Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292333
Fax: +44 (0)1684 297929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Ph: +1 412.968.9744
Fx: +1 412.968.9746

www.triomotion.com

Doc No.: **TN20-72**
Version: **1.1**
Date: **08th Nov 2004**
Subject: **Axis Tuning**

Application Note

1. Introduction:

There have been many requests for an axis tuning guide that gives more detail than the section in the Motion Coordinator Technical Reference Manual. This is an attempt to give that detail.

2. Preparation:

Before the position loop can be tuned correctly the speed loop in the drive must be set up according to the Drive Manufacturer's instructions. This example and guide assumes that the axis can be moved rapidly over a short distance without damaging any mechanical components of the machine. The true load must be connected to the motor shaft in order to be able to tune the position loop correctly.

A small tune program must be written to step the axis. Here is an example which steps the motor through 1/8 revolution in both directions:

```
' Intialise the axis for P Loop tuning
BASE(0)
UNITS=1
DEFPOS(0)
counts_per_rev=4000
' set speed to 3000 rpm
SPEED=counts_per_rev*3000/60
ACCEL=SPEED*1000
DECEL=ACCEL

' set large FE_LIMIT so it does not trigger during tuning
FE_LIMIT=counts_per_rev/2
SERVO=ON
WDOG=ON

stepsize=INT(counts_per_rev/8)
WHILE TRUE
```

```
    TRIGGER
    WA(20)
    MOVE(stepsize)
    WA(500)
    MOVE(-stepsize)
    WA(480)
WEND
```

If the mechanical configuration only allows movement in one direction then modify the program loop to this:

```
stepsize=INT(counts_per_rev/8)
WHILE TRUE
    TRIGGER
    WA(20)
    MOVE(stepsize)
    WA(480)
    DEFPOS(0)
WEND
```

Before running the program, check that the axis is clear to move and then open the axis parameter and oscilloscope windows in Motion Perfect. Set up the scope to monitor DPOS and MPOS of the axis you are tuning and set the timebase to 20 msec per division. Click the options button and set the points per division value to 10 or 20. Back on the scope main screen, set the trigger source button to "program", shown as 2 sheets of program listing, and set the one-shot / continuous button as required.

3. Tuning P_GAIN:

For safety, start with a low value of P_GAIN. In an analogue system with a typical 1000 or 1024 line encoder, the best starting gain is 0.5. Set all other gains to zero.

Start the tune program and trigger the scope. The response should be overdamped as shown in fig 1.

