ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

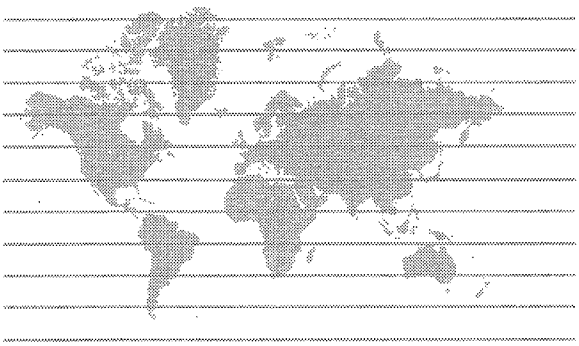
Find the *Parker PMC-6270* at our website: ***Click HERE***

Parker
Motion & Control

Bulletin 1480-M4
User Guide

PMC-6270

Effective: June, 1995



6000 Series Following User Guide

p/n 88-014546-01 B

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Important User Information

Motion Architect is a registered trademark of Parker Hannifin Corporation. CompuCAM and Servo Tuner are trademarks of Parker Hannifin Corporation. AT and IBM are registered trademarks of International Business Machines Corporation. Microsoft and MS-DOS are registered trademarks, and Windows is a trademark of Microsoft Corporation.

The information in this user guide, including any apparatus, methods, techniques, and concepts described herein, are the proprietary of Parker Hannifin Corporation or its licensors, and may not be copied, disclosed, or used for any purpose not expressly authorized by the owner thereof.

Since Parker Hannifin Corporation constantly thrives to improve all of its products, we reserve the right to change this user guide and equipment mentioned therein at any time without notice.

For assistance in the United States, contact:

Parker Hannifin Corporation
Hydraulic Valve Division
520 Ternes Avenue
Elyria, Ohio 44035
Tel: 216-366-5200
Fax: 216-366-5253

For assistance in the Europe, contact:

Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
Tel: 0202-690911
Fax: 0202-600820

WARNING

FAILURE OR IMPROPER SELECTION OR IMPROPER USE OF THE PRODUCTS AND/OR SYSTEMS DESCRIBED HEREIN OR RELATED ITEMS CAN CAUSE DEATH, PERSONAL INJURY AND PROPERTY DAMAGE.

This document and other information from Parker Hannifin Corporation, its subsidiaries and authorized distributors provide product and/or system options for further investigation by users having technical expertise. It is important that you analyze all aspects of your application and review the information concerning the product or system in the current product catalog. Due to the variety of operating conditions and applications for these products or systems, the user, through its own analysis and testing, is solely responsible for making the final selection of the products and systems and assuring that all performance, safety and warning requirements of the application are met.

The products described herein, including without limitation, product features, specifications, designs, availability and pricing, are subject to change by Parker Hannifin Corporation and its subsidiaries at any time without notice.

Offer of Sale

The items described in this document are hereby offered for sale by Parker Hannifin Corporation, its subsidiaries or its authorized distributors. This offer and its acceptance are governed by the provisions stated in the "Offer of Sale".

Copyright 1995, Parker Hannifin Corporation, All Rights Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Change Summary

6000 Following User Guide

Rev B

May 1995

Below is a summary of the primary technical changes to this manual since the last revision. This manual, p/n 88-014217-01**B**, supersedes 88-014217-01**A**.

Command Descriptions Removed

Descriptions for the Following-related commands have been removed, and are now located in the *6000 Series Software Reference Guide*.

Topic	Description
615n Resolver Feedback	**Correction:** The 615n may not use its commanded position or its internal resolver as a master.
Error Messages	**Corrections & Additions:** • "BAD RATIO" changed to "INVALID RATIO". • Additional possible cause for the message "INVALID CONDITIONS FOR COMMAND": The FOLMD command value is too small to achieve the preset distance and still remain within the FOLRN/FOLRD ratio.
Programming Examples	**Amendments:** Amended several programming examples for compliance with stepper programming requirements.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS

Chapter 1. Introduction to 6000 Following

Ratio Following 1
Technical Overview 1
- What can be a master? 1
- Performance Considerations 2
- Following Status 2

Chapter 2. Ratio Following Motion

Overview 3
Ratio Following Set-up Parameters 3
- Define the Master and Slave Axes 4
- Define the Master and Slave Scaling Factors .. 5
- Define the Slave-to-Master Following Ratio 7
- Define the Master Distance 8
- Enable the Following Mode 8

Slave vs. Master Move Profiles 8
- Continuous Positioning Mode Moves 9
- Preset Positioning Mode Moves 10

Performing Phase Shifts 11
Summary of Ratio Following Commands 13
Electronic Gearbox Application 14
Trackball Application 14

Chapter 3. Master Cycle Concept

Overview 17
Master Cycle Commands 18
- Master Cycle Length (FMCLEN) 18
- Restart Master Cycle Counting (FMCNEW or TRGFNcx1) 18
- Initial Master Cycle Position (FMCP) 19
- Transfer and Assignment/Comparison of Master Cycle Position and Number 19
- Using Conditional Statements with Master Cycle Position (PMAS) 20

Summary of Master Cycle and Wait Commands 21
Continuous Cut-to-Length Application 22

Chapter 4. Technical Considerations for Following

Overview 25
Master Position Prediction 25
Master Position Filtering 26
Following Error 27
Maximum Velocity and Acceleration 28
Dynamic Position Maintenance 28
Factors Affecting Following Accuracy 29
- Resolution of the Master 29
- Resolution of the Slave 29
- Position Sampling Accuracy 30
- Accuracy of the Slave Motor and Drive 30
- Accuracy of the Load Mechanics 30
- Master Position Prediction 30
- Master Velocity Relative to Master Position Prediction & Master Position Filtering 30
- Tuning 31
- Dynamic Position Maintenance 31
- Repeatability of the Trigger Inputs & Sensors 31

Preset vs. Continuous Following Moves 32
Master and Slave Distance Calculations 33
- When the Slave is in Continuous Positioning Mode (MC1) 33
- When the Slave is in Preset Positioning Mode (MCØ) 33
- Distance Calculation Example 34
- Beware of Roundoff Error (Scaling only) 34

Using Other Features with Following 35

Chapter 5. Troubleshooting

Symptoms and Causes 37
Error Messages 38

Appendix A — Following Commands 39

Index 41

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ①

Introduction to 6000 Following

As part of its standard features, the 6000 Series Controller family allows you to solve applications requiring *Ratio Following*.

Ratio Following

Ratio Following is, essentially, controlled motion based on the measurement of external motion. This includes concepts such as an electronic gearbox, trackball, slave feed-to-length, as well as complex changes of ratio as a function of master position. Ratio Following can include continuous, preset, and registration-like moves in which the velocity is replaced with a ratio.

The slave may follow in either direction and change ratio while moving, with phase shifts allowed during motion at otherwise constant ratio. Ratio changes or new moves may be dependent on master position or based on receipt of a trigger input. Also, a slave axis may perform Following moves or normal time-based moves in the same application because Following can be enabled and disabled at will.

In Ratio Following, acceleration ramps between ratios will take place over a user-specified master distance. Product cycles can be easily specified with the master cycle concept.

This user guide highlights the capabilities of the 6000 Following features and provides useful application examples. If you need more details on the operation or syntax of a particular command, please refer to the *6000 Series Software Reference Guide*.

Technical Overview

Before delving into the specifics of Ratio Following, let's take a look at how the 6000 controller *follows*.

What can be a master?

Any axis on the 6000 controller can be slaved to any transducer device input or commanded position of any other axis. This source is known as the *master input*, or *master*. Up to 4 axes (depending on which 6000 product you have) can be following at the same time with the same or different master.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The options available for master input sources differ by product (see table).

Master Input Options (✓ = yes)	AT6400	AT6200	AT6250	AT6450	610n	615n	620n	625n	6270
Commanded Position	✓	✓	✓	✓	n/a	n/a	✓	✓	✓
Input Devices:									
• Incremental Encoder	✓	✓	✓	✓	✓	✓	✓	✓	✓
• **ANI** Analog Input **(-ANI option only)**	n/a	n/a	✓	✓	n/a	✓	n/a	✓	✓
• Linear Displacement Transducer (LDT)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	✓

SERVOS: For servo controllers, an axis may not be slaved to its own feedback device input.

