derivative on measurement

Observe: With derivative on measurement, the immediate response of the controller is the proportional response only – there is no derivative spike.

$$P = kc * (SP - PV) = 2 * \frac{(275 - 325)}{500} * 100 = -20\%$$

What is the value of the proportional response? **-20%**

3. PROPORTIONAL ON MEASUREMENT (P on PV)

Put the controller in MAN.

In **Control | Control Options**, confirm that, for Prop on Error or Meas, **Error** is selected. This is

the default option. In the **Tune | Options** tab, set the following:

Reset Action ON

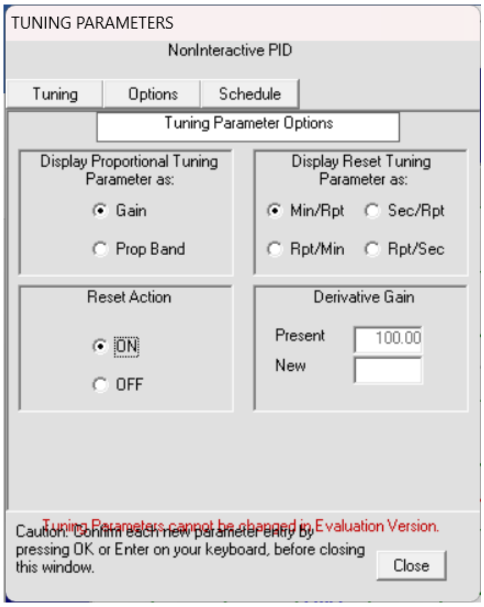


Figure 8: Reset action is on

In the **Tuning** tab, enter the following tuning values:

Gain: 1.4

Reset, Min/Repeat: 7

Derivative, Min: 0

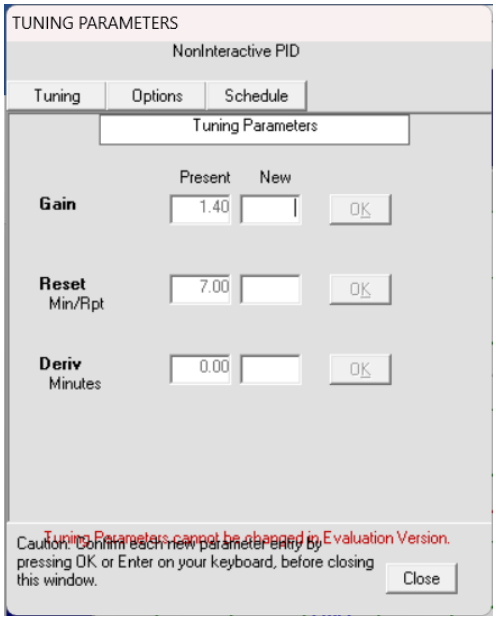


Figure 9: Tuning parameters

- Put the controller in AUTO. Increase or decrease the setpoint by 50 Deg F.

- Increasing the setpoint by 50 causes the output to increase as this is a reverse acting controller and the output is directly proportional to the error which is defined as SP-PV and the setpoint was increased above PV, so the error is positive, the response shown in fig. (10) has a proportional kick due to proportional controller and a ramp due to integral part

- $P = kc * (SP - PV) = 1.4 * \frac{(325-275)}{500} * 100 = 14\%$

- Settling time = time at change – time at 98% of PV change

- = 36.65 - 12.55 = 24.1 min

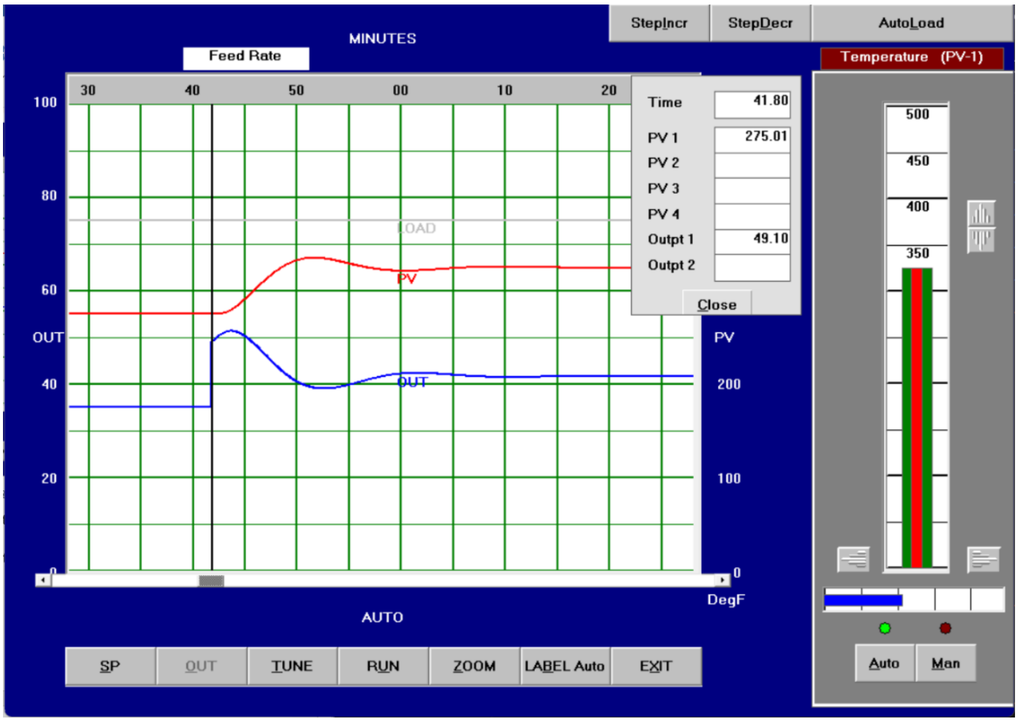


Figure 10: Response of proportional on measurement after increasing setpoint by 50 Deg F

What is the immediate change in the controller output? **14%**

This is called the proportional kick. Verify that it is equal to the controller gain multiplied by the error (in percentage).