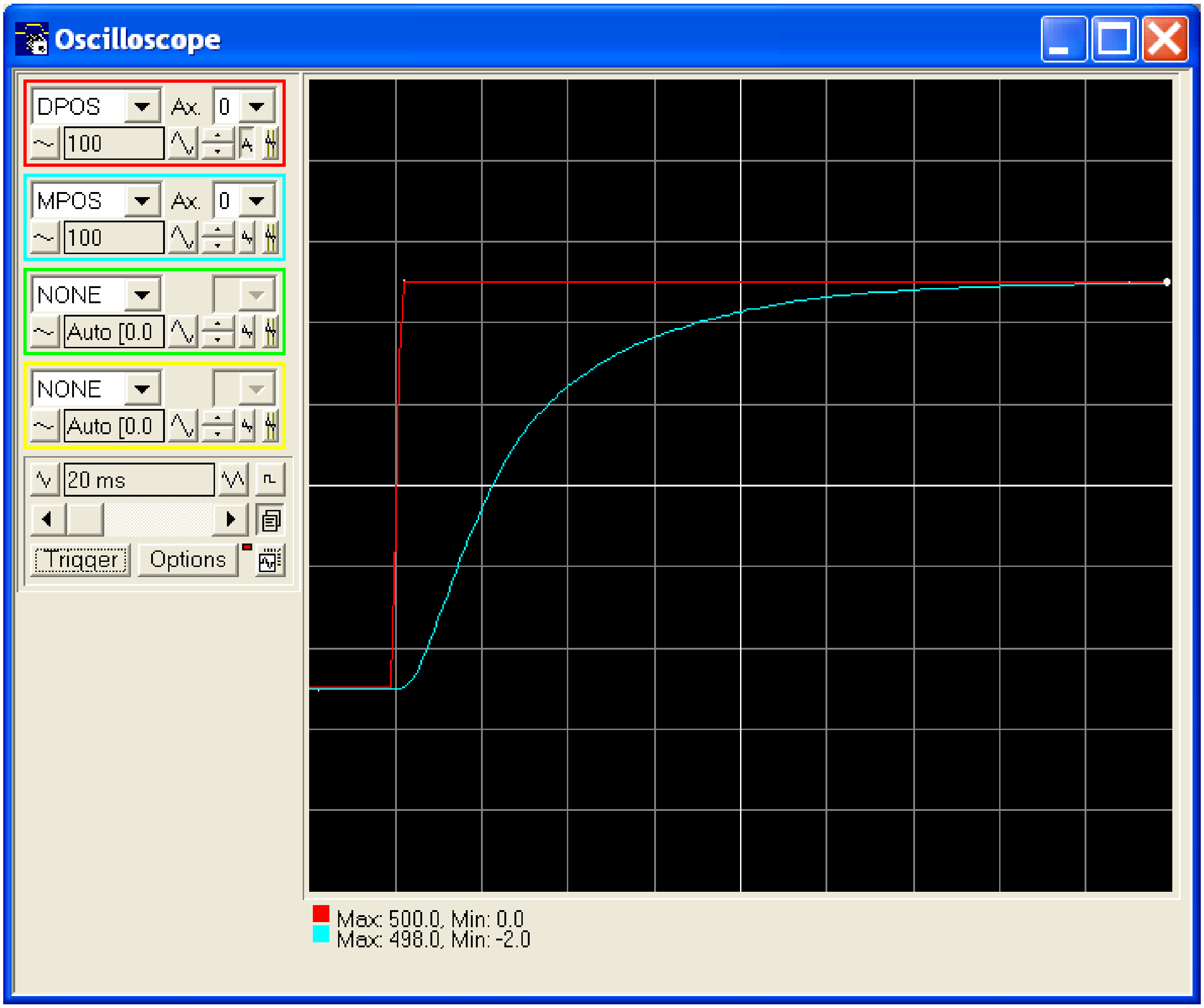

Fig 1. Overdamped response

Now increase the P_GAIN until the response looks like the critically damped trace shown in figure 2.