STEPPERS: For stepper controllers, an axis may be slaved to its own encoder input, as long as that axis does not need encoder feedback for encoder positioning (`ENC1` mode) or stall detection (`EPM1` mode).

For more information on assigning a master for a particular slave, refer to *Define the Master and Slave Axes* in Chapter 2, or refer to the `FOLMAS` command description in the *6000 Series Software Reference Guide*.

Performance Considerations *(see also Chapter 4)*

Position Sampling Period
Steppers – 2 milliseconds.
Servos – *system update period* (depends on `SSFR` and `INDAX` command values – see `SSFR` command description).

When a slave is following a master, the 6000 controller does not simply measure the master velocity to derive slave velocity. Instead, the 6000 controller samples the master position (*position sampling period*) and calculates the corresponding slave position command. This is true even if the slave is in the process of changing ratios. A slave is not simply following velocity, but rather position. With this algorithm, the master and slave position or phase relationship is maintained indefinitely, without any drift over time due to velocity measurement errors.

The 6000 controller also measures master velocity by measuring the change in master position over a number of sample periods. The present master velocity and position may be used to calculate the next commanded slave position, so the slave has no velocity-dependent phase delay. This concept is known as *Master Position Prediction* and may be enabled or disabled as needed with the `FPPEN` command.

The 6000 controller's default Following algorithm should work well for most applications; however, you can change the Following algorithm to meet application-specific needs. For instance, suppose that the speed of the master is very slow, or has some vibration. For a case like this, the 6000 controller allows you to filter the master position signal to generate a smooth slave position command. This is known as *Master Position Filtering* and is programmed with the `FFILT` command.

The major points of emphasis in the design of the 6000 controller's Following features are its ability to maintain the programmed phase relationships between master and slave, and design precise motion profiles.

More details on the technical aspects of Following can be found in Chapter 4, or in the associated command descriptions in the *6000 Series Software Reference Guide*.

Following Status (`TFS` and `FS` Commands)

Many of the Following features described in this document have associated status bits that can be displayed or used in assignment or comparison operations (see `TFS` and `FS` command descriptions). The portions of this document which describe those features also summarize the related status bits.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ②

Ratio Following Motion

Overview

Ratio Following can include continuous, preset, and registration-like moves in which the velocity is replaced with a ratio. The slave may follow in either direction and change ratio while moving, with phase shifts allowed during motion. Ratio changes or new moves may be dependent on master position or based on receipt of a trigger input. Also, a slave axis may perform Following moves or normal time-based moves in the same application because Following can be enabled and disabled at will. *Product cycles* (operations which repeat with periodic master travel) can be easily specified with the master cycle concept (see Chapter 3).

Ratio Following Set-up Parameters

Prior to executing a Following move, there are several set-up parameters that must be specified. These parameters may be established in any order:

☞ *Programming examples — see application examples later in this user guide.*

- Define the master and slave axes
- Define scaling factors for the master and the slave (if required)
- Define the slave-to-master Following ratio
- Define the master distance (define scaling first)
- Enable the Following Mode

Following Status
(see TFS and FS commands)

Following Status bits 5-8 (see table below) are meant to indicate the status required for Following motion (i.e., a master must be assigned, Following must be enabled, the master must be moving, and for many features, the master direction must be positive).

Bits 7-8 represent master motion and master direction respectively. Unless the master is a commanded position of another axis, it is likely that minor vibration of the master will cause these bits to toggle on and off, even if the master is nominally "at rest". These bits are meant primarily as a quick diagnosis for the absence of master motion, or master motion in the wrong direction. Many features require positive master counting to work properly.

Bit #	Function (yes = 1; no = ø)
5	FOLMAS Active..... A master is specified with the FOLMAS command.
6	FOLEN Active..... Following has been enabled with the FOLEN command.
7	Master is Moving..... The specified master is currently in motion.
8	Master Dir Neg..... The current master direction is negative. (bit must be cleared to allow Following move in preset mode–MCø).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Define the Master and Slave Axes (FOLMAS)

The FOLMAS command defines the masters and the slaves. The command syntax is FOLMAS<±ii>,<±ii>,<±ii>,<±ii>, where each <±ii> represents the configuration for axes 1 through 4, from left to right. *(Note that the number of axes available differs by product).* The configuration for each axis (±ii) is further defined as follows:

- Sign bit (±): Specifies the count direction of the master source which will result in positive master travel counts. The sign bit is not meant to be used simply to change the direction of slave motion. That function can be done with the sign of the D command. Rather, the sign bit is used to allow forward motion of the physical master (e.g., conveyor belt, rotating wheel, or the continuous feed of material or product) to result in positive counts. Several features described later in this document require increasing master counts for proper operation. These include Following motion in preset positioning mode (MCØ) , master cycle counting, and executing GOWHEN based on master cycle position.
- First i: Selects the master input axis number (1 through 4) that you are assigning to the slave.
- Second i: Selects the master input (input device or commanded position) according to the axis number specified with the first i. The availability of master input sources differs by product, as indicated in the table below.

NOTE

- Servo controllers: The slave axis cannot use its own commanded position or its currently selected feedback device (encoder, ANI, or LDT) as the master input.
- Stepper controllers: The slave axis cannot use its own commanded (motor) position as the master input. Also, a slave axis that is using the encoder step mode (ENC1) cannot use its own encoder as the master input.
- Multiple axes may be slaved to the same count source (e.g., encoder) from the same master. However, multiple axes may not be slaved to different count sources (e.g., encoder and commanded position) from the same master.

Product	Options for Second i	Measurement *
AT6n00, 620n & OEM6200	1—Encoder 4—Commanded (motor) position	Encoder counts Motor counts
AT6n50, 625n & OEM6250	1—Encoder 2—ANI input **(ANI option only)** 4—Commanded position	Encoder counts ADC counts Feedback device counts
610n	1—Encoder	Encoder counts
615n	1—Encoder 2—ANI input **(ANI option only)**	Encoder counts ADC counts
6270	1—Encoder (axis 1 only) 2—ANI input (6270-ANI only) 3—LDT 4—Commanded position	Encoder counts ADC counts LDT counts Feedback device counts

* If scaling is enabled (SCALE1), the measurement of the master is scaled by the SCLMAS value.

Example ☞ As an example, suppose the **ENCODER 3** input is to be the master input for slave axis 1, and forward travel of the physical master (e.g., conveyor belt) results in negative counts on **ENCODER 3**. Given these operating constraints, you would use the FOLMAS-31,,, command.

The default setting is that all axes are disabled from being slaves (FOLMAS+Ø,+Ø,+Ø,+Ø). If you do not want a particular axis to be a slave, simply leave it not configured (e.g., FOLMAS,+31,, command configures only axis #2 as a slave to encoder #3; the rest of the axes are left in the default state—not slaved). If an axis is currently configured as a slave, you can disable its slave status by putting a zero (Ø) in the parameter field (e.g., the FOLMAS,Ø,, command disables axis 2 from being a slave axis).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

As soon as the master/slave configuration is specified with the FOLMAS command, a continuously updated relationship is maintained between the slave's position and the master's position. The update period for stepper controllers is 2 ms. The update period for servo controllers (*system update*) depends on the current SSFR and INDAX command values (see SSFR command description in the *6000 Series Software Reference Guide*.)

FOLMAS Setting Not Saved in Non-Volatile Memory

The FOLMAS configuration is not saved in non-volatile memory. Therefore, you may wish to include it in the power-up program (STARTP) for serial-based controllers, or in the first program downloaded for bus-based controllers.

Notes for Stepper Controller Users

- If the slave axis is in encoder step mode (ENC1), then the encoder resolution (ERES) of the slave axis is also used. For that reason, *the encoder resolution may not be changed after the FOLMAS command configures an axis as a slave.*
- If the slave axis is in encoder step mode (ENC1), or if stall detect is enabled (ESTALL1), or if position maintenance is enabled (EPM1), that axis cannot use its own encoder input as the master.

Define the Master and Slave Scaling Factors (SCLMAS) . . . . . *if required*

IF SCALING IS NOT USED (SCALEØ)

Servos:

- Slave distance values are entered in the units of the currently selected feedback device (i.e., the device selected with the SFB command).

  Master distance values are entered in actual master counts (counts from the selected encoder, ANI, LDT, or commanded position).
- Velocity and accel/decel units of measure depend on the feedback source selected:

  Encoder feedback – revs/sec and revs/sec^2
  (counts output per rev depend on encoder resolution set with the ERES command).