Calculate the (±2%) settling time: **36.65 - 12.55 = 24.1 min**

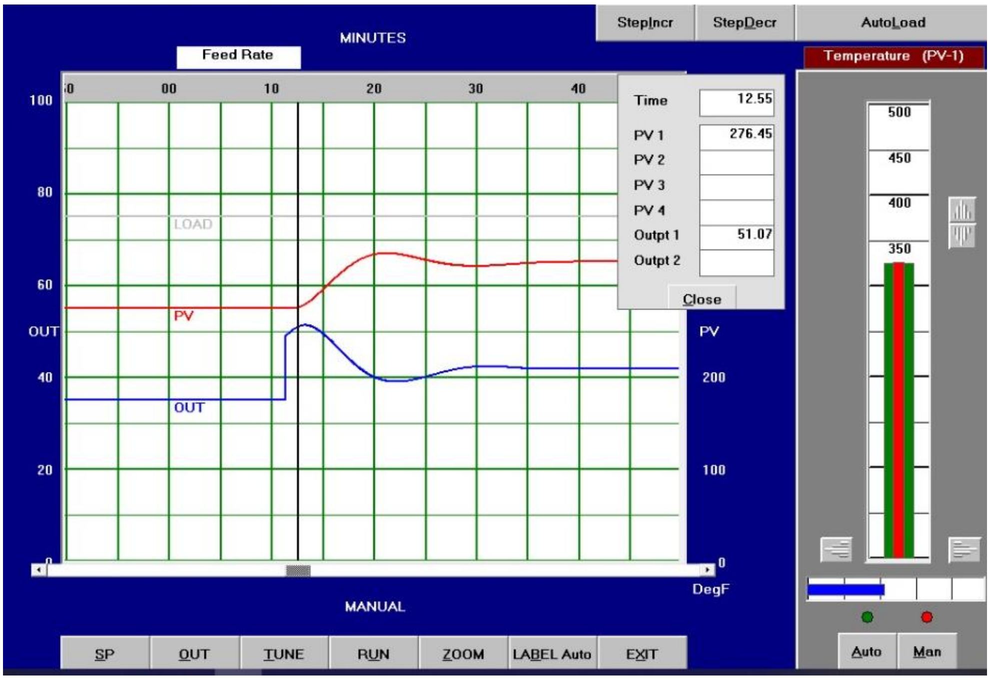


Figure 11: Beginning of PV change

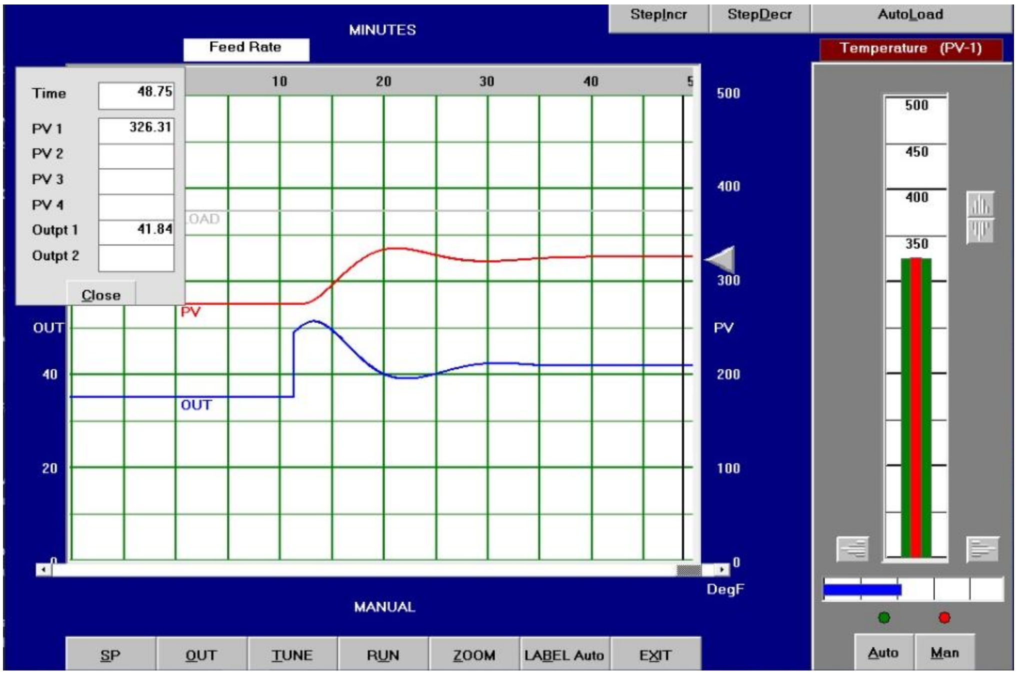


Figure 12: End of PV change

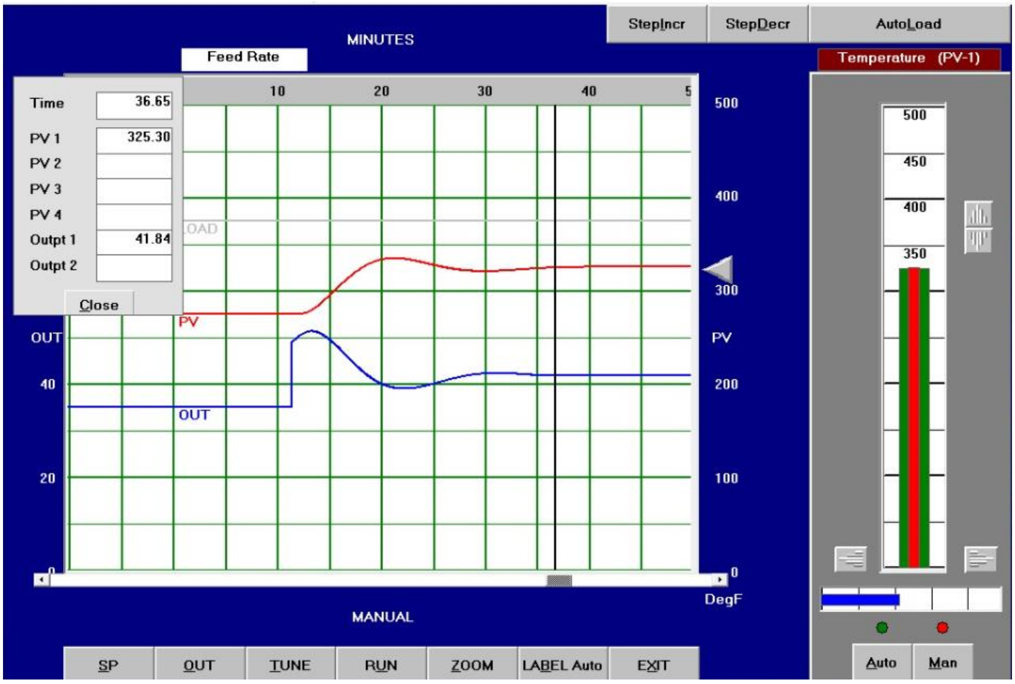


Figure 13: 98% of the PV change

Put the controller in MAN. Select **Control | Control Options**. For Prop on Error or Meas, select **MEAS**. Put the controller back to AUTO and rechange the setpoint by 50 Deg F,

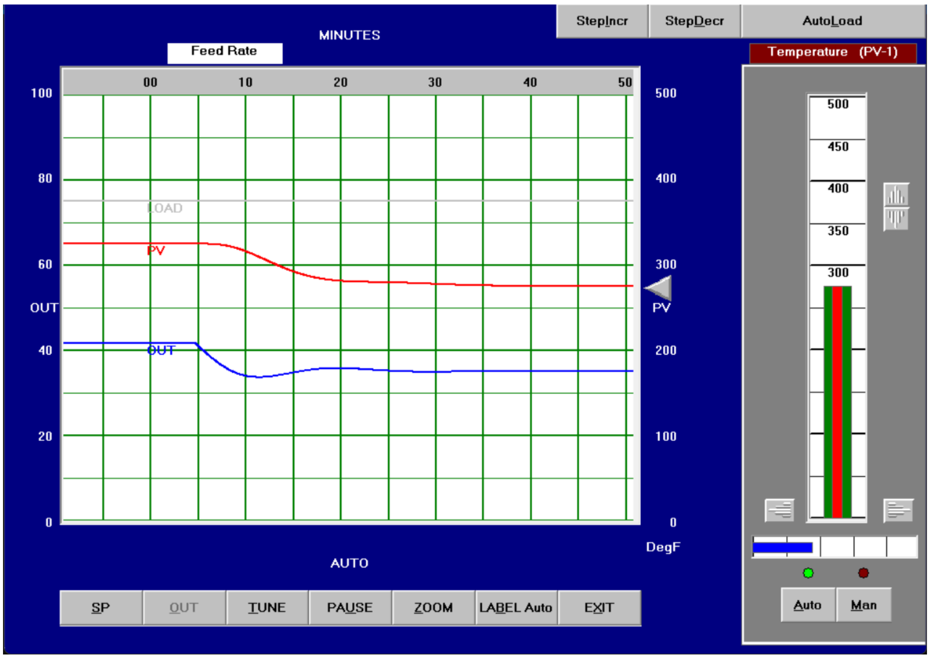


Figure 14: Response after changing SP by 50 DegF for proportional on measurement

Observe: With proportional on measurement, the immediate response of the controller is a smooth gradual change driven by the integral component. There is no proportional kick. This feature is often favored in the oil and energy industries.

What is the settling time in this case? **39.70 - 9.10 = 30.6 min**

This should confirm that this version of the algorithm is slower, i.e. it does not drive the measurement to the new setpoint as quickly as the proportional on error version would do. This is due to the absence of the initial rapid forcing of the process caused by the proportional kick.