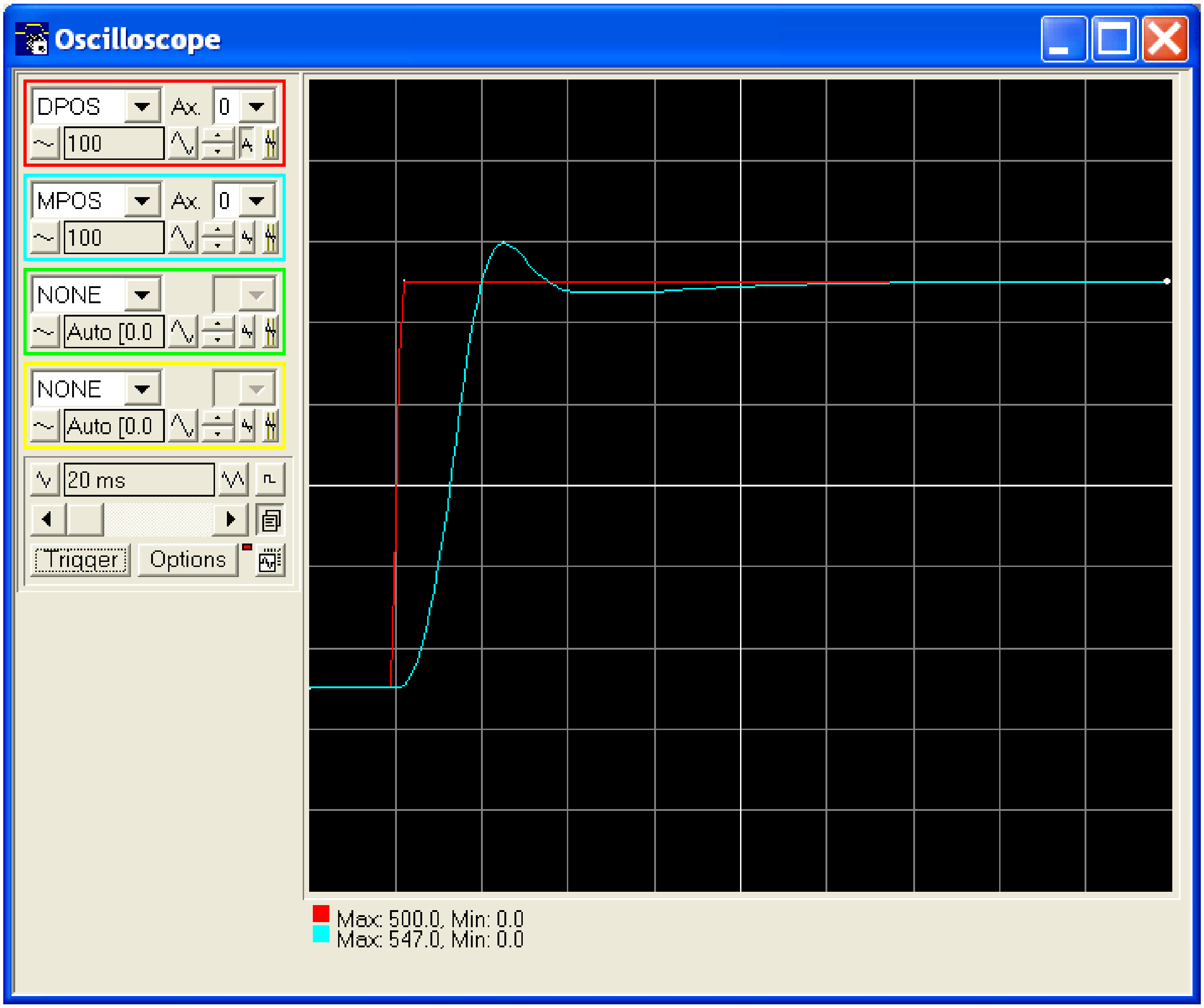

Fig. 2 – Critically Damped Response

If the P_GAIN is increased too much then the axis becomes under damped as shown in figure 3.