  LDT feedback – inches/sec and inches/sec^2
  (counts output per inch depend on LDT resolution set with the LDTRES command.

  ANI feedback – volts/sec and volts/sec^2 (resolution is 819 ADC counts per volt).

Steppers:

- Slave distance values are entered in motor steps if the motor step mode is enabled (ENCØ), or encoder steps if the encoder step mode is enabled (ENC1). *If an axis is in the Encoder Step Mode (ENC1), it cannot use its own encoder input as the master.*

  Master distance values are entered in motor steps if the master is the motor position of another axis, or encoder steps of the master is an encoder.
- Velocity and accel/decel values are entered in motor revs/sec and revs/sec^2, respectively (the steps output per rev depend on the resolution setting—DRES).

Velocity and Accel/Decel Scaling

Velocity and accel/decel scaling factors for the slave are set with the SCLV and SCLA commands, respectively. *Refer to the 6000 controller's user guide for details on using velocity and accel/decel scaling.* The SCLV and SCLA scaling factors should be set to the same value as the slave distance scaling factor (SCLD) to establish common user units (e.g., distance in inches, velocity in inches/sec, accel in inches/sec^2).

Distance Scaling

It is useful to define master and slave multipliers so later programming can take place in *user units*. The SCLD command defines the slave's distance scale factor, and the SCLMAS command defines the master's distance scale factor. The Following-related commands that are affected by SCLD and SCLMAS are listed in the table below. **NOTE**: Scaling must be enabled with the SCALE1 command before these commands will have any effect.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Commands Affected by Master Scaling (SCLMAS)	Commands Affected by Slave Scaling (SCLD)
FMCLEN: *Master Cycle Length*	FOLRN: *Slave-to-Master Ratio (Numerator)*
FMCP: *Master Cycle Position Offset*	FSHFD: *Preset Phase Shift*
FOLMD: *Master Distance*	GOWHEN: *Conditional GO* (left-hand variable ≠ PMAS)
FOLRD: *Slave-to-Master Ratio (Denominator)*	TPSHF & [PSHF]: *Net Position Shift of Slave*
GOWHEN: *Conditional GO* (left-hand variable is PMAS)	TPSLV & [PSLV]: *Position of Slave Axis*
TPMAS & [PMAS]: *Position of Master Axis*	
TVMAS & [VMAS]: *Velocity of Master Axis*	

As the distance scaling factor (SCLMAS or SCLD) changes, the resolution of all distance commands and the number of positions to the right of the decimal point also change (see table below). A distance value with greater resolution than allowed will be truncated (e.g., if scaling is set to SCLD25ØØØ, the FSHFD1.99999 command would be truncated to FSHFD1.9999). 6270 users: shift the values in the "Distance Range" column one decimal place to the left.

SCLD or SCLMAS (steps/unit)	Distance Resolution (units)	Distance Range (units)	Decimal Places
1 - 9	1	0 - ±999,999,999	0
10 - 99	0.1	0.0 - ±99,999,999.9	1
100 - 999	0.01	0.00 - ±9,999,999.99	2
1000 - 9999	0.001	0.000 - ±999,999.999	3
10000 - 99999	0.0001	0.0000 - ±99,999.9999	4
100000 - 999999	0.00001	0.00000 - ±9999.99999	5

NOTE **DEFINE SCALING FIRST** **NOTE**

If scaling is desired within a stored program, you must enable scaling (SCALE1) and define the scaling factors (SCLA, SCLV, SCLD, & SCLMAS) prior to defining (DEF), uploading (TPROG), or running (RUN) the program. This allows the 6000 Series product to store, display, and execute the scaled distance, acceleration, and velocity values within the stored program. This can be accomplished by defining all scaling factors via a terminal emulator just before defining or downloading the program; or you can put the scaling factors into a startup (STARTP) program (stand-alone controllers only) or a program that must be run prior to defining or downloading the program.

NOTE **FRACTIONAL STEP TRUNCATION** **NOTE**

If you are specifying master distance values (FOLMD), when the master distance scaling factor (SCLMAS) and the distance value are multiplied, a fraction of one step may possibly be left over. This fraction is truncated when the distance value is used in the move algorithm. This truncation error can accumulate when performing several moves over the specified master distance. To eliminate this truncation problem, set the master scale factor (SCLMAS) to 1, or a multiple of 10.

Example Distance Scaling Scenario

Typically, the master and slave scale factors are programmed so that master and slave units are the same, but this is not required. Consider the scenario below as an example.

The master is a 1000-line encoder (4000 counts/rev post-quadrature) mounted to a 50 teeth/rev pulley attached to a 10 teeth/inch conveyor belt, resulting in 80 counts/tooth (4000 counts/50 teeth = 80 counts/tooth). To program in inches, you would set up the master scaling factor with the SCLMAS8ØØ command (80 counts/tooth * 10 teeth/inch = 800 counts/inch).

The slave axis is servo motor with position feedback from a 1000-line encoder (4000 counts/rev). The motor is mounted to a 4-pitch (4 revs/inch) leadscrew. Thus, to program in inches, you would set up the slave scaling factor with the SCLD16ØØØ command (4000 counts/rev * 4 revs/inch = 16000 counts/inch).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Finding the Scale Factors

If the SCLD (slave) and SCLMAS (master) scale factor values are not immediately obvious, use the procedure below for help.

Slave *(applicable to servos, and steppers with encoder feedback)*:

① Disable scaling with the SCALEØ command.

② Servos: Issue the TFB command to display the position (based on the active feedback source).

Steppers: Issue the TPE command to display the encoder position.

Write down this value for later comparison.

③ Making sure the feedback device is monitoring movement of the slave, move the slave a known distance in the desired units. For instance, if you want to program in 5-inch units, move the slave 5 inches. This could be done in the jog or joystick modes, or perhaps by disabling the drive (DRIVEØ) and physically moving the slave.

④ Issue the position status command (TFB or TPE) again. The difference between the position values before and after the move, divided by the move distance in step 3, is the value you should use in the SCLD command parameter.

Master:

① Disable scaling with the SCALEØ command.

② Issue the TPMAS command to display the position. Write down this value for later comparison.

③ Move the master input a known distance in the desired units. For instance, if you are going to follow the motion of a conveyor belt and you want to program in 6-inch increments, move the conveyor 6 inches.

④ Issue the TPMAS command again. The difference between the TPMAS values before and after the move in step 3, divided by the distance of the move, is the value you should use in the SCLMAS command parameter.

Define the Slave-to-Master Following Ratio (FOLRN and FOLRD)

The FOLRN and FOLRD commands establish the goal ratio between the slave and master travel, just as the V command establishes the goal velocity for a typical non-Following move. The FOLRN command specifies the ratio's numerator (slave travel), and the FOLRD command specifies the ratio's denominator (master travel). If the denominator (FOLRD) is not specified, it is assumed to be 1.

FOLRN and FOLRD are specified with two positive numbers, but the resulting ratio applies to moves in both directions; the actual slave direction will depend on the direction commanded with the D command and master direction. Numeric variables (VAR) can be used with these commands for slave and/or master parameters (e.g., FOLRN(VAR1) : FOLRD3). The maximum value of the resulting quotient is 127 to 1.

For a preset Following move (MCØ mode), the FOLRN/FOLRD ratio represents the maximum allowed ratio. For a continuous move (MC1 mode), it represents the final ratio reached by the slave.

Example ☞ As an example, assume the slave-to-master ratio is set to 5-to-3 for an axis (FOLRN5 : FOLRD3). The first parameter (5) is scaled by the SCLD value to give slave steps. The second parameter (3) is scaled by the SCLMAS value to give master steps. If the SCLD setting is 25000 and the SCLMAS setting is 4000, the slave-to-master step ratio would be 5 * 25000 to 3 * 4000, or 125 slave steps for every 12 master steps.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Define the Master Distance (FOLMD)

The "master distance" for moves in the Following mode (FOLEN1) is analogous to the move time for normal time-based moves with Following disabled (FOLENØ). For time-based moves, the time required to ramp to a new velocity (MC1 mode) or move to a new position (MCØ mode) is determined indirectly by the acceleration (A), deceleration (AD), and velocity (V) command values. For Following mode moves, a ramp to a new ratio (MC1 mode) or a move to a new position (MCØ mode) takes place over a specific master distance, not over a specific time. This distance is defined directly by the user with the FOLMD command.