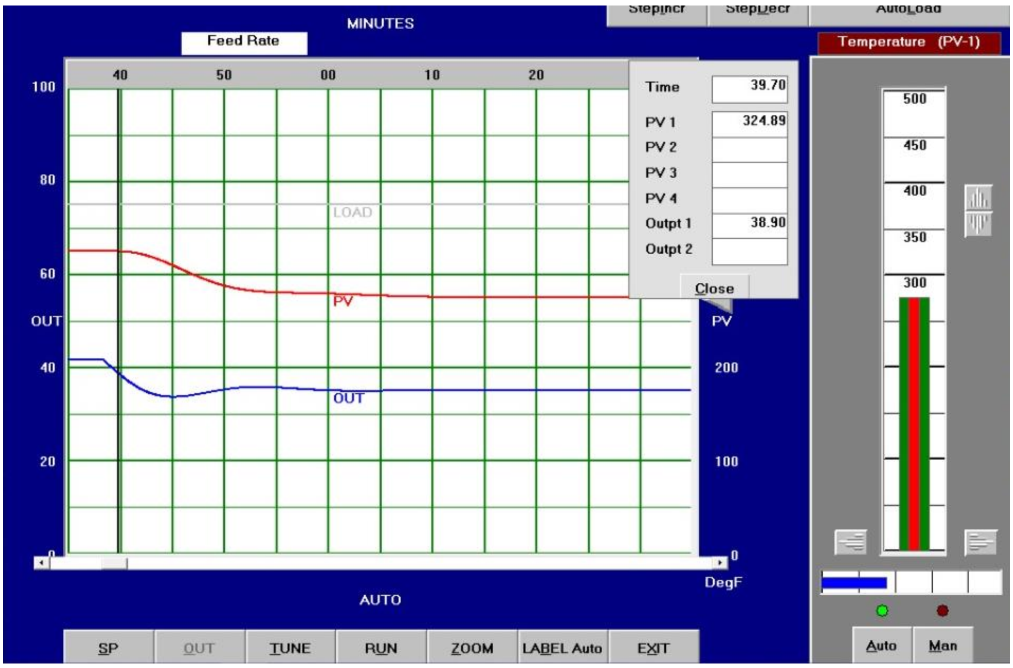


Figure 15: Beginning of PV change

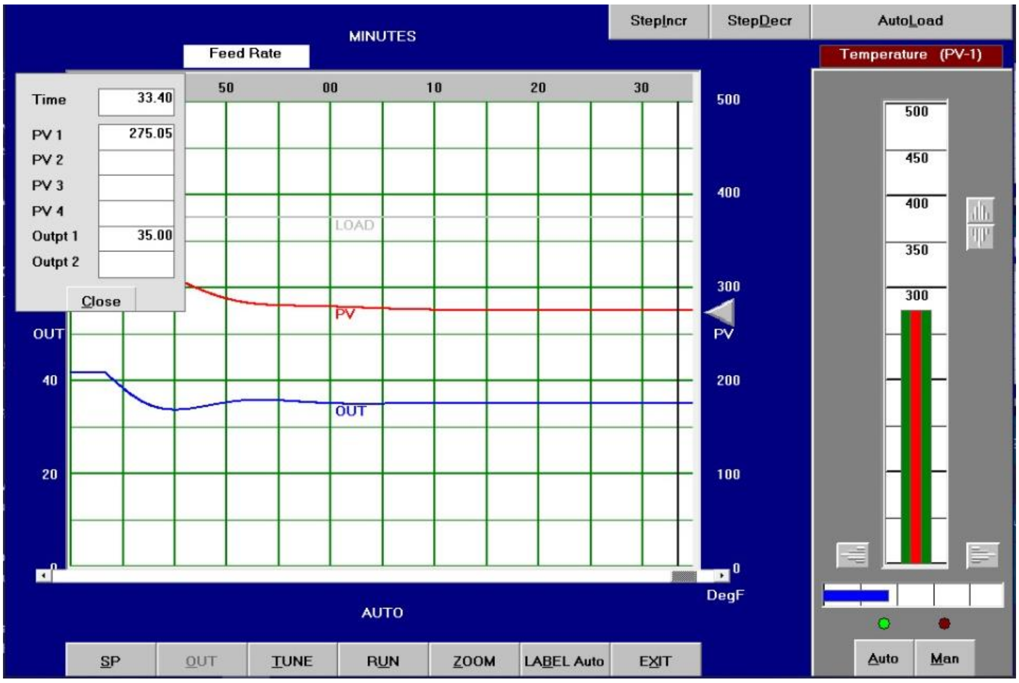


Figure 16: End of PV change

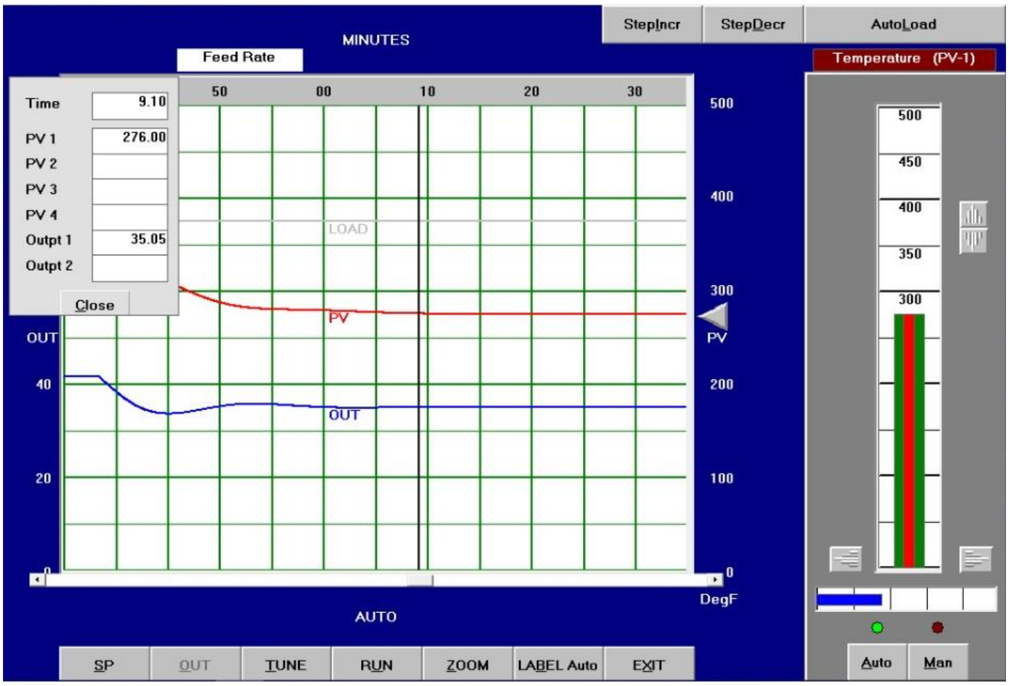
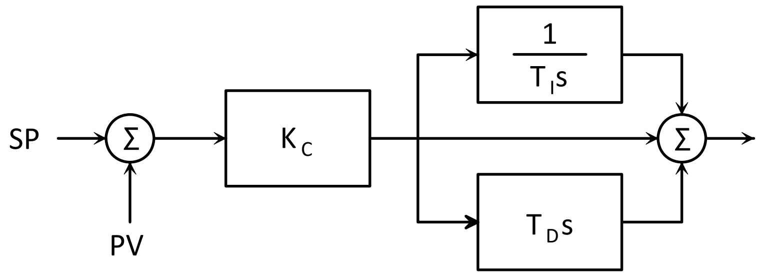


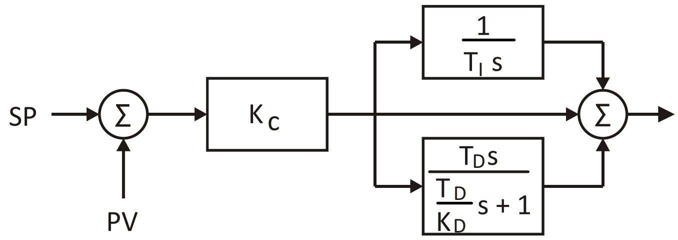
Figure 17: 98% of PV change

4. INTERNAL FILTER

A block diagram for the textbook PID controller is shown below.



Many commercial manufacturers of controllers, however, place a filter, in the form of first order lag, in the derivative portion of the controller. This results in the modified block diagram shown below:



The usual practice is for the time constant of the first order lag to be some fraction of the derivative time. This is accomplished by dividing the derivative time by a factor known as "derivative gain" (K_D). In commercial controllers, the value of K_D is usually in the range of 8 to 15 (with a default value of 10 in most of them).

The purpose of the filter is to reduce the amplification of measurement noise placed on the controller output by the derivative component. With the textbook PID, even a small but sudden change in the measurement will result in a spike in the controller output caused by the derivative unit. With the filter added, the spike height following a change in the measurement will be the value of K_D times the proportional response. The output will then decay to normal with the time constant of the filter.

Select **Process | Initialize** (This returns all options to their original settings.)

Select **View | Display Range | Percent of Span**.

Select **Control | Control Options** and confirm that for Derivative on Error or Meas, **MEAS** is selected. This is the default option.

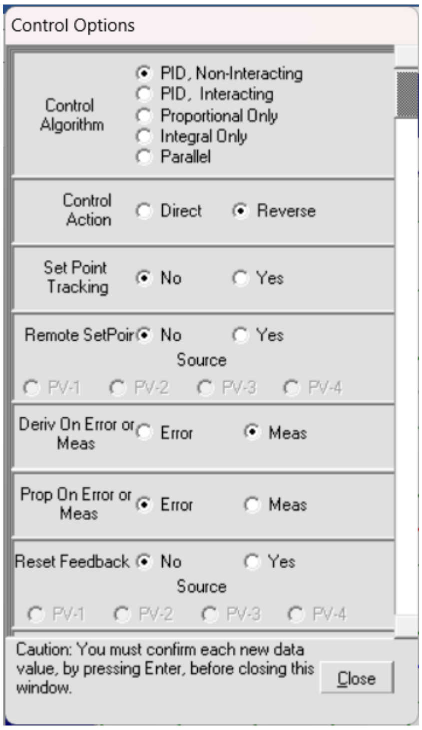


Figure 18: Default option for control options

Now eliminate the derivative filtering action by setting the derivative gain to a large value (i.e., small filter time constant).

In the **Tune | Options** tab, set the following:

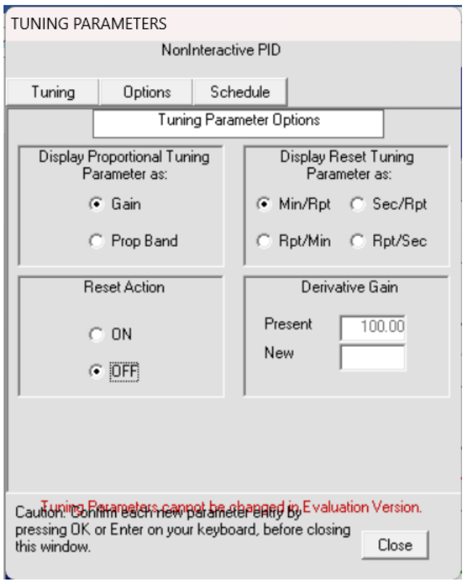


Figure 11: Derivative gain = 100

Reset Action	OFF
Derivative Gain	100.0

In the **Tuning** tab, enter the following tuning values:

Gain: 2

Reset, Min/Repeat: Not relevant

Derivative, Min: 2

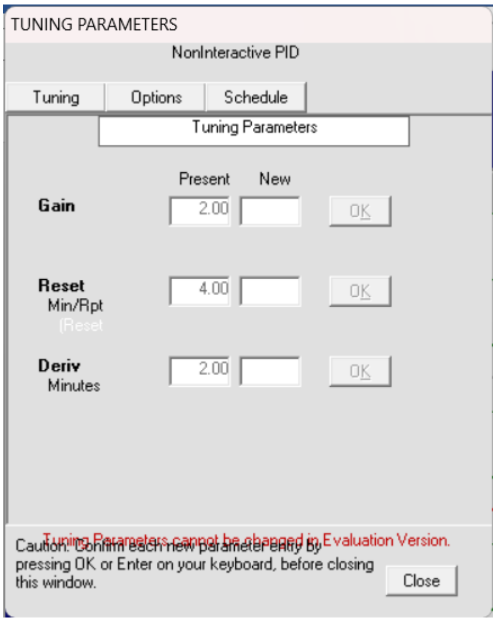


Figure 20: New values of tuning parameters

In **Control | Measurement Options**, select **Yes** for “Use substitute value …?”

