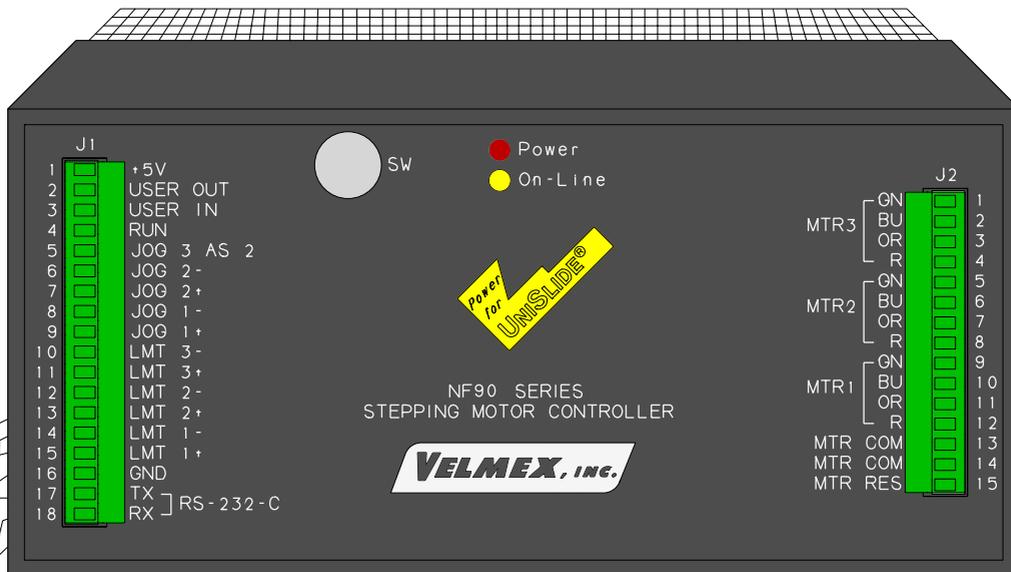


NF90 SERIES

User's Guide

One, Two, and Three-axis Stepping Motor Controller/drivers



CONTENTS

Introduction	3
Features	3
Setup	5
JOG/slew Mode	6
Digitizing	7
On-Line Mode	7
Command Summary	8
Example Programs	9
Interactive Mode	14
Stand-alone Mode	15
Daisy-chaining NF90 Controllers	16
Reference	18
Index Motor Command	18
Speed of Motor Command	19
Acceleration/deceleration Command	20
Loop Commands	20
Pause Command	21
User Commands	21
Backlash Compensation Command	23
Indicate Over-travel Command	23
Quit On-Line Mode Command	24
Run Command	24
Null Position Registers Command	24
Kill Operation Command	24
Verify Controller's Status Command	24
Clear Memory Command	24
Decelerate to a Stop Command	25
Enable On-Line Mode Command	25
Enable On-Line Mode with Echo oFF Command	25
Go after Waiting or Holding Command	25
Put Controller on Hold Command	25
<Delete> Command	25
Request Motor Position Commands	26
Commands for Daisy-Chained Controllers	26
Troubleshooting Procedure	27
Specifications	29
Appendix A	
Switch settings and RS-232-C Connections	30
Appendix B	
Limit Switch and Joystick Connections	31
Appendix C	
USER OUT, USER IN, and RUN Connections	32
Appendix D	
Motor Connections	33
Appendix E	
Motor Performance	34

Introduction

The NF90 is a programmable stepping motor controller for running up to three motors, one-at-a-time. The Controller incorporates a single chip "Super Microprocessor" that has on-chip RAM. The RAM is available for temporary storage of a user-entered program and motion parameters.

Commands and data are entered through the RS-232-C interface from a host computer, terminal, or programmable controller. Specialized Commands provide simple and efficient entry of a complex, yet compact, program.

Features

- ✓ Completely wired and tested for direct connection to UniSlide motors/ assemblies.
- ✓ A complete microprocessor-based Controller with motor drives for three motors.
- ✓ 400 steps per revolution (0.9° step angle) resolution
- ✓ Output current is factory matched for a specific motor (0.7 to 4.7 amp/phase motor).
- ✓ An all metal enclosure and Linear type motor and logic power supplies result in low RFI and EMI.
- ✓ 95-130 VAC, 50/60 Hz operation. 190-260 VAC, 50/60 Hz operation with European fuses is available.
- ✓ JOG/slew mode allows motors to be jogged one step or slewed up to 1000 steps/sec. or 2000 steps/sec.(switch settable). An optional joystick is available.
- ✓ A Digitizing function can be utilized with a host terminal connected as a readout of motor position.
- ✓ A three wire serial port, conforming to EIA standard RS-232-C, allows a host to enter Commands (ASCII characters) and Data, Poll for status, and Read Position information.
- ✓ The NF90 will run in an interactive or stand-alone mode.
- ✓ Acceleration/Deceleration settable from 2,000 to 100,000 steps/sec² in 2,000 step/sec² increments.
- ✓ Speed programmable from 1 to 6000 steps/sec in 1 step/sec increments. NOTE: Most motors have low torque above 2000 steps/sec. (see speed/torque curves in Appendix E)
- ✓ Incremental Index distance is programmable from ±1 to ±1,048,575 steps.

- ✓ Programmable Return-to-Zero position.
- ✓ Six powerful Loop Commands provide from one to continuous repeat operations, performing simple functions like auto-reverse to raster scans and other complex X,Y matrix patterns.
- ✓ Programmable pauses from 100 milliseconds to 13 minutes.
- ✓ A User Output can be programmed to turn On and Off an external solid state relay, or interface to other logic level devices.
- ✓ A User Input can be utilized in a program as a Wait for external switch or relay closure.
- ✓ Backlash Compensation can be set to automatically finish every index approaching from the positive direction.
- ✓ Run, Limit switch, Joystick, Output, Input, RS-232-C, and Motor connections are accessible at unpluggable connectors on the front panel.
- ✓ RS-232-C baud rate settings are switch settable to 300, 1200, 4800, or 9600.
- ✓ As many as 255 controllers can be "daisy-chained" together allowing the host to address each one from just one serial port.
- ✓ Limit Switches for CW and CCW directions are provided with plug-in connection to UniSlide limit switch assemblies. Limits can be used for "homing."
- ✓ Automatic Power Down reduces power consumption by de-energizing the motors when at a standstill.
- ✓ Single Step mode is provided for debugging a program or as a controlled interrupt.
- ✓ The NF90 can be polled for its status at any time; additionally a prompt ("^") is automatically sent to the host when a program has finished.
- ✓ The NF90 can be programmed to send a pulse or character at preset distances without stopping or slowing the motor.
- ✓ Motor position can be read while motor is in motion (up to 500 steps/sec.)
- ✓ The NF90 can be set to signal the host when a limit switch has been encountered.
- ✓ Terminal, Diagnostic, and Example programs for MS-DOS PCs on a diskette are included.

Setup

- * * CAUTION * *
- * HAZARDOUS VOLTAGE, DO NOT REMOVE CONTROLLER'S COVER
 - * DO NOT CONNECT OR DISCONNECT MOTOR(S) WHEN POWER IS "ON"
 - * HIGH TEMPERATURE, NF90 SHOULD BE KEPT AT LEAST 6 INCHES FROM ANY OBJECTS
 - * AIR MUST CIRCULATE AROUND THE CONTROLLER
 - * NEVER USE IN AN EXPLOSIVE ENVIRONMENT
 - * IN INDUSTRIAL ENVIRONMENTS, THE NF90 MUST BE PROTECTED TO PREVENT METAL CHIPS FROM GETTING INTO SMALL OPENINGS

NF90 controllers are factory wired with 8 foot cables for motor(s) and limit switch(es). An optional joystick or RS-232-C cable will be factory wired to the NF90.

CAUTION: Before connecting motor(s), compare the motor's label current and the resistor ohms value from the label on the back of the NF90 to the following table:

NF90 Resistor (100 watt wirewound, located in screened channel on back of NF90, see Appendix D for electrical connection)						
	2Ω	3Ω	4Ω	5Ω	7Ω	10Ω
Motor (Amps)	2.9-4.7	2.8-3.8	1.9-2.9	1.5-1.9	1.2-1.4	0.7-1.1

DANGER: The motor current/resistor combination must be within the limits of the above table. A mismatch of motors to resistor value can result in severe damage to controller or motors.

Refer to Appendix A and B to determine proper RS-232-C and joystick connections to the NF90.

1. Connect cables to motors and limit switches. Connect a cable between your computer's or terminal's serial port and the NF90.

IMPORTANT: Limit Switch inputs require a **closed** circuit for the motor(s) to operate, ie, for motors to operate, limit switch cables must be connected to the UniSlide limit assemblies or equivalent (see Appendix B).

2. The NF90 is factory set to 9600 baud, 7 data, 2 stop bits, and even parity. To change baud rate refer to Appendix A.

CAUTION: Never connect or disconnect motors with power on; this may result in severe damage to motor drives.

3. Plug the NF90 into a 120VAC outlet.

4. Turn on the NF90 by pushing the top of the rocker switch located on the back panel.

The Power light will come on and the On-Line light will flash on then off once.

The NF90 is now ready to receive commands or be jogged/slewed from the optional joystick or user inputs to the JOG inputs (see Appendix B).

The simplest method to send commands (ASCII characters) to the NF90 is with a terminal or computer operating with a terminal program (such as NFTERM included on the Utility Disk). Make sure your computer or terminal is set at 7 data, 2 stop bits and even parity, and the same baud rate as the NF90 (see Appendix A to change baud rate on the NF90).