In other words, the FOLMD command defines the master distance over which a preset slave move will take place, or the master distance over which a continuous slave move will change from its current ratio (including zero) to the commanded ratio (ratio established by FOLRN and FOLRD).

By carefully specifying a master distance (FOLMD), a precise position relationship between master and slave during all phases of the profile is ensured.

☞ **HINT**: If the slave is in continuous mode (MC1) and the master is starting from rest, setting FOLMD to Ø will ensure precise tracking of the master's acceleration ramp.

If scaling is enabled (SCALE1), the FOLMD value is scaled by the SCLMAS parameter.

Examples and more information on this topic can be found below in the section titled *Slave vs. Master Move Profiles*.

Enable the Following Mode (FOLEN1)

When an axis is configured as a slave with the FOLMAS command, it will continuously monitor the position and motion of its master, even if the slave is at rest. This allows subsequent motion to be related to the motion of the master via ratios (FOLRN/FOLRD) and ramping over master distances (FOLMD). Such moves are done with Following enabled (FOLEN1).

It is also possible, and sometimes desirable, to have the slave motion independent of master motion, yet still "aware" of master position. For example, a move may need to start at a specified master position, yet finish in a fixed time, independent of the master speed. This move would be performed with Following disabled (FOLENØ).

Following may be enabled or disabled between moves, as needed, without affecting the monitoring of the master.

If a move is performed with Following disabled, its motion profile is determined by the acceleration, deceleration, and velocity specified with the A, AD, and V commands. Its motion is the same as if the axis were not configured as a slave, but the axis does monitor the master.

If a move is performed with Following enabled, its profile is determined by the specified master distance (FOLMD) and Following ratio (FOLRN/FOLRD). The next section describes such profiles.

Slave vs. Master Move Profiles

Following Status (TFS and FS) bits 1-4 indicate the status of slave following motion. They mimic the meaning and organization of Axis Status (TAS and AS) bits 1-4, except that each bit indicates the current state of the ratio, rather than the current state of the velocity:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

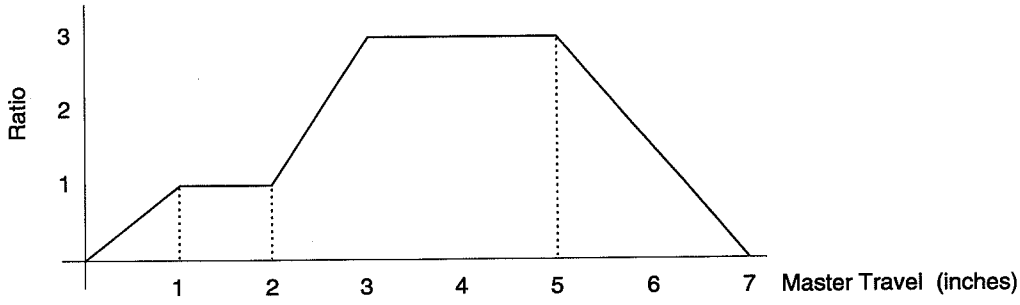
Bit #	Function (YES = 1; NO = Ø)
1	Slave in Ratio Move ..... A Following move is in progress
2	Ratio is Negative ..... The current ratio is negative (i.e., the slave counts are counting in the opposite direction from the master counts).
3	Slave Ratio Changing..... The slave is ramping from one ratio to another (including a ramp to or from zero ratio).
4	Slave At Ratio ..... The slave is at constant non-zero ratio.

Continuous Positioning Mode Moves

For Following moves in the continuous positioning mode (MC1), FOLMD specifies the exact master travel distance over which the slave ratio changes. This will be required for any application that uses multiple ratios and continuous moves for the construction of precisely defined multi-segment moves.

☞ **HINT**: If the slave is in continuous mode (MC1) and the master is starting from rest, setting FOLMD to Ø will ensure precise tracking of the master's acceleration ramp.

In the profile below, the first two moves each change ratio over one master inch, and the final ramp to zero takes place over two master inches.



Example ☞ In the programming example below, assume the slave has a 1000-line encoder on axis 1, connected to a 2-pitch leadscrew. This gives 8000 slaves steps per inch. The master is a toothed belt with a pulley connected to encoder 2, such that there are 800 master steps per inch.

Command	Description
> COMEXC1	Allow commands during motion
> FOLMAS21	Define axis 1 master to be encoder input #2
> FOLEN1	Enable Following on axis 1 (will follow encoder #2)
> SCALE1	Enable scaling
> SCLD8ØØØ	Set axis 1 scaling so that slave commands are in inches
> SCLMAS8ØØ	Set axis 1 scaling so that master commands are in inches
> FOLMD1	Define slave to change ratio over 1 inch of the master travel
> FOLRD1	Set Following ratio denominator to 1 for all subsequent ratios
> D+	Set direction positive
> MC1	Mode set to continuous clockwise moves
> FOLRN1	Set Following ratio numerator to 1, resulting in a 1 to 1 ratio
> FMCNEW1	Restart master cycle counting
> GO1	Start axis 1 from rest to reach velocity of master (encoder #2)
This command is for steppers only → > WAIT(1FS.4=B1)	Wait until slave is at ratio
> FOLRN3	Set Following ratio numerator to 3, resulting in a 3 to 1 ratio
> GOWHEN (1PMAS>=2)	Enable motion pre-processing so that the next ramp begins at master position 2
> GO1	Slave starts ratio change to 3 to 1 at master position 2
> FOLRNØ	Set Following ratio numerator to zero (ratio is Ø to 1)
> FOLMD2	Define slave to change ratio over 2 inches of master travel
> WAIT(1AS.26=bØ)	Wait until the previous ramp is started (GOWHEN bit cleared)
> WAIT(1FS.3=bØ)	Wait until the previous ramp is finished
> GOWHEN(1PMAS>=5)	Enable motion pre-processing so that slave motion begins at master position 5
> GO1	Wait for master to reach 5 revolutions before slave starts ratio change to 0 (zero)

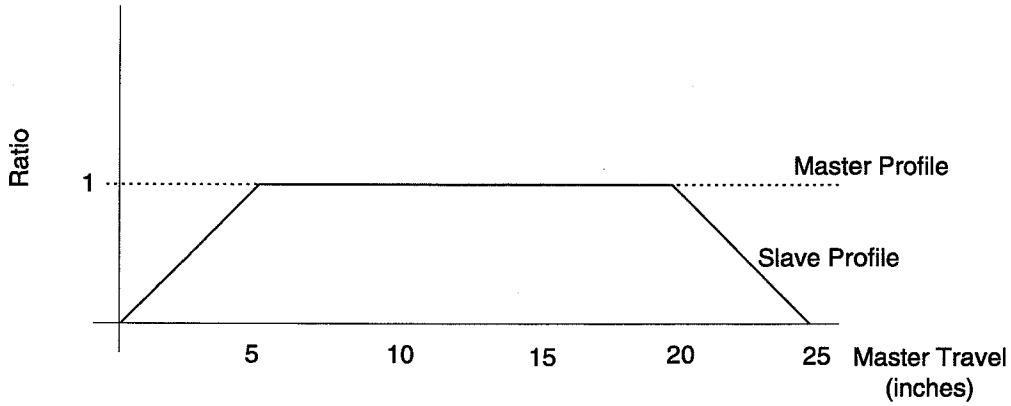
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Preset Positioning Mode Moves

For preset positioning mode (MCØ) moves, the FOLMD parameter is the master distance over which the entire slave move is to take place.

As an example, a slave is to move 20 inches over a master distance of 25 inches with a maximum ratio of 1:1 (ratio set with the FOLRN1 and FOLRD1 commands). The program and a diagram of the move profiles are provided below.

Command	Description
> FOLMAS31	Define axes 1 master to be encoder input port 3
> FOLMD25	Define slave to perform the move over 25 inches of the master (encoder 3)
> MCØ	Set positioning mode to preset
> MAØ	Set preset positioning mode to incremental
> D2Ø	Set slave distance to 20 inches
> FOLRN1	Set Following ratio numerator to 1
> FOLRD1	Set Following ratio denominator to 1 (ratio is 1:1)
> GO1	Perform the slave move



If the master distance specified is too large for the slave distance and ratio (FOLRN and FOLRD) commanded, the slave will never actually reach the commanded ratio, and the move profile will look similar to that below. Here, the FOLMD is 25 inches and the slave is commanded to move 10 inches:



If the master distance is too small for the slave distance and ratio commanded, the 6000 controller will not perform the move at all. For example, if the FOLMD is 25, the ratio is 1:1, and the slave is commanded to move 30 inches, the move will not even be attempted. The error message "INVALID DATA" will be displayed (depending on the ERRLVL setting) and program execution will continue.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Performing Phase Shifts

Following Status (TFS and FS) bits 9-12 indicate the shift status of the slave. Shifting is super-imposed motion, but if viewed alone, can have its own status. In other words, bits 10-12 describe only the shifting portion of motion.

