PMC
Precision MicroControl Corp.

TechNOTE

Title: User Units - Configuring and Programming
Products(s): MultiFlex, DCX-PCI300, DCX-AT300, DCX-AT200
Keywords: User Units, Scaling, Encoder Scaling, RPM, User Rate, MCSetScale
ID#: TN1058
Date: October 28, 2004

Summary

For most applications it is much more convenient for the programmer (and the user) to work in machine units (inches, feet, millimeters) than in encoder units (which are the default units for servo and closed loop stepper applications). Most PMC Corp. motion controllers (except for the DCX-PC100 and the DC2-PC) allow the programmer to define both encoder (or step count for open loop steppers) scaling and rate scaling.

More Information

This Visual Basic program example is based on a rotary servo with a directly coupled 500 line (2000 counts per rotation) incremental encoder. The servo motor shaft is connected to the load via a lead screw. One full rotation of the motor shaft results in a move of 0.2 inches of the load. In order to command motion of the load in units of feet per minute use the MCAPI function MCSetScale( ) to define the scaling for the Scale and Rate members of the MCSCALE data structure:

```
Encoder_Scale = 120000  'set encoder scale to 1 foot (2000 * 5 * 12)
Time_Scale = 60         'set Rate to 60 (seconds * 60)

error = MCGetScale(Controller, 1, Scaling)
Scaling.ScaleFac = EncoderScale
Scaling.Rate = TimeScale
error = MCSetScale(Controller, 1, Scaling)
Call MCEnableAxis(Controller, 1, True)    'required to init user scaling

Motion.cbSize = Len(Motion)
error = MCGetMotionConfigEx(Controller, 1, Motion)
Motion.Velocity = 10    '10 feet per minute
Motion.Acceleration = 300     'Accel to max. velocity in 2 sec.'s (60/2)*10
Motion.Deceleration = 300     'Decel in 2 sec.'s (60/2)*10
error = MCSetMotionConfigEx(Controller, 1, Motion)
Call MCEnableAxis(Controller, 1, True)

Filter.cbSize = Len(Filter)
error = MCGetFilterConfigEx(Controller, 1, Filter)
Filter.FollowingError = Filter.FollowingError / Encoder_Scale
Filter.VelocityGain = (Filter.VelocityGain * Encoder_Scale) / Time_Scale
error = MCSetFilterConfigEx(Controller, 1, Filter)
```

> This example assumes that the PID filter parameters have previously been set using the MCFILTEREX data structure. The only PID filter parameters that are affected by user scaling are Feed Forward (Velocity gain, Accel gain, and Decel gain) and Following Error, which is why these values are redefined.