The other method to send commands is with commercially available languages such as BASIC, C, PASCAL, FORTRAN, or ASSEMBLY.

JOG/slew Mode:

When the On-Line light is not lit the NF90 is in the JOG/slew mode.

With the NF90 in the JOG/slew mode, the motors can be jogged one step, or slewed to 1000 (factory setting) or 2000 steps/sec determined by the setting of SW-3 (see Appendix A).

To JOG motor 1 positive (CW), press the joystick 1+ button momentarily; the motor will move one step CW. To slew the motor maintain pressure on the 1+ button; the motor will accelerate to speed until the button is released.

To JOG motor 1 negative (CCW), press the joystick 1- button momentarily; the motor will move one step CCW. To slew the motor maintain pressure on the 1- button; the motor will accelerate to speed until the button is released.

To JOG motor 2 positive (CW), press the joystick 2+ button momentarily; the motor will move one step CW. To slew the motor maintain pressure on the 2+ button; the motor will accelerate to speed until the button is released.

To JOG motor 2 negative (CCW), press the joystick 2- button momentarily; the motor will move one step CCW. To slew the motor maintain pressure on the 2- button; the motor will accelerate to speed until the button is released.

To jog or slew motor 3, hold the shift* button down and actuate the joystick motor 2 buttons.

Digitizing:

The NF90 stores its absolute position (relative to the position when power was applied or when registers were zeroed) in memory until the NF90 is turned-off. The absolute registers reflect the accumulated distance from operating the motors in the JOG/slew mode and/or under program control. These registers can hold from -8388608 to +8388607 steps.

With a host terminal or computer connected via the RS-232-C interface, the NF90 can be used as a digitizer. In the JOG/slew mode the NF90 will send motor position when it receives a "D" from the host.

Here is an example of what the host would receive when Motor 1 is at absolute 201, Motor 2 is at absolute -1294010, and Motor 3 is at 0:

```
<lf>X+0000201<cr>  
<lf>Y-1294010<cr>  
<lf>Z+0000000<cr>
```

<lf> is a linefeed, <cr> is a carriage return.

The host can null ("zero") the registers by sending a "N" to the Controller.

There is a program called "NFDISPLY" on the Utility Disk for automatically displaying position in actual distance units for various lead screws and rotary tables.

On-Line Mode:

The NF90 can be programmed when it is in the On-Line mode.

To put the NF90 in the On-Line mode the host must send a "E" or "F". When the controller receives an "E" or "F" the On-line light will light.

The "E" puts the NF90 on-line with echo "on" (echoes all characters received back to the host). The "F" puts the NF90 on-line with echo "off".

If you are using a terminal to communicate to the NF90 use the "E" so that the characters the controller echoes will be displayed on the terminal screen.

* This button is designated by:



Command Summary:

(See Reference Section for a more complete description)

The following must end with a carriage return (Enter key) or comma:

InMx Set steps to Index motor CW (positive), m = motor# (1,2,3), x =1 to 1048575

InM-x Set steps to Index motor CCW (negative), m = motor# (1,2,3), x =1 to 1048575

InM0 Index motor to absolute zero position, m =motor# (1,2,3)

SmMx Set Speed of motor, m = motor# (1,2,3), x =1 to 6000 steps/sec.

AnMx Acceleration/deceleration, m = motor# (1,2,3), x =1 to 50

L0 Loop continually from the beginning

L-0 Sets the Loop-to-marker at the current location in the program

Lx Loop from beginning or Loop-to-marker $x-1$ times ($x=2$ to 255)

L-x Loop from beginning or Loop-to-marker $x-1$ times, alternating direction of motor 1

LM-2 Loop once from beginning or Loop-to-marker reversing index direction of motor 2

LM-3 Loop once from beginning or Loop-to-marker reversing index direction of motor 1 and motor 2

Px Pause x tenths of a second and output if output enabled ($x=0$ to 8191, 10 μ sec pause when $x=0$)

U0 Wait for a "high" on the user input

U1 Wait for a high on the user input, holding the user output high while waiting

U2 Disable user output when pausing

U3 Enable output when pausing (reset state)

U4 User output "low"

U5 User output high

U6 Send "W" to host and wait for a "G" to continue

U7 Start of Continuous Index with pulse output

U8 Start of Continuous Index sending "@" to the host

U9 End of Continuous Index

Bx Backlash Compensation, compensation on when $x=1$, off when $x=0$

Ox Indicate Limit Switch Over-travel to host, off when $x=0$, NF90 sends "0" when $x=1$ and a limit switch is encountered

The following commands do not need the carriage return or comma:

Q Quit On-Line mode (return to Jog/Slew mode)

R Run program

N Null (zero) motors 1,2,3 Absolute Position Registers

K Kill operation in progress

V Verify Controller's status; when On-Line, NF90 sends "B" to host if busy, "R" if ready; in Jog/slew Mode NF90 sends "J"

C Clear program from memory

D Decelerate to a stop (interrupts current index in progress)

E Enable On-Line mode with echo on

F Enable On-Line mode with echo **oFF**

G Go after waiting or holding

H Put Controller on Hold (single step mode)

X Send position of motor 1 to host

Y Send position of motor 2 to host

Z Send position of motor 3 to host

The following are for NF90s that are daisy-chained together:

[x] Send commands to the next NF90 in the "chain", x are any of the above commands

& Enable multiple NF90s that are daisy-chained

Example Programs: (These Examples are included on the Utility Disk)

The following examples require commands (ASCII characters) be sent to the NF90 with a terminal or computer operating with a terminal program. Make sure your computer or terminal is set at 7 data, 2 stop bits and even parity, and the same baud rate (baud rate is factory set to 9600) as the NF90 (see Appendix A to change baud rate on the NF90).

The other method to send commands is with commercially available languages such as BASIC, C, PASCAL, FORTRAN, or ASSEMBLY.

Note: The "<cr>" is a carriage return character (<Enter> key on most keyboards). Command characters are in **LARGE BOLD**.

<u>Example #1</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
On-Line	-	-	Enable On-Line mode with echo on (On-Line light lit)

E

<u>Example #2</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Index	1	3	Index Motor 1 400 steps (1 rev) CW

or

I1M400,R

I1M400<cr>

R

→

Graphic Representation: start +-----+ end

<u>Example #3</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Clear	-	-	Clear previous program from memory

C

<u>Example #4</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Index	1	3	Index Motor 1 600 steps CCW

I1M-600,R

←
end +-----+ start

<u>Example #5</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Auto-Reverse	1	6	Auto-Reverse (motor 2)

I2M600,I2M0,R

→
start/end +-----+
: :
+-----+
←

<u>Example #6</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Repeating Index	1	9	Repeating Index in two directions, pausing 1 sec. between Indexes

P10,I1M400,L10,L-2,R

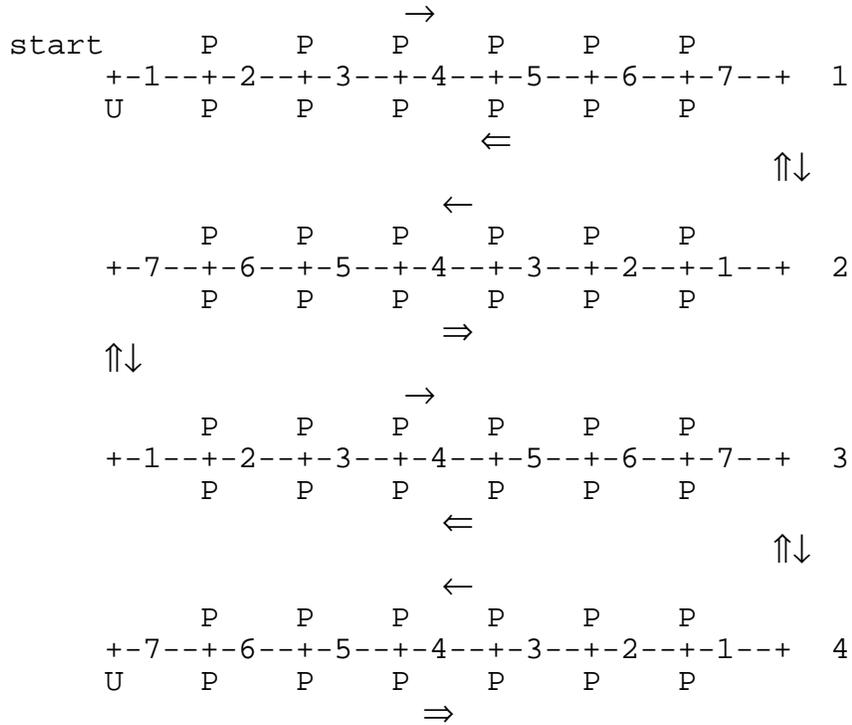
→

	P	P	P	P	P	P	P	P	P	P
start/end	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
	1	2	3	4	5	6	7	8	9	10
	:									:
	P	P	P	P	P	P	P	P	P	P
	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
	10	9	8	7	6	5	4	3	2	1

←

<u>Example #7</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Raster Scan	2	18	Raster scan with 1 sec. pauses and waiting for input at beginning and the end, then run backwards through raster scan

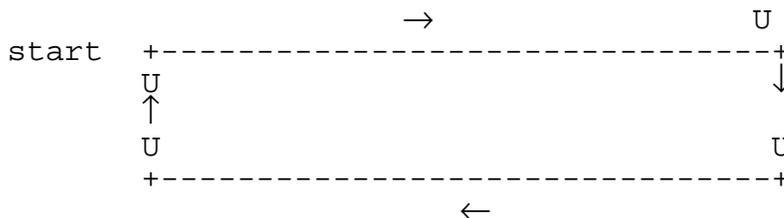
I1M200 ,P10 ,L7 ,I2M400 ,L-4 ,U0 ,LM-2 ,U0 ,L0 ,R