Bit #	Function (YES = 1; NO = Ø)
9	OK to Shift ..... Conditions are valid to issue shift commands (FSHFD or FSHFC).
10	Shifting now..... A shift move is in progress.
11	Shift is Continuous..... An FSHFC-based shift move is in progress.
12	Shift Dir is Neg ..... The direction of the shift move in progress is negative.

When a slave is following a master continuously, it may be necessary to adjust, or *shift*, the Following *phase* (slave's position with respect to the master) independent of motion due to ratio moves. The FSHFC and FSHFD commands allow time-based slave moves to be superimposed upon ratio Following moves. Because phase shifts are time-based, they are independent of master motion; in fact, the master may be at rest and a shift may still be performed.

Use the FSHFD command to perform a preset shift move with a specific change in slave phase. The FSHFD distance value will be scaled by SCLD if scaling is enabled (SCALE1).

Use the FSHFC command to superimpose a continuous shift move in the positive (FSHFC1) or negative (FSHFC2) direction. The FSHFC parameters stop (Ø) and kill (3) can be used to halt a continuous FSHFC move or a preset FSHFD move without affect the ratio motion.

The most recently defined velocity and acceleration (i.e., the V and A values) for the slave will determine the basis for the superimposed shift move profile for both FSHFC and FSHFD moves. The commanded velocity of the FSHFC or FSHFD move will be added to the current velocity at which the slave is performing the Following move. For example, assume a slave is traveling at 1 rps in the positive direction as a result of following a master. If a FSHFC move is commanded in the positive direction at 2 rps, the slave's actual velocity (after acceleration) will be 3 rps.

For servos, shifting may be performed whenever Following is enabled (FOLEN1). For steppers, this may only be done while Following is enabled and the slave is either not in a move, or is in continuous positioning mode (MC1) and moving at constant ratio. For both products, TFS/FS bit 9 indicates when a shift is allowed.

The current slave position (TPSLV value) and the net slave shift accumulated since the most recent FOLEN1 command (TPSHF value) may be read into numeric variables (VAR) using the PSLV and PSHF commands, respectively (e.g., VAR6=2PSLV). They may also be used for subsequent decision making (e.g., IF(3PSHF<6), GOWHEN(1PSLV>VAR2), etc.).

The TPSHF and PSHF values are set to zero each time the Following mode is enabled (FOLEN1), even if the slave is already in Following mode. This provides a way of clearing these values for programming convenience.

Note that the distance traveled during the time-based deceleration due to stop, kill, or limits is included in the PSHF value. By comparing "before and after" values of PSHF, a 6000 program may calculate how much shift was required to perform visual- or sensor-based alignment of a master/slave phase relationship.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Examples

An FSHFC or FSHFD move may be needed to adjust the slave position *on the fly* because of a load condition which changes during the continuous Following move. Below are programming examples to demonstrate both shift methods.

FSHFC Example

An operator is visually inspecting the slave's continuous Following motion with respect to the master. If he notices that the master and slave are out of synchronization, it may be desirable to have an interrupt programmed (e.g., activated by pressing a pushbutton switch) that will allow the operator to move the slave at a superimposed correction speed until the operator chooses to have the slave start tracking the master again (by releasing the pushbutton). The programming example below illustrates this.

Assume all scale factors and set-up parameters have been entered for the master and slave. In this example, the slave (axis #1) is continually following the master at a 1:1 ratio. If the operator notices some mis-alignment between master and slave, he can press 1 of 2 pushbuttons (connected to programmable inputs #1 and #2) to shift the slave in the positive or negative direction until the button is released. After the adjustment, the program continues on as before.

	Command	**Description**
	> DEF SHIFT	Begin definition of program called SHIFT
	- COMEXS1	Continue command execution after stop
	- COMEXC1	Continue command execution during motion
	- FOLMAS21	Axis 2 encoder input is the master for axis 1
	- FOLRN1	Set slave-to-master Following ratio numerator to 1
	- FOLRD1	Set slave-to-master Following ratio denominator to 1 — ratio set to 1:1
	- FOLEN1	Enable Following mode on axis #1
	- A25	Set acceleration
	- AD18	Set deceleration
	- V5	Set velocity
	- D+	Set direction to positive
	- MC1	Select continuous positioning mode
	- GO1	Start following master continuously
	- VARB1=b1Ø	Define input pattern #1 and assign to VARB
	- VARB2=bØ1	Define input pattern #2 and assign to VARB
	- $TESTIN	Define label called TESTIN
	- IF(IN=VARB1)	IF statement (if input #1 is activated, do the jump)
	- JUMP SHIFTP	Jump to shift slave in the positive direction when pattern 1 active
	- NIF	End of IF statement
	- IF(IN=VARB2)	IF statement (if input #2 is activated, do the jump)
	- JUMP SHIFTN	Jump to shift slave in the negative direction when pattern 2 active
	- NIF	End of IF statement
	- JUMP TESTIN	Return to main program loop
	- $SHIFTP	Define label called SHIFTP (subroutine to shift in the positive direction)
	- FSHFC1	Start continuous slave shift move in positive direction
	- WAIT(IN.1=BØ)	Continue shift until input bit #1 is deactivated
	- FSHFCØ	Stop shift move
This line is for steppers only →	- WAIT(1FS.1Ø=BØ)	Wait until the shift is completed (steppers only)
	- JUMP TESTIN	Return to main program loop
	- $SHIFTN	Define label called SHIFTN (subroutine to shift in the negative direction)
	- FSHFC2	Start continuous slave shift move in the negative direction
	- WAIT(IN.2=BØ)	Continue shift until bit #2 is deactivated
	- FSHFCØ	Stop shift move
This line is for steppers only →	- WAIT(1FS.1Ø=BØ)	Wait until the shift is completed (steppers only)
	- JUMP TESTIN	Return to main program loop
	- END	End definition of program called SHIFT

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FSHFD Example

In this example, the slave follows a master that moves in a continuous cycle. Once each cycle, the master and slave both pick parts. The master's part is detected by a sensor connector to trigger B, and the slave's part is detected by a sensor connected to trigger A. After both parts are detected, they must be aligned. The sensors are mounted 2 inches apart from each other, so that proper alignment would result in 2 inches of slave travel between detection of the master's part and detection of the slave's part.

The slave position is sampled when each of the sensors activates. The difference between the slave positions is compared to the required 2 inches. If the measured difference is greater than or less than 2 inches, then a shift move to correct the alignment is made. At that point, the slave will then start tracking the master again. The slave (axis #1) is continually following the master at a 1:1 ratio.

	Command	Description
	> DEF ALIGN	Begin definition of program called ALIGN
	- COMEXC1	Allow continuous command execution during motion
	- INFEN1	Enable input functions
	- FOLMAS31	Axis 3 Encoder input is the master for slave axis 1
	- FOLRN1	Set slave-to-master ratio numerator to 1
	- FOLRD1	Set slave-to-master ratio denominator to 1 (ratio set to 1:1)
	- FOLEN1	Enable Following mode on axis #1
	- MC1	Enable continuous positioning mode
	- D+	Set direction to positive
	- GO1	Start Following master continually
	- INFNC25-H	Enable trigger input A to latch position of slave when the slave's part is detected
	- INFNC26-H	Enable trigger input B to latch position of slave when the master's part is detected
	- $SYNCLP	Main loop where synchronizing moves occur
	- WAIT(SS.25=b1 AND SS.26=b1)	Wait for both slave and master inputs to occur
This line is for steppers only →	- VAR1Ø=1PMCA	Load VAR10 with the slave motor position due to slave input activation
This line is for steppers only →	- VAR11=1PMCB	Load VAR11 with the slave motor position due to master input activation
This line is for servos only →	- VAR1Ø=1PCCA	Load VAR10 with the slave commanded position due to slave input activation
This line is for servos only →	- VAR11=1PCCB	Load VAR11 with the slave commanded position due to master input activation
	- VAR12=VAR1Ø-VAR11	Load VAR12 with the offset distance
	- VAR13=VAR12-2	Calculate the required shift
	- FSHFD(VAR13)	Perform synchronization move of distance in VAR13
	- JUMP SYNCLP	Return (jump) to main program loop
	- END	End of program