Change the controller mode to AUTO.

The controller should be in stable operation with the following values:

Set Point 55

Process Variable 55

Controller Output 35

In **Control | Measurement Options**, enter a substitute measurement value of 54.

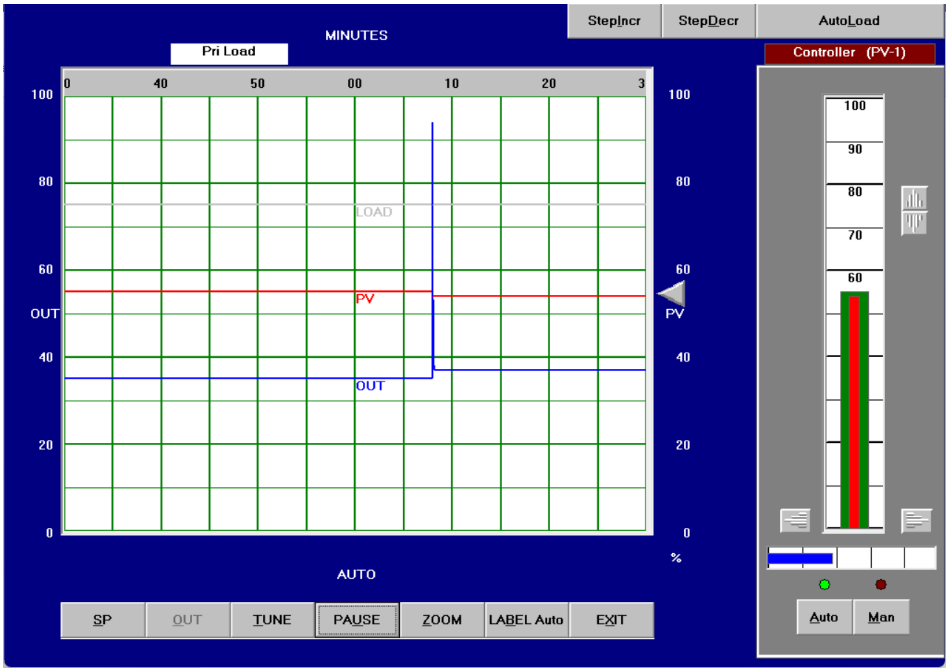


Figure 22: Response after decreasing PV to 54%

Observe: Even with this small process variable change, the controller output experiences an excessive derivative spikes, then very quickly drop back to a normal control range. In a real-world application, This is completely undesirable.

Through the **Tune | Options** tab, set:

Derivative Gain 10.0

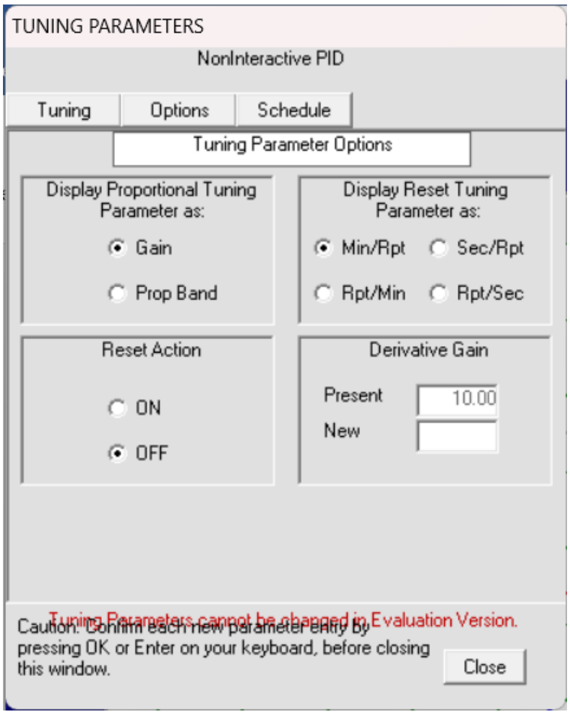


Figure 23: New value of derivative gain = 10

In **Control | Measurement Options**, enter a substitute measurement value of 53.

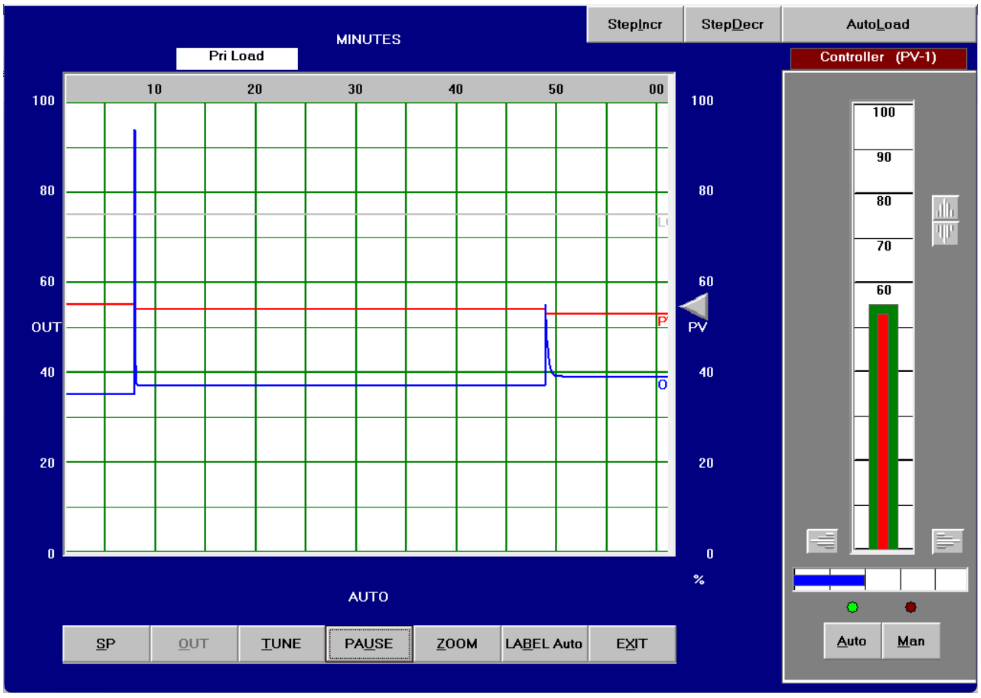


Figure 24: Response after applying filter

Observe: This time the controller output spike is limited.

Calculate the proportional kick:

- Using a PD controller with derivative on measurement and high derivative gain (100) will cause the controller output to instantly spike when the PV changes a small change (from 55 to 54) as shown in fig. (24) then return to 37 which is the effect of the proportional kick kc*error = **2%**

The now reasonable derivative spike should be:

PV change (%) X Controller Gain X Deriv Gain 1 X 2.0 X 10 = 20%

Measure the actual immediate change in the controller output: **18%**

The immediate change in controller output should equal: the proportional kick + the derivative spike

(Quantization error in the digital algorithm may cause some deviation between the actual value and the calculated value).

Also note that the return to the normal operating range is not as abrupt, due to the filter.

5. SETPOINT SOFTENING

5.1. Setpoint Softening Demonstration

Select **Process | Initialize**.

Select **Control | Measurement Options**.

Select **Yes** for "Use substitute value instead of value from process sensor?"

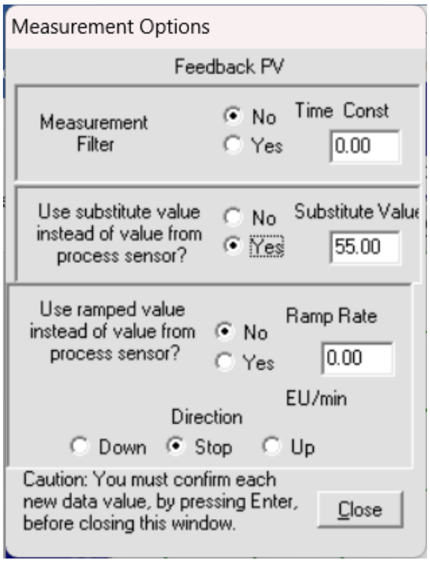


Figure 25: Use substitute value for PV

From **Control | Control Options**. Select **Proportional Only** control algorithm.