Key: → =forward path
 ⇒ =backwards path

<u>Example #8</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Rectangle	2	12	Rectangle, with Output and Wait at each corner

I1M2000 ,U1 ,I2M1000 ,U1 ,LM-3 ,L0<cr>



<u>Example #9</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
Home to Limit	1	9	Home Motor 1 to Positive Limit Switch and move 200 steps from switch and zero position

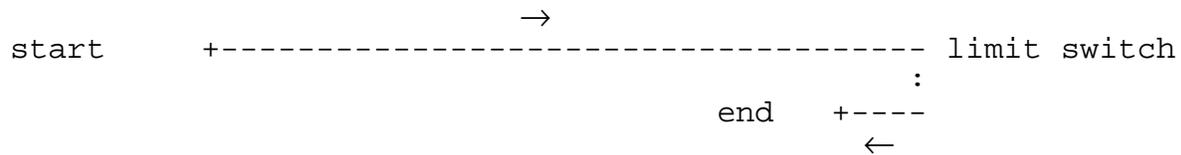
S1M600,I1M1000000,I1M-200,R<cr>

The host must wait for the above commands to finish (wait for "^" from NF90) and then zero position registers and clear the above commands from memory by executing the following:

NC

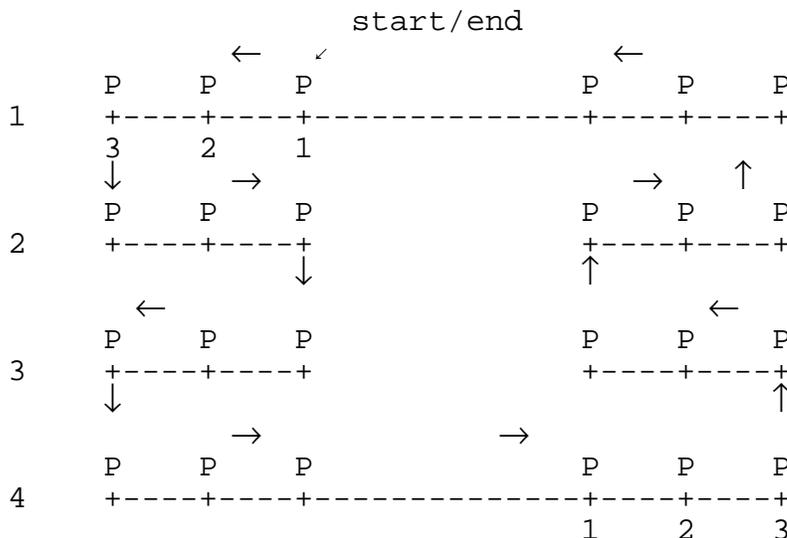
The NF90 is now zeroed 200 steps from the positive limit switch and ready for a program.

CAUTION: Motor speed should not be set above 1000 steps/sec. when homing to a limit switch.



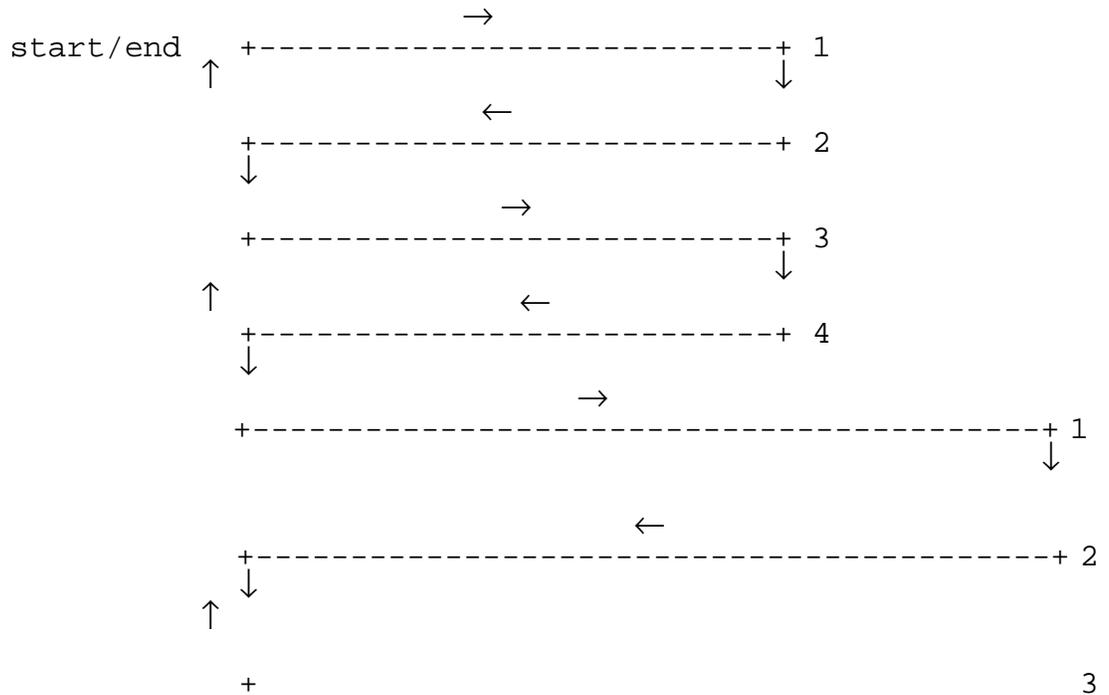
<u>Example #10</u>	<u>Motors run</u>	<u>RAM usage</u> (bytes)	<u>Function</u>
X,Y Matrix	2	17	Mirror-image X,Y Matrix

P3,I1M-400,L3,I2M400,L-4,I1M1600,LM-3<cr>



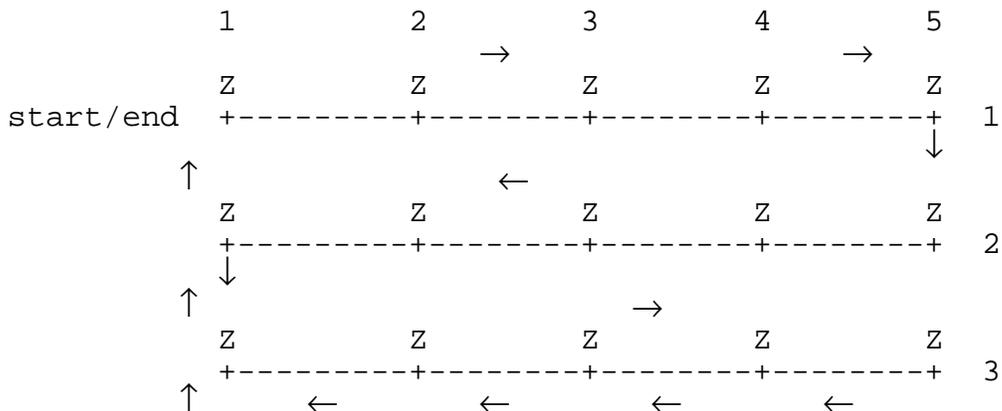
<u>Example #11</u>	<u>Motors run</u>	<u>RAM usage</u> <u>(bytes)</u>	<u>Function</u>
Two Raster Scans	2	19	Two Different Raster Scans using Loop-to-marker

I1M2000,I2M300,L-4,L-0,I2M600,I1M3000,L-3,I2M0<cr>



<u>Example #12</u>	<u>Motors run</u>	<u>RAM usage</u> <u>(bytes)</u>	<u>Function</u>
X,Y Matrix	3	22	X,Y Matrix Moving Z Axis Up then Down at each Position

I3M2000,I3M-2000,I1M1600,L5,I2M400,L-3,I1M0,I2M0<cr>



Interactive Mode

The NF90 can be controlled in an interactive mode. The following procedure would be used for running the NF90 in an interactive mode:

1. The host puts the NF90 On-Line by sending an "F"
2. The host sends a "N" to zero position registers if necessary
3. The host sends speed, and acceleration if necessary
4. The host sends an Index
5. The host sends a "R" to start the Index
6. The host then will wait until it receives a ready prompt ("^") from the NF90
7. The user's routine for outputting, measuring, etc. would be executed by the host
8. A "C" would be sent from the host to clear the previous Index command from the NF90's memory
9. The process is repeated from step # 3

Below is an interactive example written in BASIC: (This program is included on the Utility Disk under the name "NFEXINT")

```
95 REM Open RS-232 (COM1:), 9600 Baud, control lines disabled,ASCII
100 OPEN "COM1:9600,E,7,2,CS0,DS0" FOR RANDOM AS #1
105 REM Enable with echo off, Zero position registers
110 PRINT #1, "FN"
115 REM Clear any existing program, 1000 Steps/Sec, Index 1500steps
120 PRINT #1, "C S1M1000,I1M1500,R"
130 GOSUB 500
135 REM Clear existing program, Index 900 steps
140 PRINT #1, "C I1M900,R"
150 GOSUB 500
155 REM Clear existing program, Index 1000 steps
160 PRINT #1, "C I1M1000,R"
170 GOSUB 500
175 REM Clear existing program, 2000 Steps/Sec, Index to zero
180 PRINT #1, "C S1M2000,I1M0,R"
190 GOSUB 500
390 PRINT "DONE RUNNING NF90"
392 REM leave NF90 off-line
395 PRINT #1, "Q"
400 END
499 REM Wait until ready ("^") prompt appears in receive buffer
500 C$ = INPUT$(1, #1)
510 IF C$ <> "^" THEN 500
515 REM Request motor position from NF90 and print on your computer
520 PRINT #1, "X"
530 INPUT #1, P
540 PRINT "MOTOR POSITION="; P
550 REM Your routine for end of Index would go here
600 RETURN
```

Stand-alone Mode

The NF90 can be programmed in a stand-alone mode. In a stand-alone mode the host downloads all the commands necessary and the operator starts the program with the RUN (J1,4) input on the J1 connector (see Appendix C for a proper remote RUN input).