Summary of Ratio Following Commands

Command	Description
FOLEN	Enables or disables Following mode
FOLMAS	Defines masters for slave axes
FOLMD	Defines the master distance over which slave acceleration or moves are to take place
FOLRN and FOLRD	Establishes the maximum slave-to-master ratio for a preset move or the final ratio for a continuous move (FOLRN for the numerator and FOLRD for the denominator)
FSHFD	Initiates preset advance or retard (shift) of slave position during continuous Following moves
FSHFC	Initiates continuous advance or retard (shift) of slave position (or kills or stops the shift portion of motion) during continuous Following moves
SCLD	Sets the slave distance scale factor
SCLMAS	Sets the master distance scale factor
TPSHF or [PSHF]	Transfers or assigns the net position shift since constant ratio
TPSLV or [PSLV]	Transfers or assigns the current slave position
TVMAS or [VMAS]	Transfers or assigns the velocity of the master axis

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Electronic Gearbox Application for Ratio Following

An electronic gearbox is a classic application for Ratio Following. Suppose we need a three-output gearbox, with all three outputs geared off the same input. Also, each gear ratio must be individually programmed. In this example, a 1000-line encoder is mounted to the input shaft of a master motor, giving 4000 master counts per revolution after quadrature. This encoder is fed into the encoder input on axis #4 (**ENCODER 4** connector) of the 6000 controller. The motors on axes 1, 2, and 3 have resolutions of 2000, 4000, and 5000 steps/revolution.

In this example, a precise position relationship is not required between master and slaves, but a ratio change during motion is required. The ratio change will take place over one master revolution in order to avoid abrupt acceleration of the slave. The slaves will accelerate to their initial ratios (in terms of revolutions), and after 10 seconds the gear ratio on each axis will change to their final ratios.

In this example, **ENCODER 4** is specified as the master. This means that this *external* master encoder is wired to the 6000 controller's axis #4 encoder input.

Program

Command	Description
> SCALE1	Enable scaling
> SCLD2000,4000,5000	Set slave scale factors equal to the motor resolutions
> SCLA2000,4000,5000	Set acceleration scale factors equal to the motor resolutions
> SCLMAS4000,4000,4000	Master scale factor to number of pulses per rev
> COMEXC1	Allow continuous command execution during motion
> FOLMAS+41,+41,+41	Encoder #4 is master axis for slave axes #1 - #3. The + sign indicates that the master input is not inverted before it is read as master counts.
> FOLEN111	Enable slaves to follow
> FOLMD1	Change to new ratio over one master revolution
> FOLRN1,3,2	Set slave-to-master ratio numerators to 1, 3, and 2
> FOLRD1,1,1	Set all slave-to-master ratio denominators to 1 (initial Following ratio for axis 1 is 1:1, axis 2 is 3:1, and axis 3 is 2:1)
> MC111	Enable continuous mode
> GO111	Begin slave continuous Following move
> TIMST0	Reset and start the timer
> FOLRN10,6,1	Set slave-to-master ratio numerators to 10, 6, and 1
> FOLRD1,7,2	Set slave-to-master ratio denominators to 1, 7 and 2 (change Following ratios: axis 1 is 10:1, axis 2 is 6:7, axis 3 is 1:2)
> WAIT(TIM>=10000)	Wait 10 seconds to change to new ratio
> GO111	Start moving to new ratio

Trackball Application for Ratio Following Motion

A trackball is a two-axis, two-dimensional positioning device; just as a mouse is used to position the cursor on a computer screen, a trackball could be used to position an X-Y stage.

In this example, a two-axis trackball is needed which can do fine and coarse positioning of an X-Y stage. The fine or coarse setting is selected by the user with a two-position switch connected to programmable input #1 on the 6000 controller. Programmable input #2 on the 6000 controller is used to switch back and forth from trackball to standard point-to-point positioning mode. *Unlocking* the stage from the trackball is necessary because of other point-to-point move requirements elsewhere in the 6000 controller program.

The trackball housing has two encoders mounted at 90 degrees to each other which are driven by rubber wheels in contact with the ball. The stage is driven by motors and leadscrews.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For one inch of trackball motion to result in one inch of stage motion, the slave-to-master ratio must be 10-to-1; this will be the ratio for coarse positioning. The fine positioning ratio will be one tenth of that, or 1-to-1. When programmable input #1 is low, coarse positioning is selected, and when programmable input #2 goes low, the stage becomes locked to the trackball. Each change of state of inputs #1 and #2 calls a different subroutine in the 6000 controller program; however, the ratios can only change if the stage is locked to the trackball positioning mode.

The trackball is initially unlocked and fine positioning is selected.

Program

Command	Description
`> SCALE1`	Enable and define scale factors prior to loading program
`> SCLD4ØØØ,4ØØØ`	Slave axes 1 and 2 have 4000 counts per rev resolution post-quadrature
`> SCLV4ØØØ,4ØØØ`	Set velocity scaling factors
`> SCLA4ØØØ,4ØØØ`	Set acceleration scaling factors
`> SCLMAS2ØØ,2ØØ`	Master axis 3 and 4 have 200 counts per rev resolution post-quadrature
`> DEL UNLOCK`	
`> DEF UNLOCK`	Program that unlocks the stage from the trackball
`- S11`	Stop moves
This line is for steppers only → `- WAIT(1AS.1=BØ AND 2AS.1=BØ`	Wait for motion to stop on both axes
`- ONIN.26-1`	Set up trigger B to lock trackball to stage
`- FOLENØØ`	Stop Following mode
`- ONP LOCK`	Select LOCK as an ON program
`- JUMP WAITLP`	Return to main loop
`- END`	End program definition
`> DEL LOCK`	
`> DEF LOCK`	Program that locks the stage to the trackball
`- FOLEN11`	Enable Following on both axis
`- IF(VAR1=Ø)`	If in the FINE mode, set ratio to 0.5:1
`- FOLRN.5,.5`	
`- ELSE`	
`- FOLRN1.5,1.5`	If in the COARSE mode, set ratio to 1.5:1
`- NIF`	
`- ONIN.26-Ø`	Set up trigger B to unlock trackball from stage
`- GO11`	Start following the trackball
`- ONP UNLOCK`	Select UNLOCK as ON program
`- JUMP WAITLP`	Return to main loop
`- END`	End program definition
`> DEL COARSE`	
`> DEF COARSE`	Declare COARSE label
`- FOLRN1.5,1.5`	Coarse positioning ratio of 1.5 to 1
`- GO11`	Move to begin travel at new ratio
`- VAR1=1`	Flag to indicate we are in coarse mode
`- END`	Return to main loop
`> DEL FINE`	
`> DEF FINE`	Subroutine to assign fine positioning
`- FOLRN.5,.5`	Fine positioning ratio is 0.5:1
`- GO11`	Move to begin travel at new ratio
`- VAR1=Ø`	Flag to indicate we are in fine mode
`- END`	Return to main loop
`> DEL TRACK`	
`> DEF TRACK`	Main track ball program
`- IF(IN.25=b1 AND VAR1=Ø)`	If trigger input A is set to 1 and we are currently in fine mode, then enter coarse mode
`- GOSUB COARSE`	Set high ratio
`- NIF`	

