Desktop Milling Machine CNC Retrofit
Using *PIC-SERVO* Motor Controllers

Overview

This application note describes the motors, electronics and software needed to convert a small 3-axis desktop milling machine to servo-based CNC control. It applies to small milling machines such as those made by Sherline or MAXNC. It also contains some guidance for working with larger machines requiring larger motors.

While many stepper based CNC packages are available, servo control offers faster, smoother and more reliable performance. In addition, the sample CNC control program, PSCNC.EXE, runs under Windows, making the system very clear and easy to use.

Please note that this application note and the sample software and source code are provided for your convenience in evaluating the *PIC-SERVO* motion controllers. They are not supported as products, and all risks associated with their use are with the user. Also, please take the time to read the *PIC-SERVO* chip data sheet and the *PIC-SERVO* motor control board data sheet to understand their operation[1].

This application note does not cover the mechanical aspects of physically attaching the motors to milling machine.

What Is Needed

The following items are needed for your CNC retrofit:

> 3 Motors
> 3 *PIC-SERVO SC* or *CMC* (standard or 3-phase) Motor Control boards
> 1 *Z232-485* Serial Port Converter board
> 3 Motor cables
> 3 NMC Communication cables
> 1 RS232 cable
> 6 Limit switches
> 1 Logic power supply
> 1 Motor power supply
> 1 Enclosure
> PC (200 MHz Pentium or better) running Windows 98/NT/2000/XP
> PSCNC Example Software
> NMCTEST test utility program
> PIC-SERVO, PIC-SERVO CMC data sheets
> PIC-SERVO Motor Control board data sheet

---

[1] Unfortunately, we at **JEFFREY KERR, LLC** do not have the resources to answer all questions regarding the basics of motion control systems. However, this application note, the other supporting documentation, and other basic information available on the Internet should help you understand the fundamentals of CNC and motion control systems.

The cost for all of these items is a little over $1000.00. Of course, you can always save a little money by searching for surplus components.

Motors

The motors recommended for a small desktop milling machine are Pittman 14204S006 24v DC brush motors with 500 line (2000 count) optical encoders. They will deliver 26 oz-in continuously at a little over 3000 RPM. They can be ordered from Pittman Motors (1-215-256-6601).

In general, it is simplest to connect the motor shafts directly to the drive screws, and the rest of this application note assumes that this is the configuration used. However, this gives you a much higher speed capability that you would actually want to use. (1000 RPM, or 50 in/min with 20 TPI screws, is a good practical speed limit for small machines.) To increase the available torque (which in turn, improves the accuracy), it would be better to use a 2:1 or 3:1 toothed belt drive reduction to connect the motors to the drive screws.

The **PIC-SERVO SC** or **CMC** controller boards can drive up to 3 amps continuously, or 6 amps peak, at up to 48vdc. The 3-Phase versions of these boards can drive up to 6 amps continuously, also at up to 48v. Therefore, is possible to drive more powerful motors if they have closer to a 48v winding. The specification to check when looking at motors it the torque constant, which gives the amount of torque per amp.

If you need to use motors which require more than the 1/3 HP max. available from the **PIC-SERVO** controller board's amplifier, you have a couple of options:

1. Use an external amplifier such as those available from Advanced Motion Controls. Make sure you get one which accepts PWM and Direction inputs.
2. Build your own **PIC-SERVO** based controller and high current amplifier. This is only for the truly ambitious.

Motor Cables

You will need to make cables for connecting the motors to the connectors on the **PIC-SERVO or PIC-SERVO 3PH** controllers. The pin definitions for the motor connectors can be found in the **PIC-SERVO** board data sheets. The connections for the Pittman motor and encoder are as follows:

Encoder	Definition
Black wire	Ground
Red wire	+5v
Yellow wire	Channel A
Blue wire	Channel B
Green wire	Index

Motor	*Definition*
Red terminal	Motor +
Black terminal	Motor -

The encoder wires and motor wires should be in separate shielded cables, with the shield wires connected to ground only at the **PIC-SERVO** motor controller end. In the motor wire cable, M+ and M- should be twisted together. In the encoder cable, Channel A should be twisted with GND, and Channel B should be twisted with +5v. The Index wire can be optionally twisted with a spare wire connected to GND. Do not try using the shield wire as a ground wire. The motor/encoder cables should not be longer than 10 feet.

Also in the encoder cable, you will want to include 2 or 4 wires for connecting your limit switches. Only two wires for a single limit switch are mandatory, but a second limit switch is a good option.

The motor wires should be at least 22 gage, and the encoder wires can be 24 or 26 gage. Larger motors should have heavier gage wires.

PIC-SERVO Control Components

The **PIC-SERVO** motor control boards, the **Z232-485** converter, the logic power supply, and the communications cables can be purchased from the **PIC-SERVO** distributor, HdB Electronics (1-800-287-9432) using the following part numbers:

Motor controller boards (any combination of the following 3 boards)

PIC-SERVO SC Motion Control Board	KAE-T0V10-BDV1
PIC-SERVO SC 3PH Motion Control Board	KAE-T0V10-BD3PHV1
PIC-SERVO CMC Motion Control Board	KAE-T0V5-BDV1
PIC-SERVO 3PH Motion Control Board	KAE-T0V5-BD3PHV1
NMC Communication Cables (3)	125250
Z232-485 Serial Port Converter (1)	108939
Logic Power Supply (1)	123964
RS232 cable (1)	106907

The NMC communications cables listed are 20 inch twisted-pair flat ribbon cables. Because you will probably not need such long cables for interconnecting your boards inside the enclosure, you can also make your own shorter 10 pin flat ribbon cables using conventional flat cable.