Below is a BASIC example that sends example #6 as a stand-alone program: (This program is included on the Utility Disk under the name "NFEXSTD")

```
95 REM Open RS-232 (COM1:), 9600 Baud, control lines disabled,ASCII
100 OPEN "COM1:9600,E,7,2,CS0,DS0" FOR RANDOM AS #1
105 REM Enable with echo off, Zero position registers,clear any pgm
110 PRINT #1, "FNC P10,I1M400,L10,L-2,"
999 END
```

Another way of sending programs from an IBM PC would be to first write them as text files with an editor such as DOS Edlin or Wordstar nondocument, and use the DOS commands MODE and COPY to send them to the NF90.

If the following program was stored under the file name of "EXAMPLE":

```
FNC P10,I1M400,L10,L-2,
```

Then the following DOS commands would send the program at 9600 baud to a NF90 connected to COM1 serial port of the PC:

```
MODE COM1:9600,E,7,2
COPY EXAMPLE COM1
```

Daisy-chaining NF90 Controllers

Users that require more than three motors or simultaneous motion can daisy-chain NF90 Controllers together. Multiple Controllers can be operated from a single RS-232 port by daisy-chaining. When daisy-chaining multiple NF90s together, addressing is accomplished by enclosing information in brackets. Information between brackets is relayed to succeeding Controllers. The number of brackets used determines the destination of the information. As each Controller receives the information, it removes a set of brackets and relays the remainder of the instruction to the next Controller. Up to 255 Controllers can be daisy-chained together and programmed in this manner. For example, if four Controllers are daisy-chained together and the fourth Control is to be programmed, the instruction for that Controller would be placed within three brackets. The first Controller receiving the information would remove one set of brackets and relay the information along, the second Controller would remove the second set of brackets and the third Controller would remove the third set and relay the instructions to the fourth without any brackets. The targeted fourth Controller would then be programmed with the information.

EXAMPLES:

This character will put all the NF90s On-Line.

&

This will take two NF90s linked together off-line.

[Q]Q

This will program Controller #4 to Index Motor 1 800 Steps.

[[[C,I1M800,R]]]

This polls Controller #2 to see if it is busy.

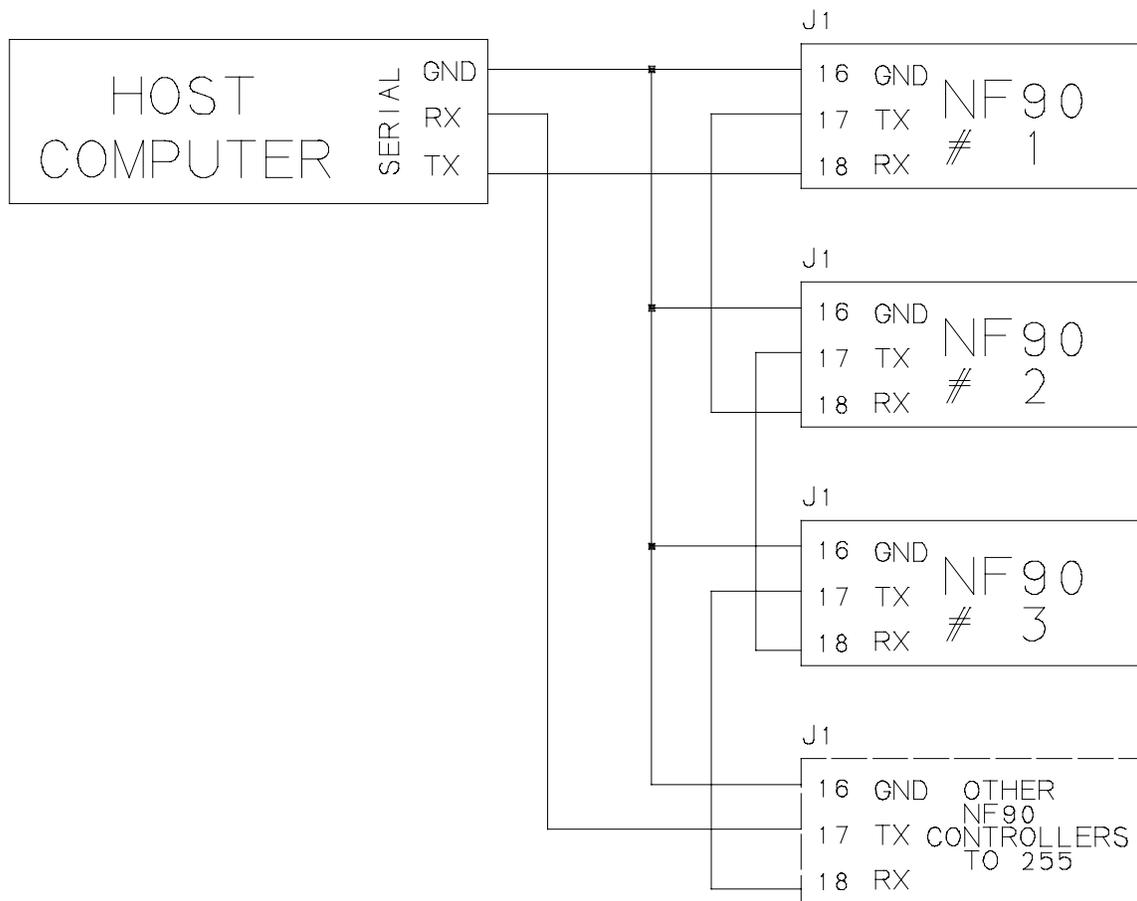
[V]

This requests position of Motor 1 of Controller #3.

[[X]]

The procedure for daisy-chaining Controllers:

1. Connect the RS-232 from the host to the NF90 Controllers as shown.



2. Set RS-232 parameters on the host and Controllers the same.
3. Initialize Controllers by sending the following character (**CAUTION:** do not use the "E" or "F" command to put the NF90s On-Line):

&

The "&" commands all the Controllers On-Line with echo off (the host will receive a "!" from the last NF90).

4. Controller #1 is now ready to receive commands, all other controllers will be in a relay mode. Use brackets to address a different NF90 in the chain.

Reference

This section gives detailed explanations of the NF90's commands

The following commands must end with a carriage return (Enter key or Return on most keyboards) or a comma. Most of the commands use the NF90's (RAM) memory. The required memory needed per command is specified in the following descriptions. The NF90 has 101 bytes of RAM available for commands, which is reestablished when the "C" (Clear) command is used. The NF90 will ignore any incoming commands and send a "^" if it's memory is totally full.

I m M x Set steps to Index (Move) motor CW* (positive, UniSlide Slider will move away from motor end, Rotary Table will rotate CCW*), m = motor# (1,2,3), x =1 to 1048575.
Memory usage = **3 bytes**.

Examples: Note: The "<cr>" is a carriage return character (<Enter> key on most keyboards). Command characters are in **LARGE BOLD**.

This example sets motor 1 to index 1200 steps CW:

I1M1200<cr>

This example sets motor 2 to index 9200 steps CW:

I2M9200<cr>

This example sets motor 3 to index 10200 steps CW:

I3M10200<cr>

To calculate the number of steps for indexing, divide the distance desired to move by the Adv/step. Example: To move 1 inch with the W4 lead screw ($1 \div 0.001 = 1000$) requires a 1000 step index.

UniSlide Lead Screw	Adv/rev	Adv/step	Speed at 1000 steps/sec
P40,C	0.025"	0.0000625"	0.0625 ips
P20,B	0.050"	0.000125"	0.1250 ips
P10,W1	0.100"	0.00025"	0.2500 ips
P5,W2	0.200"	0.0005"	0.5000 ips
P2.5,W4	0.400"	0.001"	1.0000 ips
WF	1.000"	0.0025"	2.5000 ips
K1,Q1	1.00 mm	0.0025 mm	2.50 mm/sec
K2,Q2	2.00 mm	0.005 mm	5.00 mm/sec
K4,Q4	4.00 mm	0.01 mm	10.00 mm/sec

UniSlide Rotary Table	Gear Ratio	Adv/step	Speed at 1000 steps/sec
B4872TS	72:1	0.0125°	12.5°/sec
B4836TS	36:1	0.025°	25.0°/sec
B4818TS	18:1	0.050°	50.0°/sec
	180:1	0.005°	5.0°/sec
5900-90	90:1	0.010°	10.0°/sec
5900-45	45:1	0.020°	20.0°/sec

* To reverse this direction see Appendix D for wiring changes

ImM-x Set steps to Index (Move) motor CCW* (negative, UniSlide Slider will move toward motor end , Rotary Table will rotate CW*), *m*= motor# (1,2,3), *x*=1 to 1048575.
Memory usage = **3 bytes**.

Examples:

This example sets motor 1 to index 120 steps CCW:

I1M-120<cr>

This example sets motor 2 to index 20 steps CCW:

I2M-20<cr>

This example sets motor 3 to index 1 step CCW:

I3M-1<cr>

ImM0 Index motor to absolute zero position, *m*=motor# (1,2,3). When this command is used the NF90 calculates the distance and direction to get back to absolute zero position. The "absolute zero" position was established when the NF90 was turned on or when the "N" (Null Absolute Position Registers) command is used.
Memory usage = **3 bytes**.