(Continued)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Trackball Program *(Continued)*

```
- IF(IN.25=bØ AND VAR1=1)        If trigger input A is set to 0 and we are currently in coarse
                                 mode, then enter fine mode
- GOSUB FINE                     Set low ratio
- NIF
- IF(LIM.1=bØ OR LIM.2=bØ)       If a limit is hit, allow track ball to move off
- GO1
- NIF
- IF(LIM.4=bØ OR LIM.5=bØ)
- GOX1
- NIF
- END

> DEL MAIN
> DEF MAIN                       Begin definition of main program
- V1,1                           Set non-Following move parameters
- A99,99
- LH3,3,Ø,Ø
- FOLMAS+31,+41                  Encoder #3 is master axis for slave axes #1 and Encoder
                                 #4 is master axis for slave axes #2.  The slave axes will
                                 move in the same direction as master.
- FOLRN.5,.5
- INDEB25,25Ø                    Noisy switch debounce of 250 milliseconds
- INDEB26,25Ø
- FOLRD1,1                       Initial slave-to-master ratio is set to fine positioning (0.5:1)
- VAR1=Ø                         Flag set to fine positioning
- MC11                           Set both axes 1 and 2 to continuous positioning mode
- FOLENØØ                        Following is initially disabled
- COMEXC1                        Continue command execution during motion.
- COMEXS1                        Continue command execution after stop
- COMEXL11                       Continue command execution after a limit is hit
- INFEN1                         Enable input functions
- SGP2Ø,2Ø                       Set servo gains
- SGV5,5
- DRIVE11ØØ                      Enable drives
- DRFLVL11XX                     Set drive fault level for Compumotor 670-T drives
- ONP LOCK                       Select LOCK as an ON program
- ONIN.26-1                      Trigger B locks trackball to stage
- ONCOND1ØØØ                     Inputs enabled for interrupts
- $WAITLP                        Main program loop
- IF(IN.26=b1)                   If trigger input B is set to 1 (stage locked), enter trackball
                                 mode
- GOSUB TRACK
- NIF                            End of IF statement
```