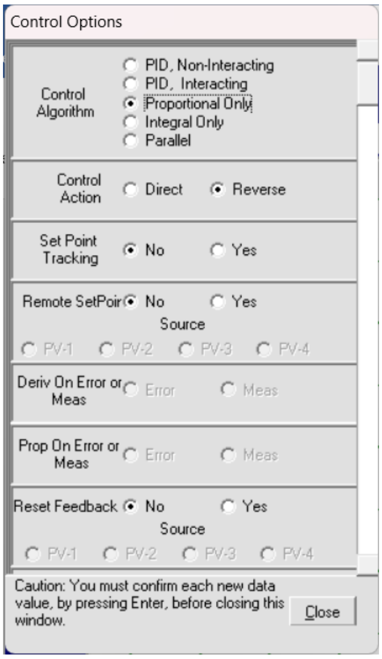


Figure 26: Selecting proportional only controller

In the **Tuning** tab, set: Gain = 1

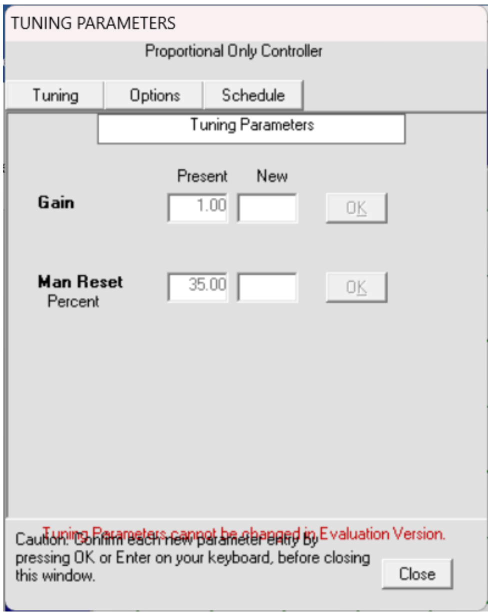


Figure 27: Controller gain = 1

What would be the change in the controller output (proportional kick), if the setpoint is abruptly increased by 10%? **10%**

Put the controller in AUTO. Increase the setpoint to 325 Deg F, enabling the **Set Point Ramp** option (enter a rate of 5 Deg F/min) before making the setpoint change. We use setpoint ramping as the more common setpoint filtering is not supported by this version of simulation software.

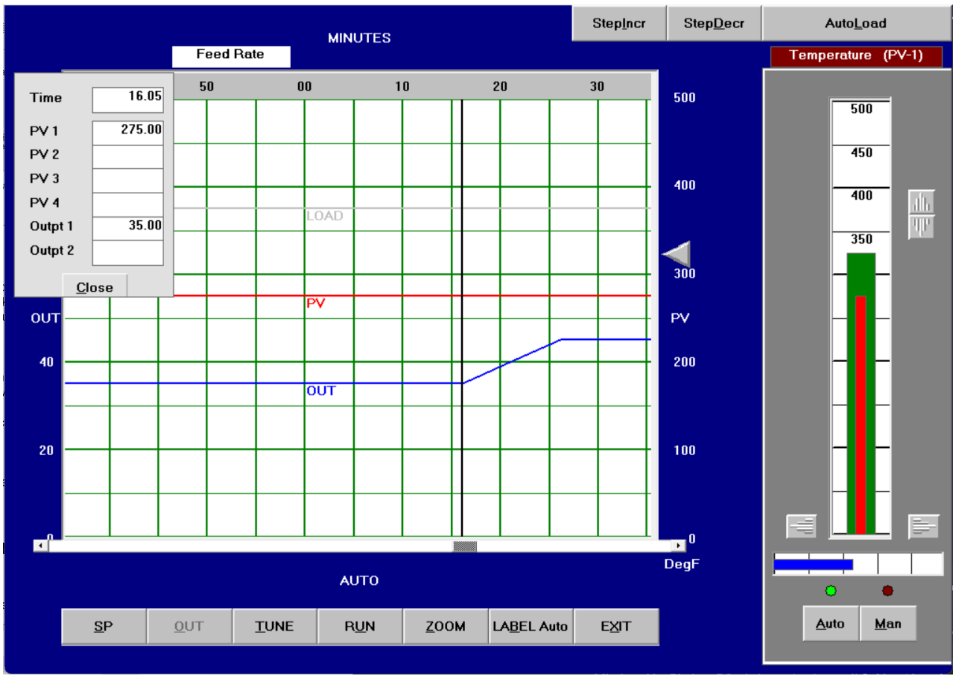


Figure 28: Controller output's initial value

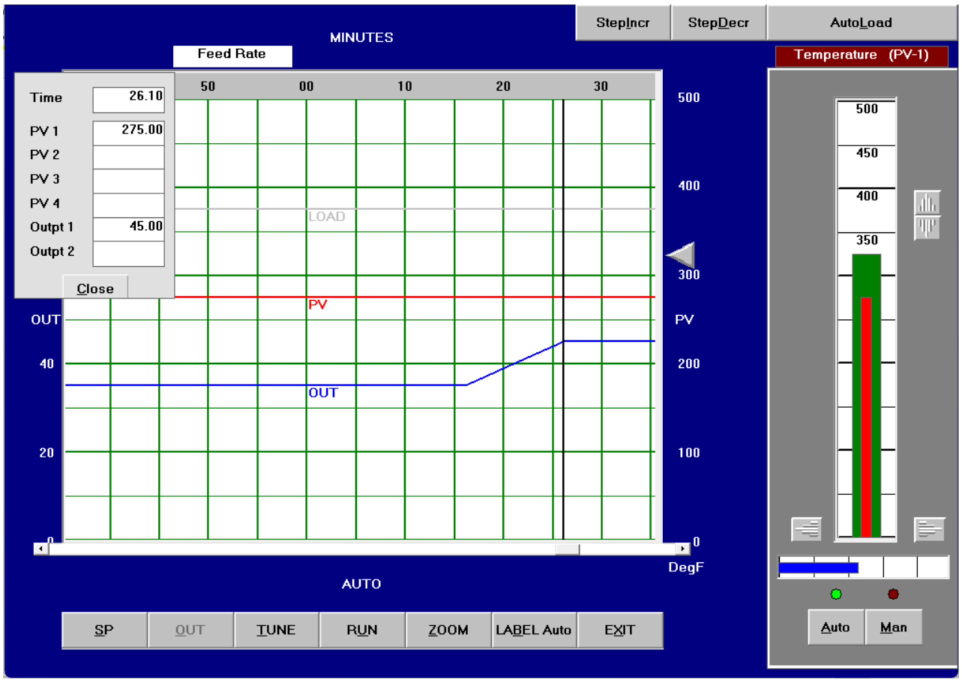


Figure 29: Controller output's final value

Observe: The effective setpoint used by the controller ramps up towards the new value. The controller output follows the setpoint waveform.

How long does it take for the controller to reach its new steady value? **10** min

Does this match your expectations? **Yes**

Note that in most commercial controllers, the effective setpoint changes are transparent to the user and are calculated behind the scenes.

5.2 Setpoint Softening, A Second Example

Select **Process | Initialize**.

Select **Load** and set Manual Load Change Step size to 20%.

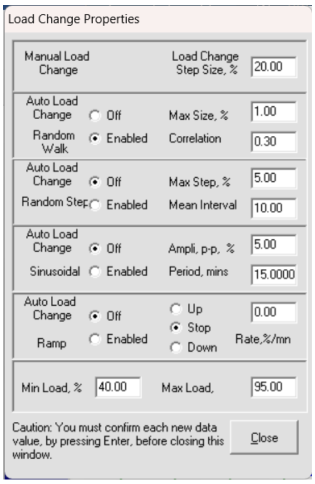


Figure 30: Setting manual load change step size to 20%

Press **ZOOM** and select 100 and 400 respectively.

In the **Tuning** tab, enter the following tuning values:

Gain:	2
Reset, Min/Repeat:	6
Deriv, Min:	0

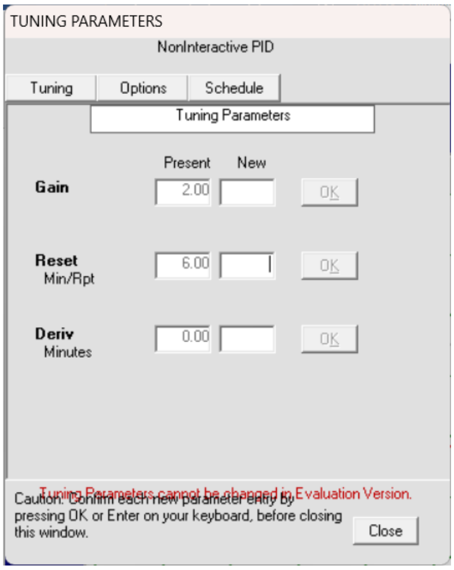


Figure 31: Values of tuning parameters

Put the controller in AUTO. Make a 20% load change. Press **StepDecr** once.