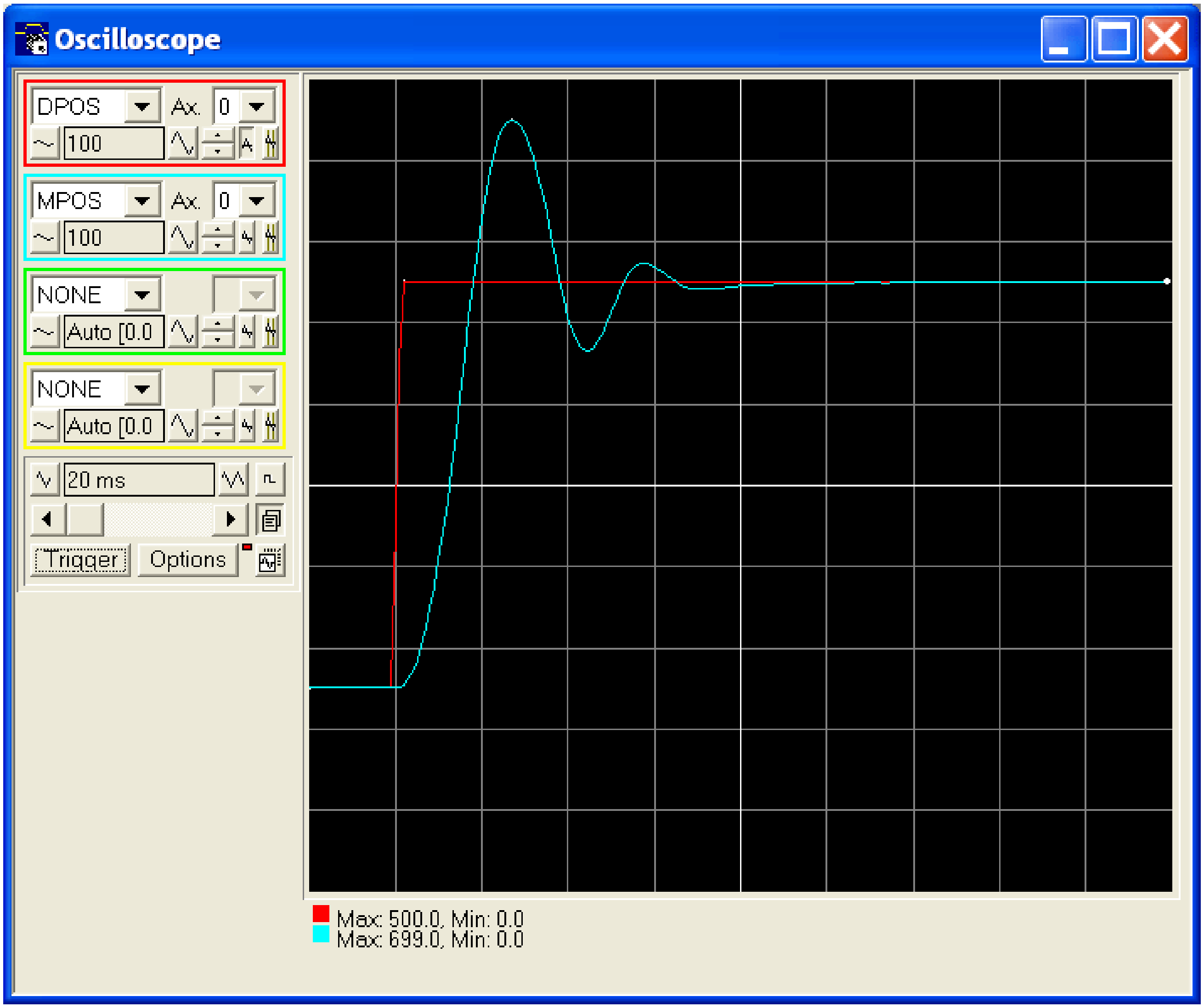

Fig. 3 – Under damped Response

Once the P_GAIN is set up then that could be all that is required. However in most systems it is necessary to reduce the following error to a minimum as much as possible.

There are 2 different strategies for reducing the following error;

1. Use P_GAIN and VFF_GAIN only
2. Use P_GAIN, I_GAIN and D_GAIN only

4. Setting VFF_GAIN:

The simplest strategy is to set the P_GAIN as discussed above and then either calculate the VFF_GAIN or set it by running the axis at a constant speed and increasing the VFF_GAIN until the FE value is at a minimum. The speed used for setting VFF_GAIN is not really important but somewhere between 30% and 60% of max RPM is usual. Once set, the VFF_GAIN value is correct for all speeds.

If calculation of the VFF_GAIN is required, then the formula for a 12 bit analog output (or where DAC_SCALE=16) at 1ms SERVO_PERIOD is this:

VFF_GAIN = (60 / motor_rpm_10V) * (1000 / encoder_pulses_per_rev) * 2048

Or for a system with 16 bit analog output and 1ms SERVO_PERIOD:

VFF_GAIN = (60 / motor_rpm_10V) * (1000 / encoder_pulses_per_rev) * 32768

For a general approach you can use the following:

(60/max_speed) * (1/counts_per_rev) * (10^6/SERVO_PERIOD) * (2^dac_resolution/2).

When using the equation it is important to get the correct relationship between the speed and voltage. If you cannot get the maximum speed from the manufacturers data then it can be found by setting the DAC to half the full range and reading the VP_SPEED and doubling it.

5. Setting I_GAIN and D_GAIN:

Strategy 2 is to set I_GAIN in order to reduce the following error and then to set D_GAIN to reduce (Dampen) the oscillations that will result from the integrator time lag. Using I_GAIN always has the secondary effect of making a large overshoot of final position almost inevitable, unless deceleration is done very carefully.

Like the VFF_GAIN setting, first set up P_GAIN as shown in section 2. Now increase I_GAIN slowly to reduce the final FE without too large an overshoot. Be careful as the I_GAIN values needed are very small. E.g. starting value of

0.001!

Notice that the overshoot becomes very large and the period of the oscillations is slightly longer than with the P_GAIN alone. See figure 4. Notice the vertical scale is double that of the other scope traces.