Limit Switches

Six limit switches, typically pushbutton-type mechanical microswitches should be located (1 each) at either end of the range of motion for each axis. The example software only actually uses the switches at the positive end of the range of motion, and therefore only three are actually required. Limit switch signals should be wired to the "normally closed" and "common" terminals of the switches.

Motor Power Supply

The motor power supply should be 24vdc, capable of supplying 10 amps. The supply may be regulated or unregulated. Make sure that there is an appropriately sized fuse in the AC side of the power supply, and that the power supply case is connected to earth ground.

The DC output of the motor power supply is connected to the screw terminals of each of the **PIC-SERVO CMC** controller boards.

Enclosure

The **PIC-SERVO** controller boards, the **Z232-485** converter board, the logic power supply and the motor power supply should be housed in a sealed enclosure. The enclosure should have no holes (even vent holes) in which metal chips could enter. The enclosure should be metal, ideally aluminum. The enclosure should have a common AC power switch.

When mounting the **PIC-SERVO** controller boards inside the enclosure, the power amplifier chip on each board (LMD18201) should be screwed to a heat sink which makes good thermal contact with the inside of the metal enclosure.

PC Running Windows

A PC running Windows 95/98 is needed as the host for the CNC control system. The PC should have at least a 166 MHz Pentium processor with at least 32 meg. of memory and one free COM port (COM1, COM2 COM3 or COM4). Note that COM1 and COM3, or COM2 and COM4 cannot be used simultaneously. No other programs should be running while the milling machine is actually in motion, although utilities such as text editors and the like can be used while the machine is not in motion.

Software and Documentation

You will need to download the following files from *www.jrkerr.com*:

>   **PIC-SERVO** board data sheet
>   **PIC-SERVO** chip (or chipset) data sheet
>   **PIC-SERVO CNC** example program (PSCNC.ZIP)
>   **PIC-SERVO** test utility program (NMCTEST.ZIP)

PSCNC.ZIP contains the PIC-SERVO CNC Milling Machine controller program, plus documentation and source code files. When you un-zip PSCNC.ZIP, all files should be located in a single working directory.

NMCTEST.ZIP is a test utility program useful for verifying that all of your hardware is working. It can also be used for testing different servo control parameters.

The controller chip data sheet contain background information on the **PIC-SERVO** controllers, and are necessary for understanding how your **PIC-SERVO** controllers work.

Assembling and Testing the Hardware

Before assembling the controllers inside your enclosure, you should connect everything up (using **PIC-SERVO** board data sheet as a guide) on your bench. The motors should be disconnected from your milling machine. It will be convenient to plug in both the logic power supply and the motor power supply to a power strip with a common power switch.

Run the program NMCTEST.EXE, and make sure that all three controllers are recognized by the software. Try some test motions using the default gains. If encoder seems to be working, but the motor jerks and does not servo properly, the polarity of the motor might be reversed. In this case, try swapping the M+ and M- motor wires.

When you are confident that all of your motors are connected and working properly, try connecting your motors to your milling machine. Examine the servo gain parameters located in the sample initialization file PSCNC.INI (using a text editor), and enter them into the NMCTEST.EXE program. Again, try some short test motions to verify that these gains will work with your machine. Be careful not to run the machine into its stops.

If you are using different motors than recommended, or if your machine is appreciably different from a Sherline mill, you may need to adjust your servo gain parameters. Please refer to the **PIC-SERVO** data sheet for guidelines on setting these parameters.

Once you are confident that all of your wiring is correct and that the motors work with your machine, you can assemble the controllers into your enclosure. Once assembled, re-run the NMCTEST.EXE program to verify that everything is still working.

Configuring and Running PSCNC

Before running the PSCNC controller program, you should thoroughly read the documentation file PSCNC.PDF (which is included in PSCNC.ZIP). Make sure that you have at least the positive limit switch connected for each axis.

The sample initialization file PSCNC.INI contains parameters nominally suitable for a Sherline milling machine. Some of these parameters, however, may be different depending on how you hook up your motors, and on the location of your limit switches. In particular, the sign of the X, Y and Z scale factors (+ or -) might be reversed, and the range of motion parameters (xmin, xmax, etc.) may be different.

To test these parameters, you should run the PSCNC.EXE program, but do not home the milling machine when prompted. Instead, enable all servos, and use the jog controls to move the mill around, and verify the directions of motion and the ranges of motion.

After making any corrections in PSCNC.INI that may be necessary, run PSCNC.EXE again, and this time, choose the homing option. The homing procedure will first home the Z axis, then the X axis, then the Y axis. When homing starts, try pressing the limit switches manually before each axis arrives at the limit to make sure that the switches are responding correctly.

When you are confident that the switches are working correctly, re-run PSCNC.EXE and allow the homing to execute properly. The machine should end up in the final position of X=-0.3500, Y=-0.3500, Z=-0.3500. Again, use the jogging controls to move each axis to the negative end of the range of motion, and verify that xmin, ymin and zmin are set correctly. You should now be ready to use your CNC milling machine.

Safety

Computer controlled machinery, and especially machining equipment, can be very dangerous. The PSCNC example CNC control program provided is exactly that - an example. While it has been designed to be quite usable, it is not necessarily failsafe, and users should use the utmost of caution when operating their milling machine. This is especially critical when you are first familiarizing yourself with its operation, and especially if you undertake modifying any of the source code.

Aside from the caution used when operating any piece of home-built electronic equipment, you should also follow the customary safety precautions of wearing safety glasses, not wearing loose clothing or jewelry, keeping your hands away from moving components, and observing electrical safety rules regarding AC power.