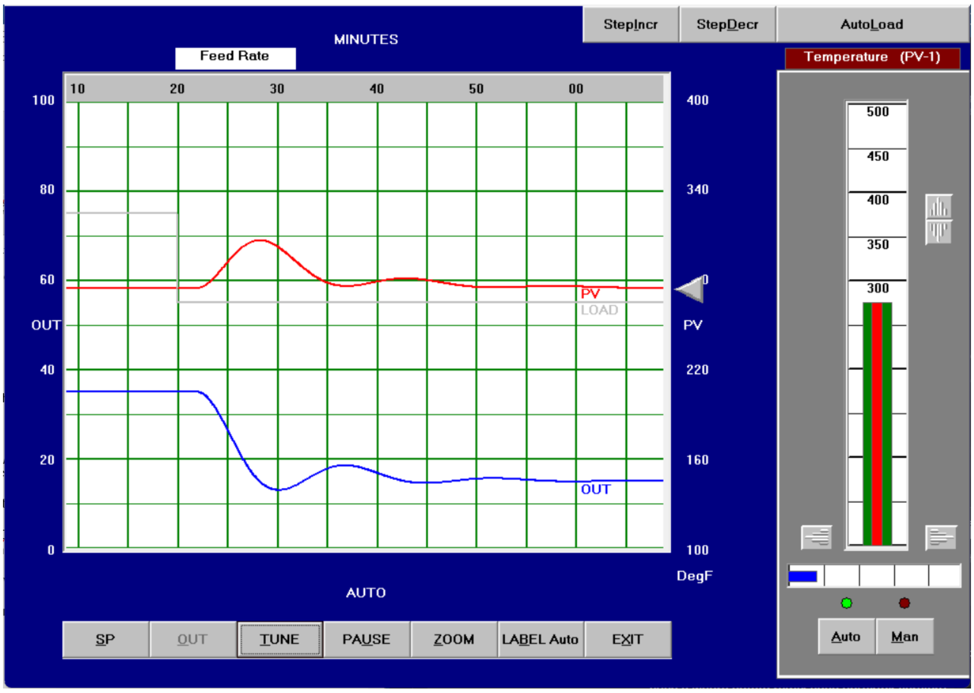


Figure 32: Response after a load change

Observe: A good response to a load upset.

Test the controller by making a setpoint change. Change the setpoint to 325 Deg F.

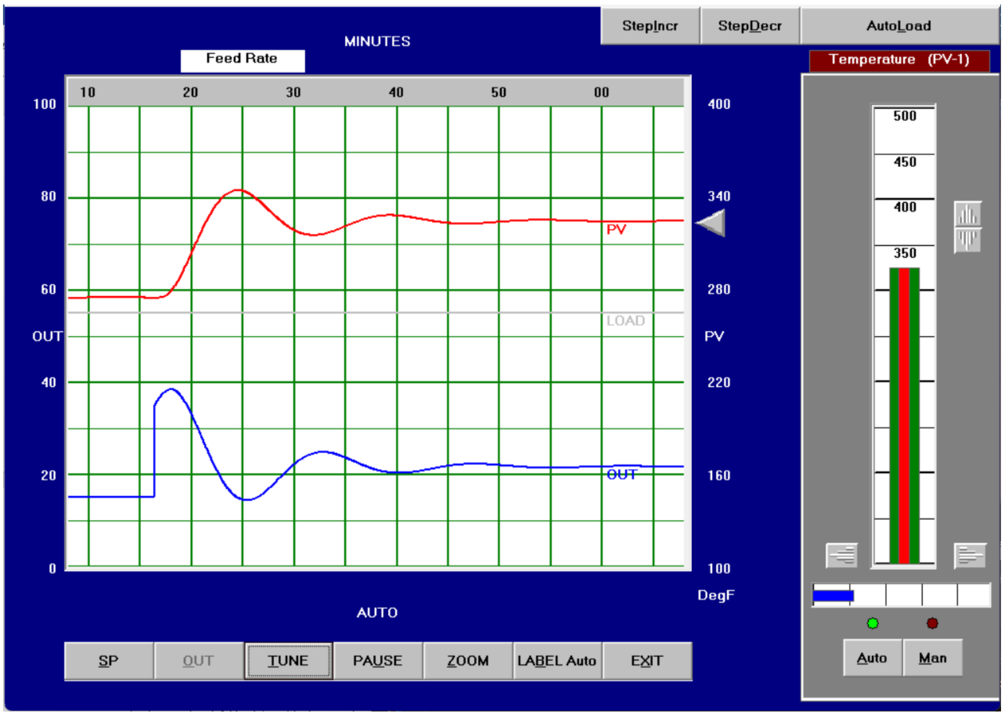


Figure 33: Response after increasing setpoint to 325 Deg F without setpoint softening

Calculate the process variable overshoot: $\frac{345.18-324.98}{324.98-276} * \mathbf{100} = \mathbf{41.24}\%$

(For a damp response, the industry standard is that a well-tuned controller will bring the process variable to the setpoint in at most one cycle and with an overshoot not exceeding 10%)

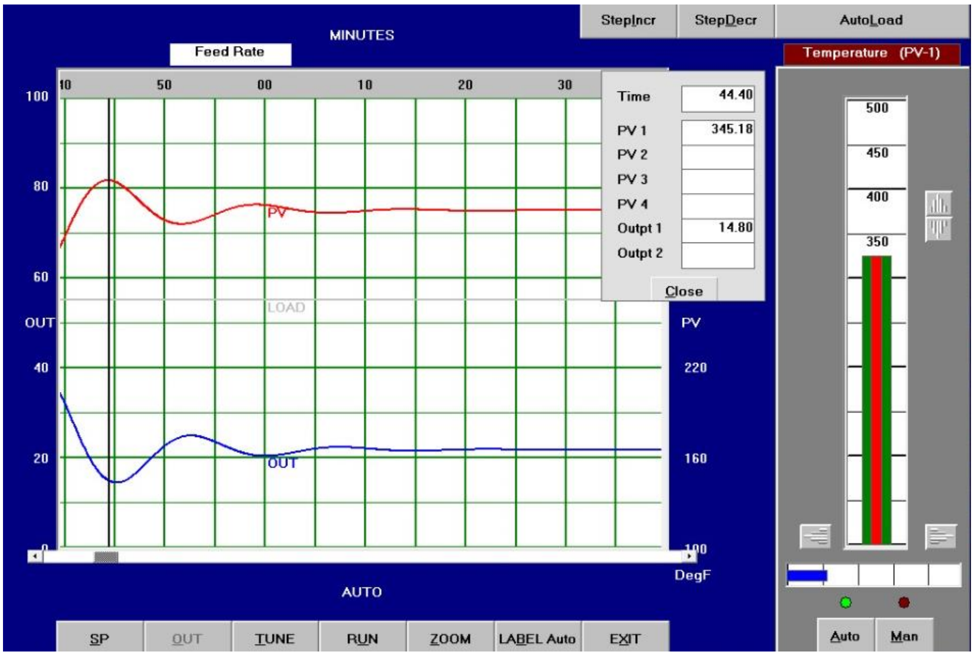


Figure 12: Maximum peak of PV

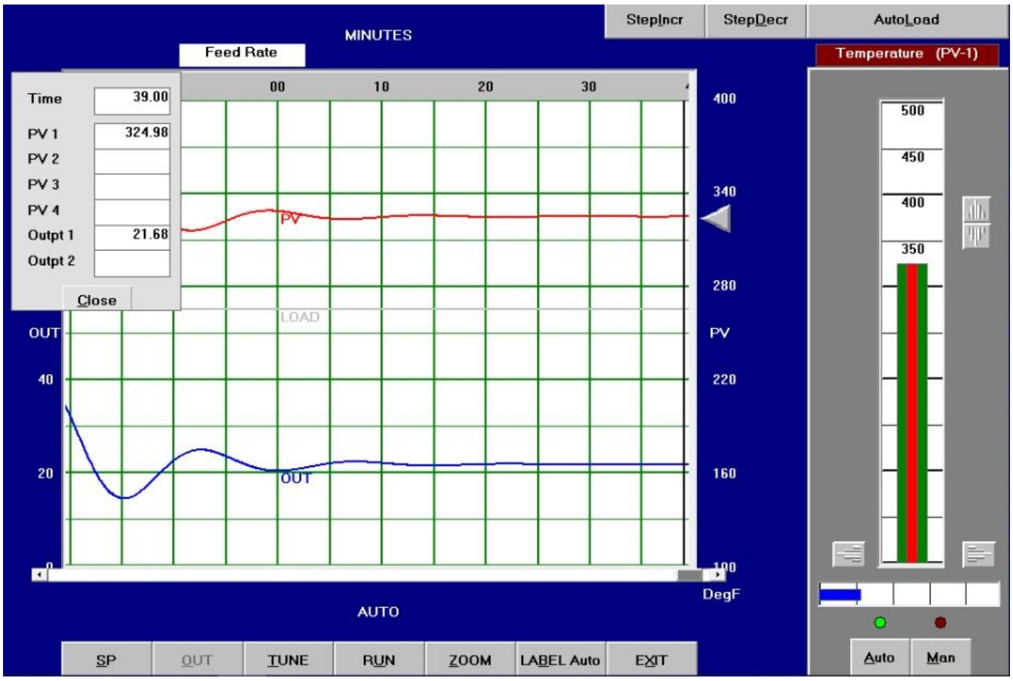


Figure 13: Final value of PV

Bring the setpoint back to 275 Deg F. Wait till the response settles.