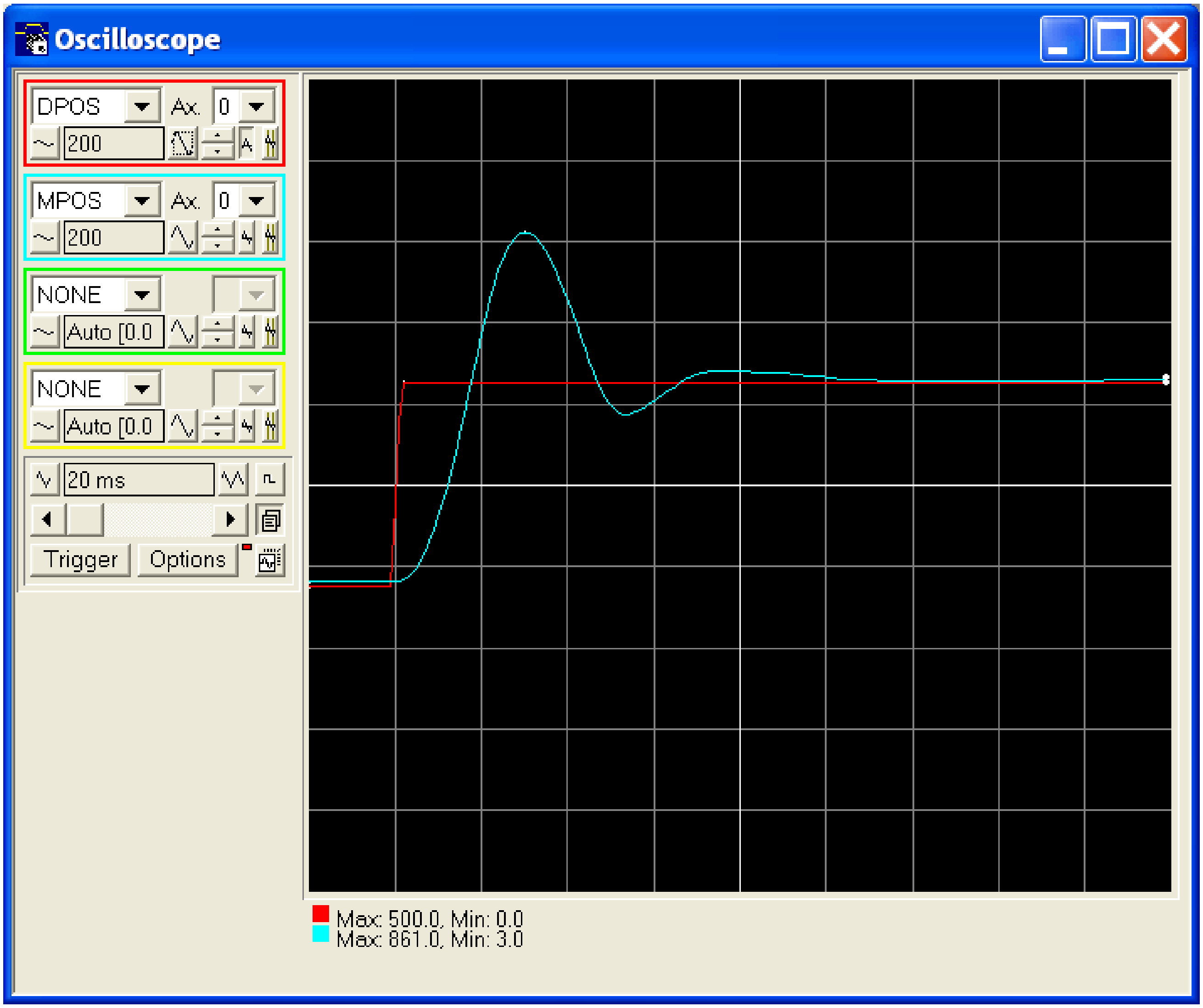

Fig. 4 – I_GAIN Response

D_GAIN can be added to reduce the oscillation. D_GAIN values are normally around 8.0 to 20.0. Figure 5 shows a typical response with P, I and D gain all set. Notice that the overshoot is much reduced and the following undershoot has been removed.