Examples:

This example sets motor 1 to index to absolute zero position:

I1M0<cr>

This example sets motor 2 to index to absolute zero position:

I2M0<cr>

This example sets motor 3 to index to absolute zero position:

I3M0<cr>

SmMx Set Speed of motor, *m*= motor# (1,2,3), *x*=1 to 6000 steps/sec. If this command is not used the default speed will be 1000 steps/sec. When the speed is set in a program followed by a "C" (Clear) command, the speed will default to this last set speed. **NOTE:** motor torque decreases as speed increases, and most motors have limited torque above 2000 steps/sec, see Appendix E for speed/torque curves. Stepping motors produce vibration at certain speeds, if noise or vibration at a chosen speed is objectionable, try a speed slightly higher or lower.
Memory usage = **3 bytes**.

Example:

This example sets the speed of motor 1 to 500 steps/sec:

S1M500<cr>

* To reverse this direction see Appendix D for wiring changes

A_mM_x Acceleration/deceleration, *m*= motor# (1,2,3), *x*=1 to 50. A value of 1 is 2000 steps/sec², 2 is 4000 steps/sec², 50 is 100000 steps/sec², etc. If this command is not used the default will be 1 (2000 step/sec²). The higher the number used, the faster the motor will reach the set speed, and the faster it will slow down to a stop. **CAUTION:** motors may stall if this value is set to high.
Memory usage = **0 bytes** (this command is immediate, and uses a reserved memory location)

Example:

This example sets the acceleration/deceleration of motor 1 to 6000 steps/sec²:

A1M3<cr>

L0 Loop continually from the beginning. This command can be used once in a program as the last command, it functions the same as a "continuous run input".
Memory usage = **2 bytes**.

L-0 Sets the Loop-to-marker at the current location in the program. All looping commands, except "L0", that follow this command will branch to here. Loop commands prior to this one will branch to the beginning of the program. **NOTE:** This command can be used only once in a program.
The maximum number of loop commands a program can hold is 10 before and 10 after the loop-to-marker.
Memory usage = **0 bytes** (this command is immediate, and uses a reserved memory location)

L_x Loop from beginning or Loop-to-marker *x*-1 times (*x*=2 to 255). Loop commands can be nested for looping more than 255-1 times.
Memory usage = **2 bytes**.

The following example loops equal (100x100x10)-1 = 99,999 times:

L100,L100,L10<cr>

NOTE: When the Loop reaches its last count, the non-loop command directly preceding the Loop will be ignored.

L-x Loop from beginning or Loop-to-marker x-1 times, alternating direction of motor 1 indexes (x=2 to 255). Loop commands can be nested for looping more than 255-1 times. Memory usage = **2 bytes**.

The following example loops equal $(100 \times 5) - 1 = 499$ times:

L-100,L5<cr>

NOTE: When the Loop reaches its last count, the non-loop command directly preceding the Loop will be ignored.

LM-2 Loop once from beginning or Loop-to-marker reversing index direction of motor 2. See Example Program section for use of this command. Memory usage = **2 bytes**.

LM-3 Loop once from beginning or Loop-to-marker reversing index direction of motor 1 and motor 2. See Example Program section for use of this command. Memory usage = **2 bytes**.

Px Pause x tenths of a second and output if output enabled (x=0 to 8191, 10 μ sec pause when x=0). The USER OUT (J1,2) will go to +5V (if "pause control" is enabled) for the duration of the pause. The default is output enabled when pausing (see U2 and U3 commands for "pause control"). Memory usage = **2 bytes**.

Example:

This example pauses for 15 seconds:

P150<cr>

U0 Wait for a "high" on the user input. A "high" is a voltage between +3V and +25V applied to USER IN (J1,3). A simple pushbutton or toggle switch can be used between +5V (J1,1) and USER IN (J1,3) to satisfy this input. Memory usage = **1 byte**.

U1 Wait for a "high" on the user input, holding the user output high while waiting. A "high" is a voltage between +3V and +25V applied to USER IN (J1,3). The USER OUT (J1,2) will go to +5V for the duration of the wait. Memory usage = **1 byte**.

U2 Disable user output when pausing. Use this command when you do not want pauses to affect the state of USER OUT (J1,2). Memory usage = **1 byte**.

U3 Enable output on USER OUT (J1,2) when pausing (reset state) Memory usage = **1 byte**.

U4 User output low. The USER OUT (J1,2) will go to +0V. This is the state of the user output on power-up. This command is used in conjunction with the "U5" command. Memory usage = **1 byte**.

U5 User output high. The USER OUT (J1,2) will go to +5V. This command is used in conjunction with the "U4" command. Memory usage = **1 byte**.

U6 Send "W" to the host and wait for a "G" to continue. The NF90 sends the single character "W" to the host when this command is executed. The NF90 will wait until a "G" is received from the host before proceeding in the program. Memory usage = **1 byte**.

U7 Start of Continuous Index with pulse output. This command is used when it is desirable to make several Indexes on one axis without stopping or slowing between each Index. Instead of stopping a 20 μ sec wide pulse will appear on the USER OUT (J1,2) at each Index distance. This pulse would be used to trigger measurement/sampling equipment. The "U9" command must be used as the last command to decelerate properly to a stop from the last Index. Memory usage = **1 byte**.

Continuous Indexes require the following:

- a) Each Index must be the same motor, speed, and direction.
- b) After the last Index the motor will move an extra distance determined by the deceleration.
- c) The first Index should be long enough (or acceleration high enough) for motor to reach set speed.
- d) The maximum speed that should be used is 2000 steps/sec.
- e) The "D" and the "H" commands can not be used.

Examples:

This example produces a pulse when motor 1 reaches positions 1000,1100,1150,1250, and then runs back to the start position:

S1M1500,U7,I1M1000,I1M100,I1M50,I1M100,U9,I1M0<cr>

This example will Index motor 2 and pulse 100 times:

U7,I2M400,L101,U9<cr>

This example will run continuously producing a pulse each revolution of Motor 1:

U7,I1M400,L0<cr>

U8 Start of Continuous Index sending "@" to the host. This command is the same as the "U7" except the single character "@" is transmitted at each Index distance, instead of a pulse on the USER OUT (J1,2). Always use the highest baud rate possible (9600).
Memory usage = **1 byte**.

U9 End of Continuous Index. This command is used, as the ending command of a Continuous Index, in conjunction with the "U7" or "U8" commands.
Memory usage = **1 byte**.

Bx Backlash Compensation, compensation is on when x=1, off when x=0 (reset state). The NF90 can compensate for mechanical backlash by ending every index in the positive direction. When backlash compensation is on, and a motor makes a negative Index, 20 steps will be added to the Index. The Motor will then immediately reverse, indexing positive 20 steps.
Memory usage = **0 bytes** (this command is immediate, and uses a reserved memory location)

Ox Indicate Limit Switch Over-travel to the host, off when x=0 (reset state), NF90 sends "O" when x=1 and a limit switch is encountered. This command is useful when the host needs to know if a positioner's travel has been exceeded due to a motor stall or an index(es) that are too long. When Indicate Limit Switch Over-travel is on, the NF90 transmits the single character "O" to the host when an indexing motor activates it's limit switch input.
Memory usage = **0 bytes** (this command is immediate, and uses a reserved memory location)

NOTE: limit switches also stop motor motion immediately.

The following are immediate (not stored) commands, therefore they do not use any of the NF90's memory and do not need an ending carriage return or comma:

Q Quit On-Line mode (return to Jog/Slew mode). The "Q" command is used to get back to the power-up state, where the NF90 is in the JOG/slew mode, and the On-Line light is off.

R Run program. The "R" command will start execution of commands stored (program) in the NF90's memory. At the end of the "run" the single character "^" will be transmitted to the host. Additional "R" commands received by the NF90 will repeat the same program. See the "C" command to clear a program from memory. The RUN input (J1,4) functions the same as this "R" command.

N Null (zero) motors 1,2,3 Absolute Position Registers. This command can be used in the JOG/slew or the On-Line mode. The "N" command zeros the position registers that have been counting steps from indexing and/or jog/slewing the motor(s).

K Kill operation in progress. This command will immediately interrupt any running program and reset the user output. If a motor is indexing it will be stopped immediately. If the motor speed is above 1000 steps/sec. when the interrupt occurs, the motor may lose position due to mechanical overshoot (see the "D" command for a less abrupt method to interrupt indexes). The NF90 will transmit the "^" to the host after receiving the "K" command.

V Verify Controller's status, when On-Line the NF90 sends a "B" to the host if it is busy, or an "R" if it is ready. The "V" command is used to poll the NF90 to see if it is busy running a program, or ready to receive more commands. Use of this command is optional, since the NF90 automatically transmits a "^" character to the host when a program has finished. If the NF90 is running a program when it receives a "V" the NF90 will respond by transmitting the single character "B". If the NF90 is idle waiting for a command the NF90 will respond by transmitting the single character "R". In the JOG/slew Mode, the NF90 will send a "J" in response to the "V".

C Clear program from memory. The "C" removes all commands stored in memory. The Absolute Position Registers, state of the Backlash Compensation and Indicate Limit Switch Over-travel will not be affected. The acceleration settings will not be changed and motor speeds will default to the last set speed.

D Decelerate to a stop (interrupts current index in progress). When the NF90 receives the single character "D" while it is indexing a motor, that motor will be decelerated to a stop at the set deceleration. The NF90 will then proceed to the next command in the program. The "D" command has a different function when in the JOG/slew mode, refer to the section on Digitizing for more information.