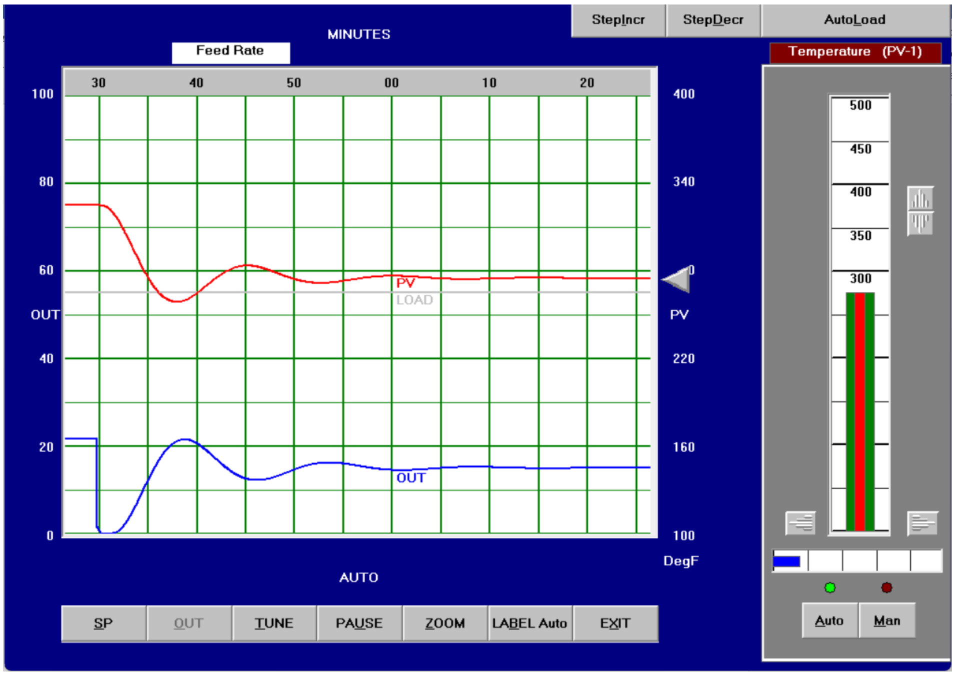


Figure 14: Response after decreasing setpoint to 275 Deg F

Once more, increase the setpoint to 325 Deg F, enabling the **Set Point Ramp** option this time (enter a rate of 4 Deg F/min) before making the setpoint change.

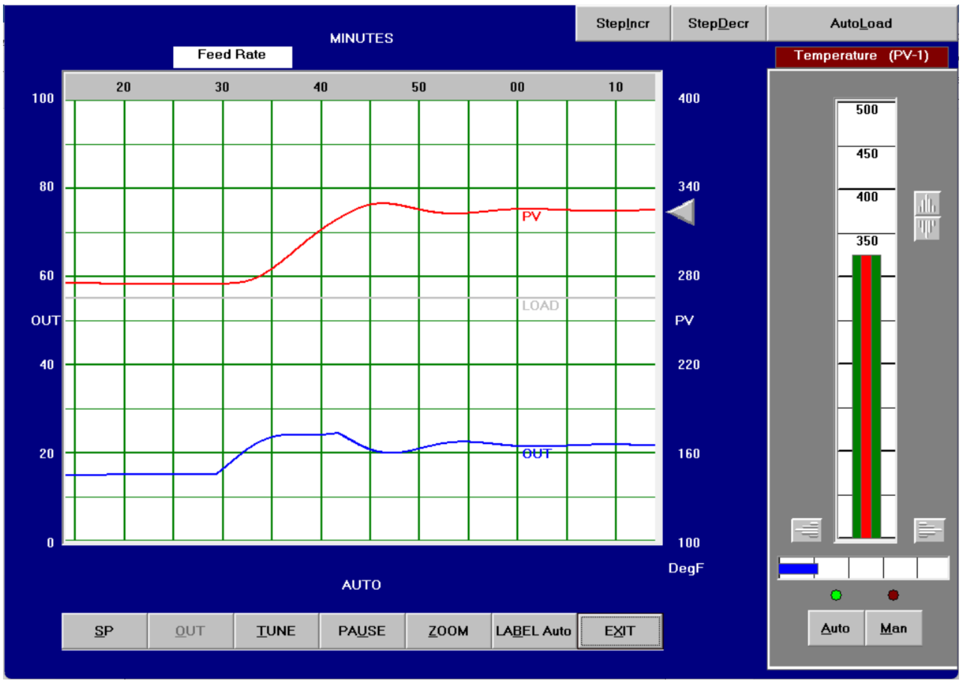


Figure 15: Response after increasing setpoint to 325 Deg F with setpoint softening

Calculate the process variable overshoot with setpoint ramping: $\frac{329.65-324.93}{324.93-276} * 100 = 9.6464\%$

Also note that the settling time is shorter in this case.

Figure 16: New maximum peak of PV

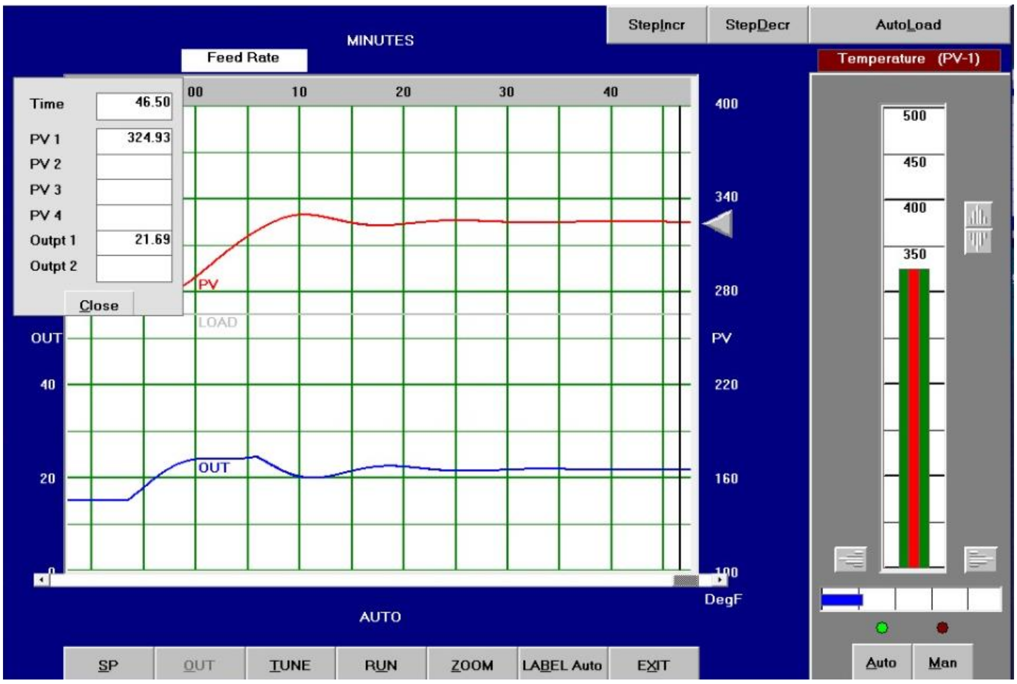


Figure 17: New final value of PV

What you should have observed:

A controller tuning that provides an excellent load upset rejection can produce large overshoot for setpoint changes. With setpoint softening, when a step setpoint change is made towards a new target value, the effective setpoint used by the controller is changed rather gradually (ramped) from its present value to the target value at a specified rate. This will result in a more gradual change in the process variable, considerably reducing overshoot.

Therefore, it is practical to tune the controller for satisfactory load upset response and utilize setpoint softening in order not to sacrifice setpoint tracking. This is called two degrees of freedom controller.

Note that this advantage comes with the complexity of adding one more additional tuning parameter to the algorithm: the setpoint ramp rate, or the setpoint lead-lag ratio for the more common setpoint lead-lag filter.

6. INTERACTING VS NONINTERACTING CONTROLLER

Select **Process | Initialize**.

Select **Load** and set Manual Load Change Step size to 20%.

Press **ZOOM** and select 100 and 400 respectively.

In the **Tuning** tab, enter the following tuning values:

Gain:	2
Reset, Min/Repeat:	5
Derivative, Min:	1

Put the controller in AUTO. Make a 20% load change by pressing **Step Dec** once. Take a screenshot the response and save it to another file. Return the load to its original value.

Calculate the apparent tuning values for the interacting controller.

$$\lambda = \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{T_D}{T_I}} = \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{1}{5}} = \mathbf{0.7236}$$

$$\widehat{K}_C = \lambda K_c = 0.7236 * 2 = \mathbf{1.4472}$$

$$\widehat{T}_I = \lambda * T_I = 0.7236 * 5 = \mathbf{3.618\ min/rpt}$$

$$\widehat{T}_D = \frac{T_D}{\lambda} = \frac{1}{0.7236} = \mathbf{1.3819\ min}$$

Switch the controller to MAN. From **Control | Control Options**. Select **PID, Interacting** control algorithm. In the **Tuning** tab, enter the values calculated above. Put the controller in AUTO.

Make the exact same load change above. Compare with the saved screenshot. The two responses should essentially be the same.

7. BUMPLESS TRANSFER

7.1. With Setpoint Tracking

Select **Process | Initialize**.

From **Control | Control Options**. Select **YES** for Set Point Tracking.

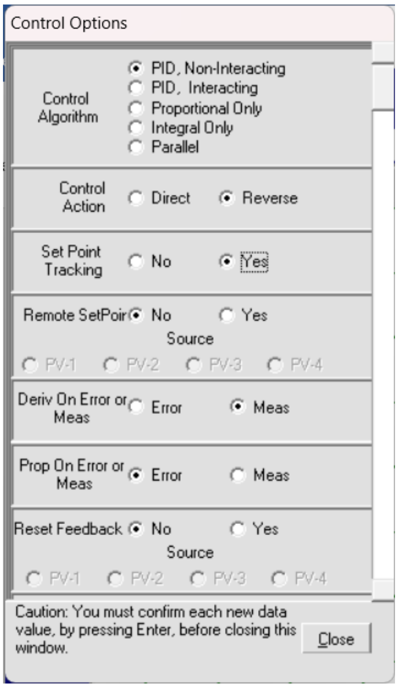


Figure 18: Enable setpoint tracking

With the controller in MAN, go to **Control | Measurement Options**, select **Yes** for "Use substitute value …?", then successively enter the following measurement values: 280, 285, 290, 295, 300. Notice how the setpoint tracks the process variable. Press **Close** to remove the Measurement Option box.