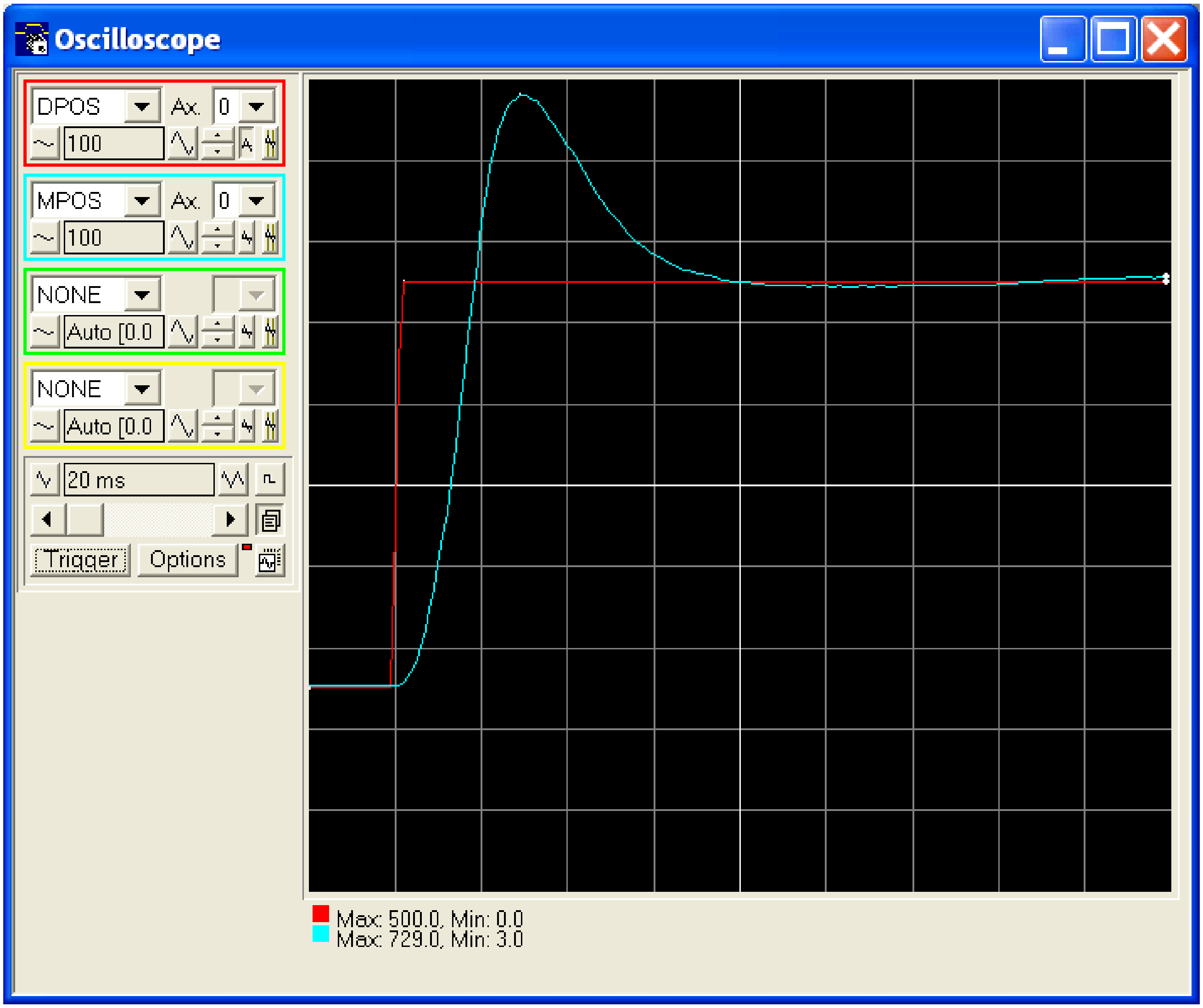

Fig. 5 – P, I and D gains set

OV_GAIN can be used instead of D_GAIN and it has more or less the same damping effect. Note that OV_GAIN values must be negative, but otherwise of the same order of magnitude as the D_GAIN.

APPENDIX – Motion Coordinator Technical Reference Manual

The manual has the following information about setting gains:

Proportional Gain

Description The proportional gain creates an output voltage, Op that is proportional to the following error E.

Op = Kp x E

Axis parameter is called **P_GAIN**
Syntax: P_GAIN=0.8
Note: *All practical systems use proportional gain, many use this gain parameter alone.*

Integral gain

Description The Integral gain creates an output Oi that is proportional to the sum of the errors that have occurred during the system operation.

Oi=Ki x SE

Integral gain can cause overshoot and so is usually used only on systems working at constant speed or with a slow acceleration.

Axis parameter is called **I_GAIN**
Syntax: I_GAIN=0.0125

Derivative gain

Description This produces an output Od that is proportional to the change in the following error and speeds up the response to changes in error whilst maintaining the same relative stability.

Od = Kd x DE

This gain may create a smoother response. High values may lead to oscillation.

Axis parameter is called **D_GAIN**
Syntax: D_GAIN=5

Output Velocity Gain

Description This increases the system damping, creating an output that is proportional to the change in measured position.

Oov = Kov x DPm.

This parameter can be useful for smoothing motions but will generate high following errors. Note that a NEGATIVE OV_GAIN is required for damping.

Axis parameter is called **OV_GAIN**
Syntax: OV_GAIN=-5

Velocity Feed Forward Gain

Description As movement is created by following errors at high speed the following error can be quite appreciable. To overcome this the Velocity Feed Forward creates an output proportional to the change in demand position so creating movement without the need for a following error.

Ov = Kvff x DPd

Axis parameter is called **VFF_GAIN**
Syntax: VFF_GAIN=10

The VFF_GAIN parameter can be set by minimising the following error at a constant machine speed AFTER the other gains have been set.