E Enable On-Line mode with echo on. The single character "E" is used to put the NF90 in the On-Line mode after power-up. All characters the NF90 receives will be echoed back to the host. Refer to the section On-Line Mode for more information.

F Enable On-Line mode with echo **OFF**. The single character "F" is used to put the NF90 in the On-Line mode after power-up. No characters will be echoed back to the host. The NF90 will still respond to motor position and status requests. Refer to the section On-Line Mode for more information.

G Go after waiting or holding. The single character "G" is used to continue when the NF90 is in a single step mode ("H" command) or when a "U6" command is being executed.

H Put Controller on **Hold** (single step mode). When the NF90 receives the single character "H" the Hold Flag will be set. With the Hold Flag set, a "running" program stops after each operation (command) and sends a ":" to the host. An additional stop occurs at the beginning and end of the program and when a loop reaches its last count. When stopped, the "X", "Y", and "Z" commands can be used to read motor position. A "G" will cause the program to continue to the next operation. An "H" toggles the flag off and the program continues as normal. The "K" terminates the program and clears the Hold Flag. This Command allows single stepping through a program for debugging or as a program interrupt from the host.

<Delete>(ASCII 127) Deletes current partial value "keyed-in". Any ASCII character greater than the value 57 will function as a <Delete>. The NF90 will send a "^" character when it receives a <Delete>.

X Send position of motor 1 to the host. When the NF90 receives the single character "X" it will transmit the value from it's motor 1 Absolute Position Register. Below is what the host would receive if motor 1 is at negative 1200. This command can be used when the motor is indexing, however speed should be limited to 500 steps/sec. and the baud rate should be 9600. See the "N" command for information on zeroing the Absolute Position Registers.

-0001200<cr>

Y Send position of motor 2 to the host. When the NF90 receives the single character "Y" it will transmit the value from it's motor 2 Absolute Position Register. Below is what the host would receive if motor 2 is at positive 9201. This command can be used when the motor is indexing, however speed should be limited to 500 steps/sec. and the baud rate should be 9600. See the "N" command for information on zeroing the Absolute Position Registers.

+0009201<cr>

Z Send position of motor 3 to the host. When the NF90 receives the single character "Z" it will transmit the value from it's motor 3 Absolute Position Register. Below is what the host would receive if motor 3 is at negative 20. This command can be used when the motor is indexing, however speed should be limited to 500 steps/sec. and the baud rate should be 9600. See the "N" command for information on zeroing the Absolute Position Registers.

-0000020<cr>

NOTE: When using the above commands, with PCs that operate below 7MHz clock speed, a buffer overflow may occur in the PC's input buffer. Use a lower baud rate if this error condition occurs.

The following commands are for NF90 controllers that are daisy-chained together.

[x] Send commands to the next NF90 in the "chain", x are any of the previous commands. Refer to the section Daisy-chaining NF90 Controllers (P.16) for more information.

& Enable multiple NF90s that are daisy-chained. The "&" command is used in place of the "E" or "F" command when NF90 controllers are daisy-chained together. Refer to the section Daisy-chaining NF90 Controllers (P.16) for more information.

Troubleshooting Procedure

SYMPTOM	POSSIBLE CAUSE	CORRECTIVE ACTION
Power light does not "light" when NF90 is switched on.	Blown fuse.	Check fuse located on the back of the Controller.
	Line cord not fully plugged in	Check line cord at power receptacle on back of Controller
Motor does not operate.	Limit switch(es) circuit open or switches improperly wired or missing.	Check limit switches for proper action and connection. Make sure connector J1 is fully seated
	Wire connections or connector J2 may have come loose	Make sure connector J2 is fully seated and connector's screws are tight.
	In-line fuse located on side of NF90 may have blown.	Check fuse. If fuse is blown, check motor and motor cables.
On-Line light flashes continuously after power-up	RUN input (J1,4) is activated by user switch or jumper.	Open connection to RUN input (J1,4).
Controller does not come On-Line when sent "E" or "F".	RS-232C may not be connected properly.	Trace Transmitted Data, Received Data, and Signal Ground wires from your computer to the NF90.
	Your computer or terminal is not sending upper case letters.	Transmit only upper case letters. The Controller will not respond to lower case.
	Your computer may require a high on its Data Set Ready (DSR) line.	Check with the computer manufacturer to see if the DSR line must be pulled high, or if it can be disabled in software.
NF90 does not come On-Line or program does not operate.	The RS-232-C parameters are not set properly.	Match the RS-232-C settings on the NF90 to those of your computer or terminal.

Your computer does not receive data from the Controller.	Your computer may require a "high" on its Request To Send (RTS) line.	Check with the computer manufacturer to see if the RTS line must be pulled high, or if it can be disabled in software.
Motor stalls, it does not move at all.	Inertia in system is too high, or mechanism has seized.	Hand rotate the system to locate any binding. A larger motor, or a different ratio (pitch) may be required.
Motor stalls, after rotating slightly.	Acceleration too high.	Use a lower acceleration /deceleration.
	Motor cannot overcome friction or load.	Check mechanism for ease of movement. Load will have to be reduced or counterbalanced.
Motor stalls before reaching maximum speed.	Motor torque decreases as it's speed increases.	Reduce steps/sec. setting of motor.
With Motor removed from equipment, Motor runs erratically at all speeds, has no torque.	Broken motor cable.	Check cable and connector for broken wires and repair breaks.
	Faulty Motor. (Rare)	Replace Motor.
Motor or system resonates (vibrates loudly).	The motor speed is the motor or system's natural resonant frequency (common at low speeds).	Increase or decrease speed to avoid resonance points. A damper or flywheel added to the motor shaft or lead screw may dampen the resonance.
Controller stops operating for no apparent reason.	Inductive surge on AC power-line.	Isolate or remove any equipment that may be putting "spikes" on the power-line.
	Inductive/Static surges coming in the I/O connections.	Make sure all external equipment connected is properly grounded and inductive loads are isolated from the NF90.

Specifications

FUNCTIONAL

Packaged Controller/Driver, using Microcomputer control of stepping motors. Operates one to three (dependent on model) motors, one-at-a-time.

Interactive limit switch inputs (TTL), (CW and CCW for each axis).

One User Input (0V to +3V min., -25V to +25V max.), and one User Output (0 or +5V, 10 ma sinking and 3 ma sourcing capability).

Programming through full-duplex RS-232-C; 300,1200,4800,9600 Baud (switch settable), 7 Data bits, Even parity, 2 Stop bits, ASCII; special configurations with 8 data bits, odd or no parity, are available.

User available RAM for program storage is 101 bytes

Remote Run and joystick Inputs (TTL).

Eight foot (244 cm) motor and limit switch cables with connectors

MOTOR COMPATIBILITY

1.8° PM 6 or 8 lead stepping motors, 4.7 amp/phase maximum. Factory matched for a particular motor current, motors on each axis to be the same amp/phase value.

PHYSICAL

Weight: 7.2 lbs. (3.2 kg)
Height: 5.0 inches (12.7 cm)
Width: 10.8 inches (27.4 cm)
Depth: 7.8 inches (19.8 cm)

ELECTRICAL REQUIREMENTS

90 to 130 VAC 50/60Hz, 150 watts

ENVIRONMENTAL

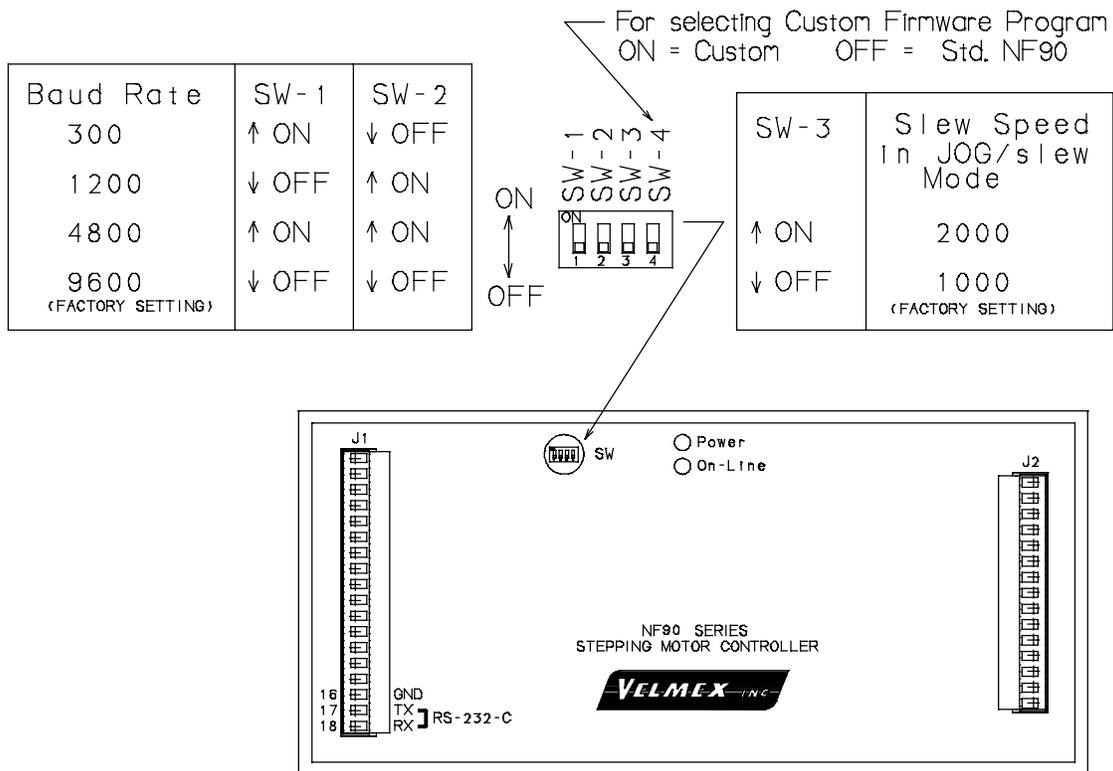
35° to 95° F (2° to 35° C) Convection cooled