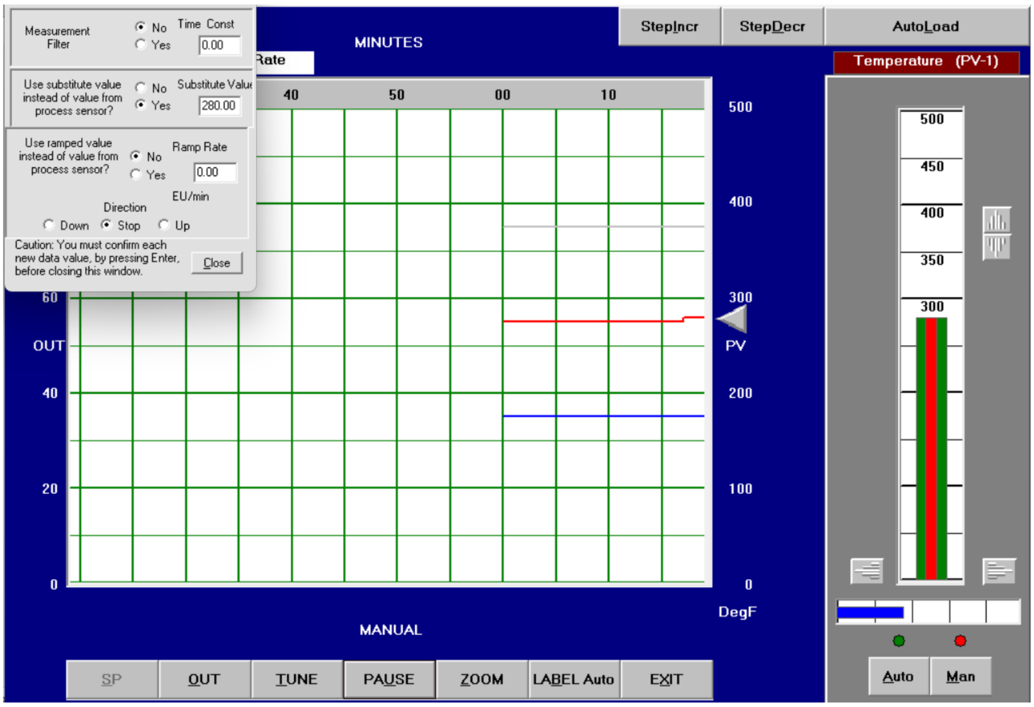


Figure 19: PV = 280 Deg F

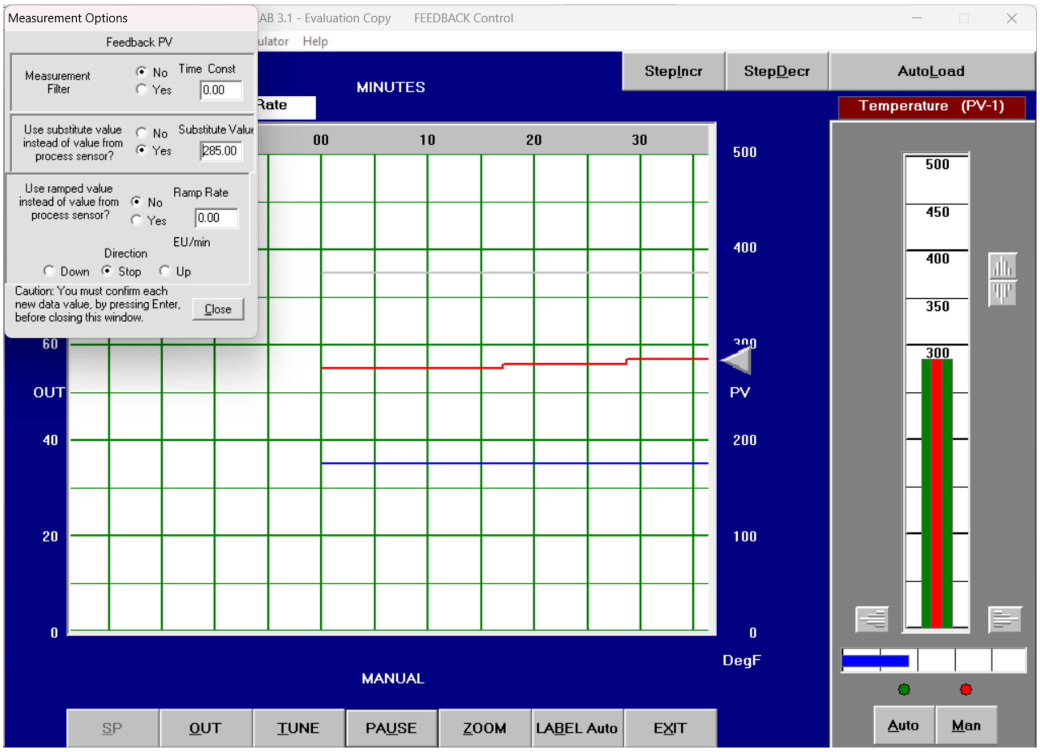


Figure 20: PV = 285 Deg F

Try changing the setpoint via the **SP** button or by moving the arrow to the right of the display.

As the setpoint is now tracking the measurement (as long as the controller is in MAN),

this should not be possible. Switch the controller to AUTO.

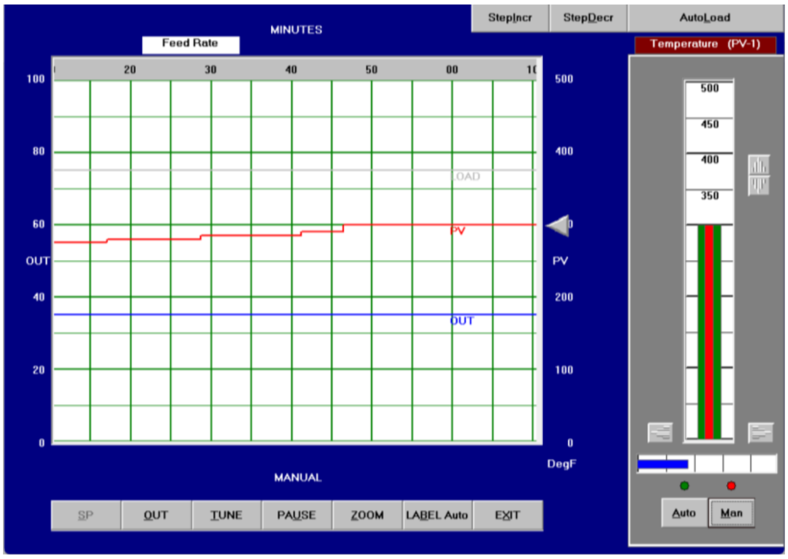


Figure 21: SP cannot be changed in manual mode

Observe: Since –with setpoint tracking– there is always no error when switching from MAN to Auto, The controller's output does not change at all. You can now change the setpoint, and the controller will act accordingly.

7.2. Without Setpoint Tracking

Select **Process | Initialize**.

With the controller in MAN, increase the setpoint to 325. Notice that you can manipulate the set point since setpoint tracking is not enabled. You have just noticed an error of 10%.

Switch the controller to AUTO.

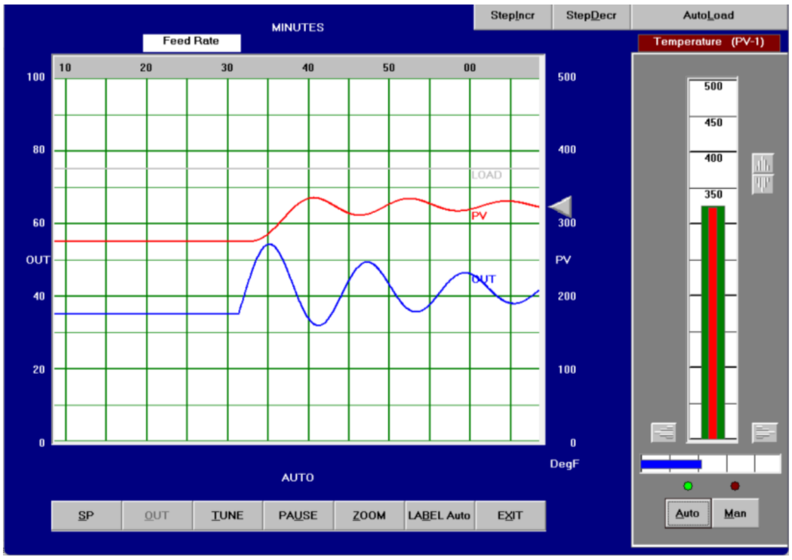


Figure 22: Response after introducing 10% error without setpoint tracking

Does the controller output exhibit a bump due to the proportional kick? ________**No**__________

This demonstrates the AUTO/MAN bump less transfer feature that all commercial controllers now have.