MODELS

Model # NF90-1 One motor version
Model # NF90-2 Two motor version
Model # NF90-3 Three motor version

Appendix A

Switch settings and RS-232-C Connections (7 Data, Even, 2 Stop)



Suggested RS-232 Cable Configuration



IBM PC, XT, AT, &
Compatibles

NF90
Controller



Serial (25 Pin)

J1

TX	2	_____	18	RX
RX	3	_____	17	TX
GND	7	_____	16	GND
RTS	4	□		
CTS	5	□		
DSR	6	□		
DTR	20	□		

Serial (9 Pin)

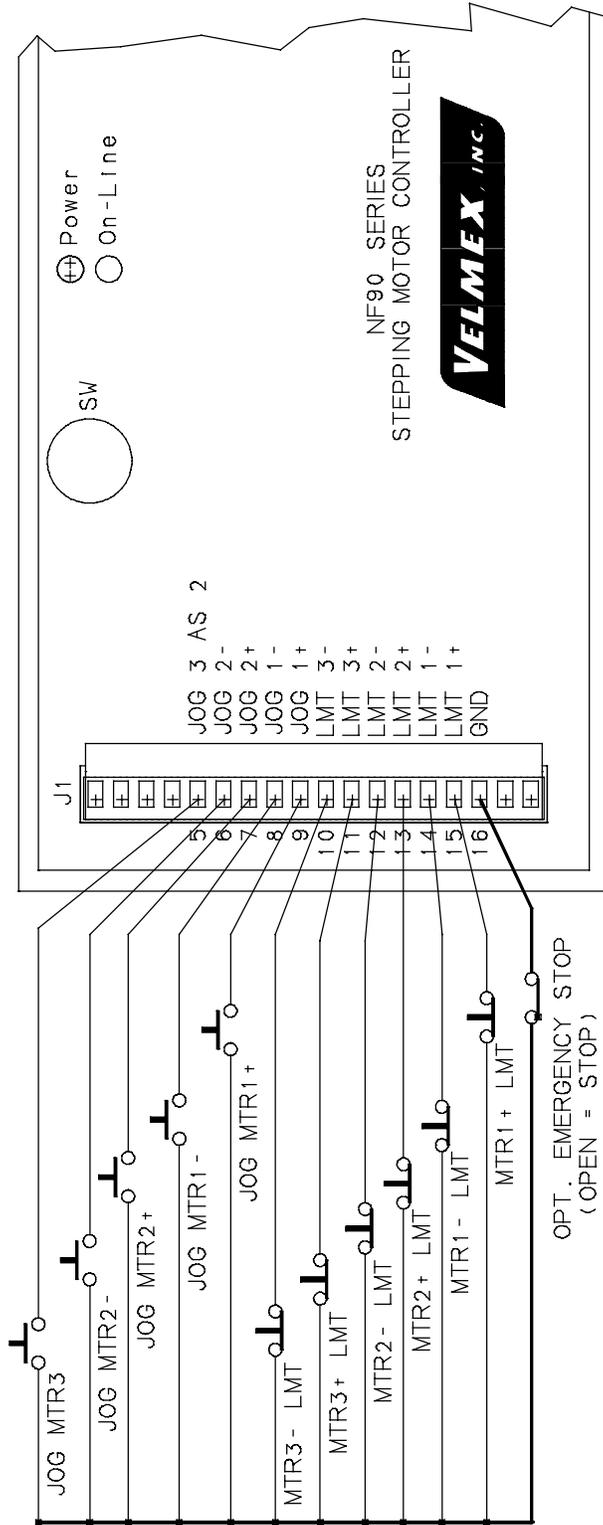
J1

RX	2	_____	17	TX
TX	3	_____	18	RX
GND	5	_____	16	GND
DTR	4	□		
DSR	6	□		
RTS	7	□		
CTS	8	□		

Appendix B

Limit Switch and Joystick Connections

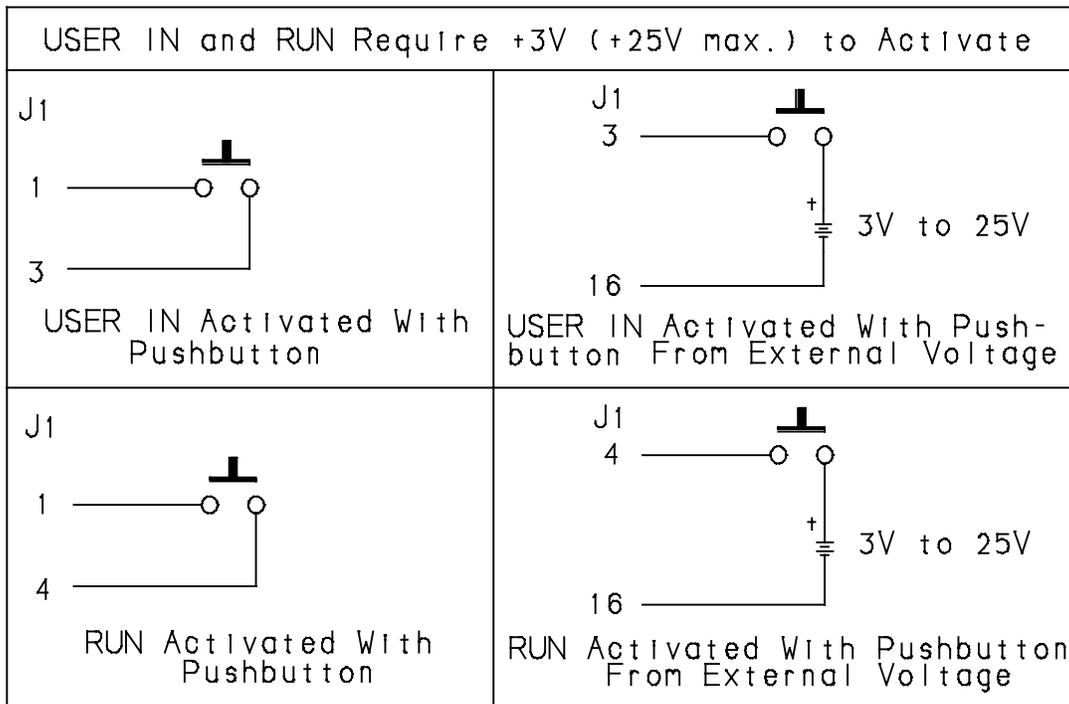
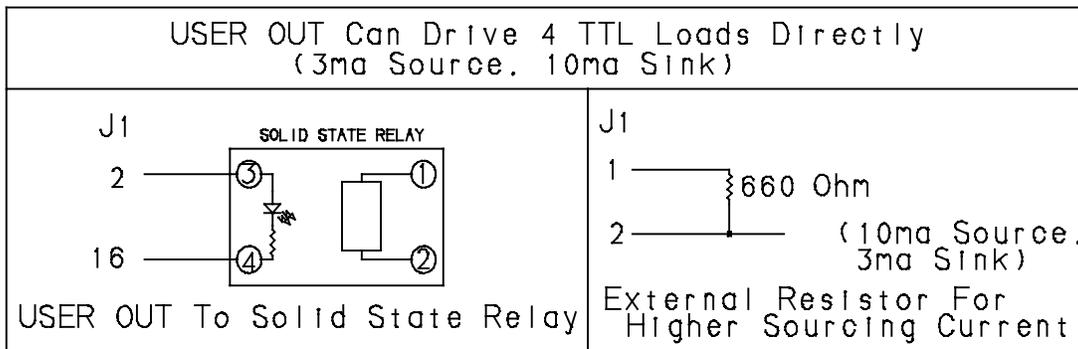
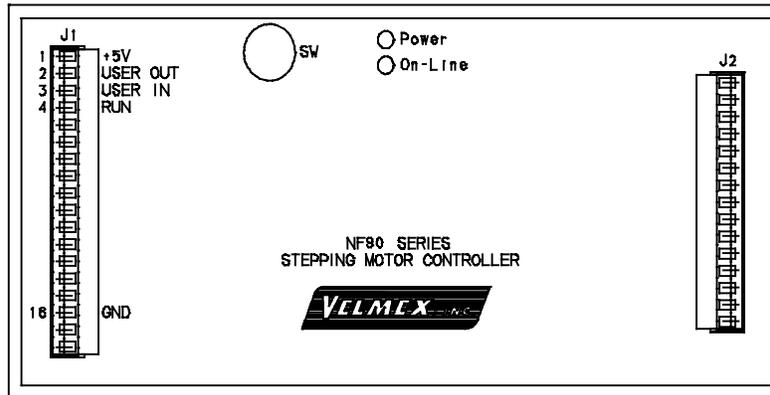
JOG/SLEW WHEN PUSHBUTTON CLOSED



MOTOR STOPS WHEN LIMIT PUSHBUTTON OPENS

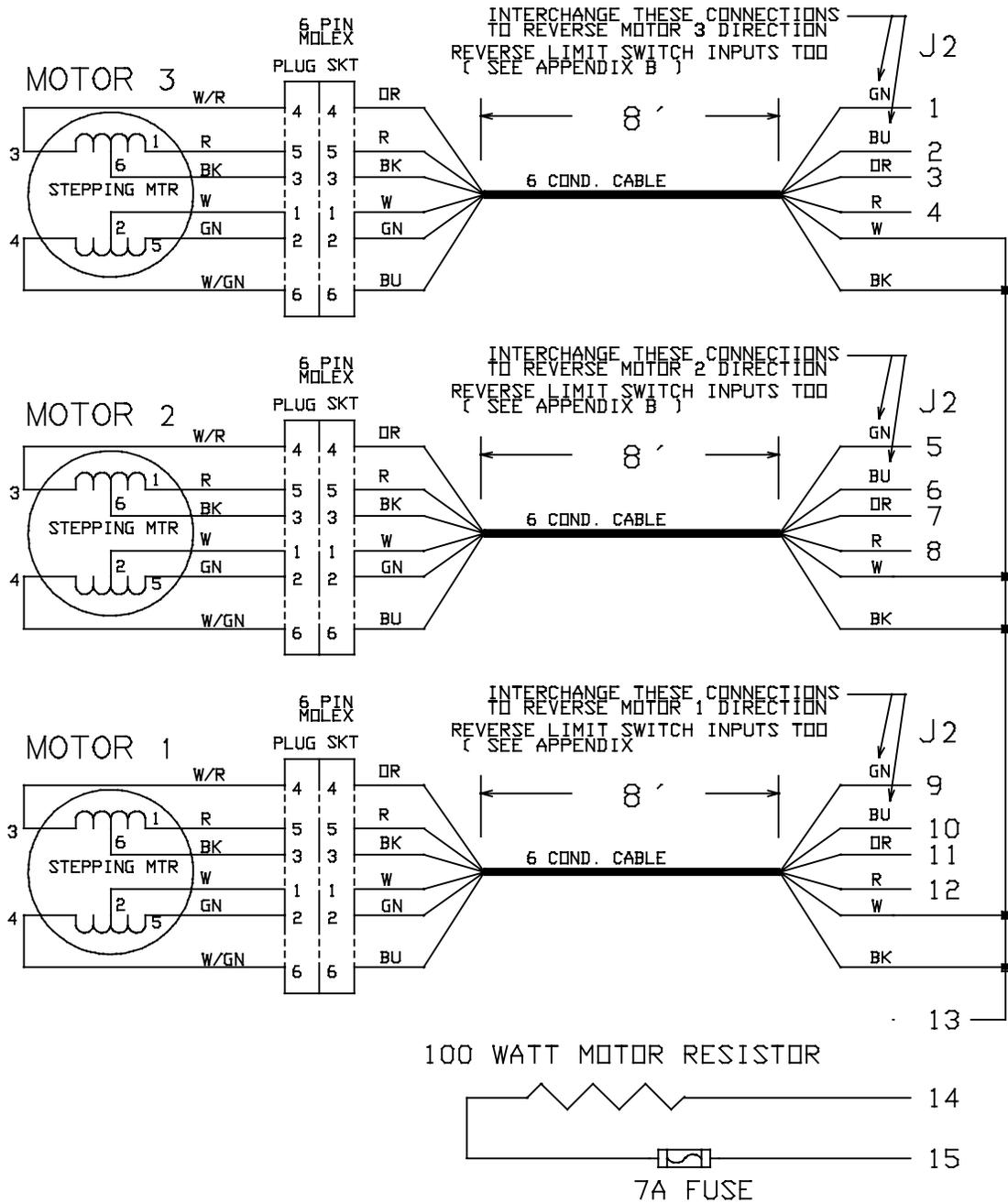
Appendix C

USER OUT, USER IN, and RUN Connections



Appendix D

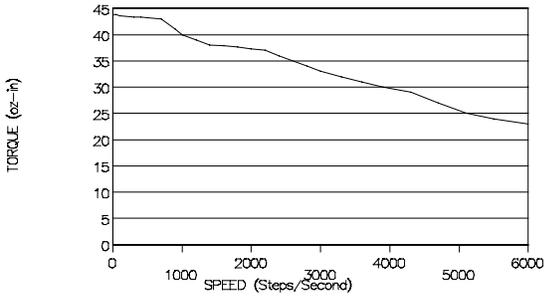
Motor Connections



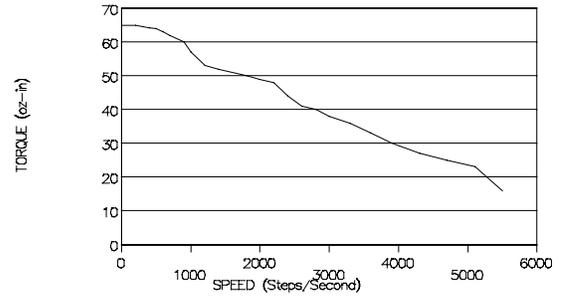
Appendix E

Motor Performance (Superior Electric / Bodine model numbers)

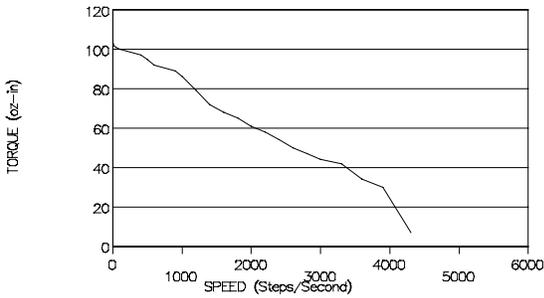
6108 / 2400



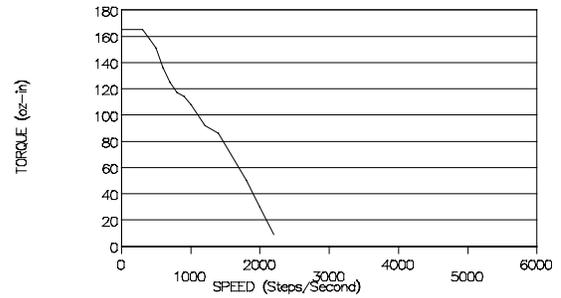
6209 / 2401



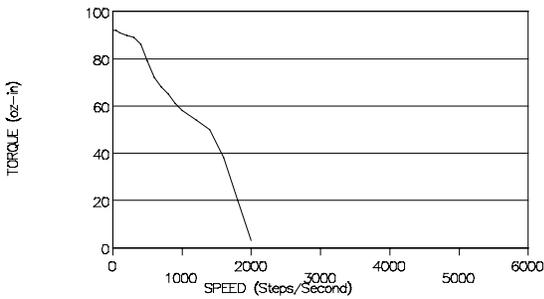
9109 / 2003



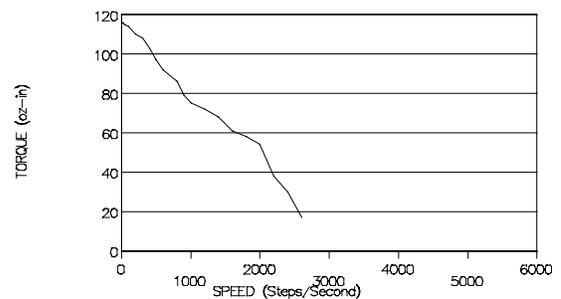
9209 / 2004



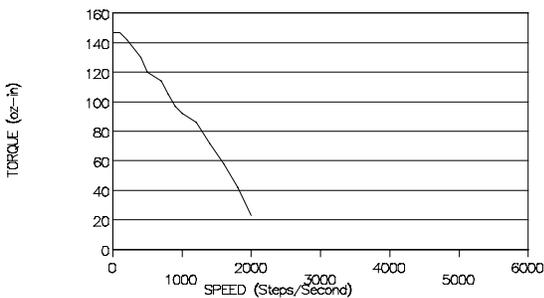
6204



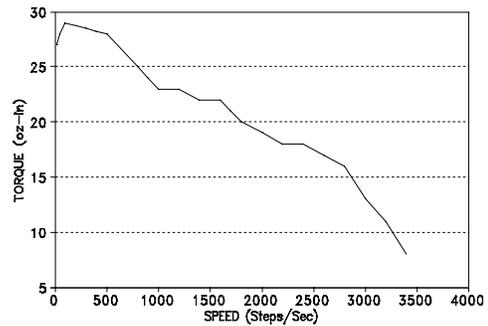
6306 / 2411



9106



Vexta PX245-01AA





Limited Two Year Warranty

Velmex warrants this Controller against defects in material and workmanship for a period of two years from date of shipment. In case of defect, Velmex will repair this Controller without charge. The user will be responsible for shipment to Velmex, and will be charged a minimum of \$45.00 for non-warranty service.

For service under this warranty:

1. Contact Velmex to obtain a Return Authorization number (Tel. 716-657-6151).
2. Pack Controller in the original container or equal.
3. Enclose a description of the problem, along with the return address, the name of the end user, his/her telephone and fax number.
4. Ship Controller Prepaid and insured to :

Velmex, Inc
7550 State Route 5&20
Bloomfield, NY 14469

Exclusions and Limitations:

This warranty covers step motor controllers manufactured by Velmex, Inc. External electrical cables and connectors are not covered for wear and breakage. This warranty does not extend to any damage or malfunction resulting from misuse, neglect or accident. Except for any implied warranty, this warranty contains the entire obligation of Velmex, and the remedies described above are the exclusive remedies under this warranty or any implied warranty. The duration of any implied warranty is limited to two years. **In No Event Shall Velmex Be Liable For Incidental or Consequential Damages.**