Other user programs can be added here for performing motion when the stage is not locked to the trackball.

```
- JUMP WAITLP                    Return to main loop
- END                            End program definition
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ③

Master Cycle Concept

Overview

Ratio Following can also address applications that require precise programming synchronization between moves and I/O control based on master positions or external conditions. The concept of the master cycle greatly simplifies the required synchronization.

A master cycle is simply an amount of master travel over which one or more related slave events take place. The distance traveled by the master in a master cycle is called the *master cycle length*. A *master cycle position* is the master position relative to the start of the current master cycle. The value of master cycle position increases as positive-direction *master cycle counts* are received, until it reaches the value specified for master cycle length. At that point, the master cycle position becomes zero, and the *master cycle number* is incremented by one—this condition is called *rollover*.

The master cycle concept is analogous to minutes and hours on a clock. If the master cycle is considered an hour, then the master cycle length is 60 minutes. The number of minutes past the hour is the master cycle position, and current hour is the master cycle number. In this analogy, the master cycle position decrements from 59 to zero as the hour increases by one.

By specifying a master cycle length, periodic actions may be programmed in a loop or with subroutines which refer to cycle positions, even though the master may be running continuously. To accommodate applications where the feed of the product is random, the start of the master cycle may be defined with trigger inputs. Two types of *waits* are also programmable to allow suspension of program operation or slave moves based on master positions or external conditions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Following Status (TFS and FS) bits 13-16 indicate the status of master cycle counting. If a following application is taking advantage of master cycle counting, these bits provide a quick summary of some important master cycle information:

Bit #	Function (YES = 1; NO = ø)
13	Master Cyc Trig Pend .....A master cycle restart is pending the occurrence of the specified trigger.
14	Mas Cyc Len Given .....A non-zero master cycle length has been specified with the FMCLEN command.
15	Master Cyc Pos Neg .....The current master cycle position (PMAS) is negative. This could be by caused by a negative initial master cycle position (FMCP), or if the master is moving in the negative direction.
16	Master Cyc Num > 0.....The master position (PMAS) has exceeded the master cycle length (FMCLEN) at least once, causing the master cycle number (NMCY) to increment.

Master Cycle Length (FMCLEN)

The FMCLEN command is used to define the length of the master cycle. The value entered with this command is scaled by the SCLMAS parameter to allow specification of the master cycle length in user units. This parameter must be defined before those commands which wait for periodically repeating master positions are executed.

The default value of FMCLEN is zero, which means the master cycle length is practically infinite (i.e., 4,294,967,246 steps, after scaling). If a value of zero is chosen, the master cycle position will keep increasing until this very high value is exceeded or a new cycle is defined with the FMCNEW command (or triggered after a TRGFNcx1 command) described below. If a non-zero value for FMCLEN is chosen, the internally maintained master cycle position will keep increasing until it reaches the value of FMCLEN. At this point, it immediately rolls over to zero and continues to count.

The master cycle length may be changed with the FMCLEN command even after a master cycle has been started. The new master cycle length takes affect as soon as it is issued. If the new master cycle length is greater than the current master cycle position, the cycle position will not change, but will rollover when the new master cycle length is reached. If the new master cycle length is less than the current master cycle position, the new master cycle position becomes equal to the old cycle position minus one or more multiples of the new cycle length.

Example

Command	Description
`> FMCLEN23,1ø,12,22`	Set master cycle length for all four axes— (axis 1: 23 units, axis 2: 10 units, axis 3: 12 units, and axis 4: 22 units)

Restart Master Cycle Counting (FMCNEW or TRGFNcx1)

Once the length of the master cycle has been specified with the FMCLEN command, master cycle counting may be restarted immediately with the FMCNEW command, or based on activating a trigger input as specified with the TRGFNcx1 command. The new master cycle count is started at an initial position specified with the FMCP command (see below).

When the TRGFNcx1 command is used, the restart of master cycle counting is pending activation of the specified trigger. If an FMCNEW command is issued while waiting for the specified trigger to activate, counting is restarted immediately with the FMCNEW command, and the TRGFNcx1 command is canceled.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When using TRGFNcx1

- Before the TRGFN command can be used, you must first assign the trigger interrupt function to the specified trigger input with the INFNCi-H command, where "i" is the input number of the trigger input desired for the function (input bit assignments vary by product).
- Because the 6000 controller program will not wait for the trigger to occur before continuing on with normal program execution, a WAIT or GOWHEN condition based on PMAS will not evaluate true if the restart of master cycle counting is pending the activation of a trigger. To halt program operation, the WAIT command can be used.

A new master cycle will restart automatically when the total master cycle length (FMCLEN value) is reached. This is useful in continuous feed applications.

Example

Command	Description
> FMCNEW11xx	Restart new master cycle counting on axes 1 and 2
> INFNC26-H	Assign input #26 (trigger B) the *trigger interrupt* function **(prerequisite to using the TRGFN features)**
> TRGFNBx1xxxxxx x1	When trigger input B (**TRG-B**) goes active, restart new master cycle counting on axes 1 and 2

Initial Master Cycle Position (FMCP)

The FMCP command allows you to assign **for the first cycle only**, an initial master cycle position to be a value other than zero. When master cycle counting is restarted with the FMCNEW command or with the trigger specified in the TRGFNcx1 command, the master cycle position takes the initial value previously specified with the FMCP command. The value for FMCP is scaled by SCLMAS if scaling is enabled (SCALE1)

FMCP was designed to accommodate situations in which the trigger that restarts master cycle counting occurs either before the desired cycle start, or somewhere in the middle of what is to be the first cycle. In the former case, the FMCP value must be negative. The master cycle position is initialized with that value, and will increase right through zero until it reaches the master cycle length (FMCLEN). At that point, it will roll over to zero as usual.

The continuous cut-to-length example below illustrates the use of a negative FMCP (a trigger that senses the motion of the master is physically offset from the master position at which some action must take place). If it is desired that the first cycle is defined as already partially complete when master cycle counting is restarted, the FMCP value must be greater than zero, but less than the master cycle length.

To give a value for FMCP which is greater than master cycle length is meaningless since master cycle positions are always less than the master cycle length. The 6000 controller responds to this case as soon as a new master cycle counting begins by using zero instead of the initial value specified with FMCP.

Transfer and Assignment/Comparison of Master Cycle Position and Number

The current master cycle position and the current master cycle number may be transferred/displayed with the TPMAS and TNMCY commands, respectively. These values may also be read into numeric variables (VAR) at any time using the PMAS and NMCY commands (e.g., VAR6=NMCY).

Very often, the master cycle number will be directly related to the quantity of product produced in a manufacturing run, and the master cycle position can be used to determine what portion of a current cycle is complete.

Position Sampling Period
Steppers – 2 milliseconds.
Servos – *system update period* (depends on SSFR & INDAX command values – see SSFR command).

The master cycle number is sampled once per *position sampling period* (see note, left). If the master cycle length (FMCLEN) divided by the master's velocity (VMAS) is less than the position sampling period, then the sample (TNMCY or NMCY value) may not be accurate.

Details on using PMAS in conditional expressions is provided below in *Using Conditional Statements with PMAS*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Conditional Statements with Master Cycle Position (PMAS)

The current master cycle position (PMAS) value may be used in comparison expressions, just like other position variables such as PC, PM, PE, and FB. PMAS is a special case, however, because its value rolls over to zero when the master cycle length (FMCLEN) is met or exceeded. This means that PMAS values greater than or equal to the master cycle length will never be reported with the TPMAS command, or with expressions such as (VAR1=1PMAS).

The other fact that makes PMAS special is that master cycle counting may be restarted after the command containing the PMAS expression has been executed. Either the FMCNEW command or the TRGFNcx1 command may be used to restart counting, each with a different effect on the evaluation of PMAS.

The treatment of PMAS in comparison expressions depends on the command using the expression, as described below. WAIT and GOWHEN are treated as special cases.

IF, UNTIL, and WHILE

These commands evaluate the current value of PMAS in the same way that TPMAS does (i.e., PMAS values will never be greater than or equal to the master cycle length). With these commands, avoid comparing PMAS to be greater than or equal to variables or constants which are nearly equal to the master cycle length, because rollover may occur before a PMAS sample is read which makes the comparison true. If such a comparison is necessary, it should be combined (using OR) with a comparison for master cycle number (NMCY) being greater than the current master cycle number.

Also, master cycle counting restart may be pending activation of a trigger, but this will not affect the evaluation of PMAS for IF, WAIT, and WHILE. It is simply evaluated based on counting currently underway.

WAIT and GOWHEN

These commands evaluate the current value of PMAS differently than TPMAS does, in such a way that it is possible to compare PMAS to variables or constants which are greater than or equal to the master cycle length and still have the comparison be reliably detected.

Effectively, PMAS is evaluated as if the master cycle length were suddenly set to its maximum value (2^{32}) at the time the WAIT or GOWHEN command is encountered. It eliminates the need to OR the PMAS comparison with a comparison for master cycle number (NMCY) being greater than the current master cycle number. Such multiple expressions are not allowed in the GOWHEN command, so this alternative evaluation of PMAS offers the required flexibility.

This method of evaluation of PMAS allows commands which sequence slave events through a master cycle to be placed in a loop. The WAIT or GOWHEN command at the top of the loop can execute, even though the actual master travel has not finished the previous cycle. If it is desired to WAIT or GOWHEN for a master cycle position of the next master cycle, the variable or constant specified in the command should be calculated by adding one master cycle length to the desired master cycle position.

Finally, master cycle counting restart may be pending activation of a trigger (TRGFNcx1), and this will suspend the evaluation of PMAS for these commands. PMAS is not sampled, and the comparison evaluates as false. During this time, if the pending status of master cycle counting restart is aborted with FMCNEWØ, the GOWHEN condition is also cleared, and any motion profile of any axis waiting on *that* PMAS comparison will be canceled. Otherwise, when master cycle counting is restarted by a trigger, evaluation takes place as described above. This allows GOWHEN to include waiting on a trigger without explicitly including it in the GOWHEN expression.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Synchronizing Following Moves with Master Positions

A final special case allows perfect synchronization between the start of a Following motion profile of a slave axis and a specified position of its master. If a GOWHEN(nPMAS >= xxx) expression is used to synchronize a slave with its own master, with the operator specifically ">=", a special synchronization occurs. Although it may be impossible for the 6000 product to sample the exact master position specified, the Following motion profile is calculated from master travel based on that position. This allows for the construction of profiles in which the synchronization of master and slave positions is well defined and precisely maintained. This feature requires positive travel of the master, which can be achieved with the appropriate sign for the FOLMAS specification.

Summary of Master Cycle and Wait Commands

Command	Description
FMCLEN	Defines the length of the master cycle
FMCNEW	Immediately restart master cycle counting
FMCP	Defines the initial position of a new master cycle
GOWHEN( )	GOWHEN( ) suspends execution of the next move on the specified axis or axes until the specified conditional statement (based on FB, NMCY, PC, PE, PM, PMAS, PSLV, or PSHF) is true
TNMCY or [NMCY]	Transfers or assigns the current master cycle number
TPMAS or [PMAS]	Transfers or assigns the current master cycle position
TRGFN	TRGFNc1x initiates a GOWHEN, suspending execution of the next slave move until the specified trigger input (c) goes active. TRGFNcx1 causes master cycle counting to restart when the specified trigger input (c) goes active.
WAIT( )	WAIT( ) suspends program execution until the specified conditional statement (based on PMAS, FS, NMCY, PSHF, PSLV, or VMAS) is true; WAIT(SS.i=b1) suspends program execution until a trigger input is activated. The "i" is the programmable input bit number corresponding to the trigger input. (Input bit assignments vary by product; refer to the Programmable I/O Bit Patterns table in the Programming Guide section of the *6000 Series Software Reference Guide* to determine the correct bit pattern for your product.)
AS and TAS bit #26	AS and TAS bit #26 is set when there is a profile suspended pending GOWHEN condition, initiated either by a GOWHEN command or a TRGFNc1x command; this bit is cleared when the GOWHEN condition is true or when a stop or kill command is issued.
ER and TER bit #14	ER and TER bit #14 is set if the GOWHEN condition is already true when the GO, GOL, FSHFC, or FSHFD command is given (ERROR bit #14 must first be enabled to check for this condition)

NOTE

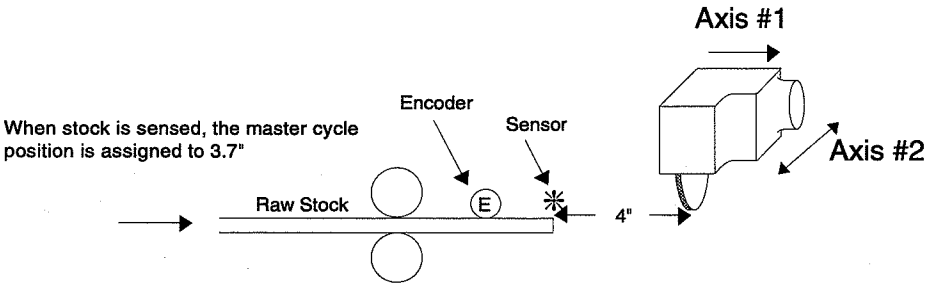
The continuous cut-to-length application example below illustrates the use of the master cycle concept and the commands above.

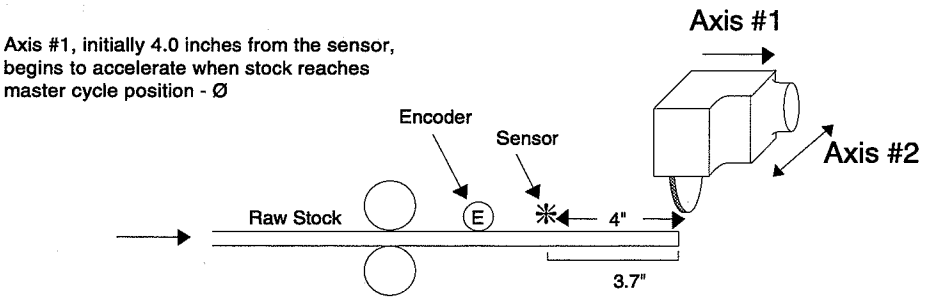
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

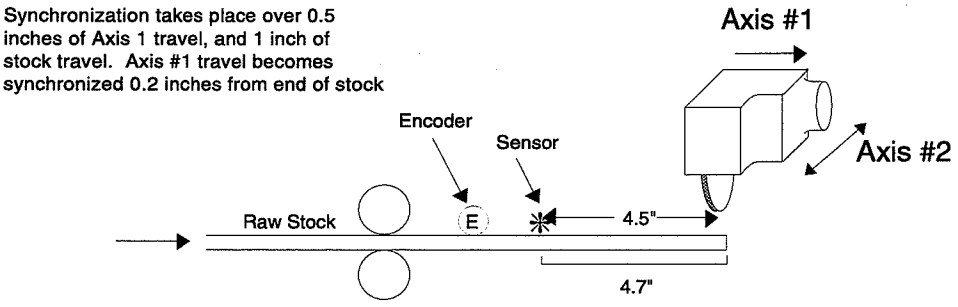
Continuous Cut-to-Length Application

This application requires automobile trim to be cut to a pre-defined length. The saw is controlled by axes #1 and #2 on the 6000 controller. It must be moving with the material while the cut is being made (axis #1), and also move perpendicular to the trim (axis #2) to actually make the cut. The trim comes in long stock which moves continuously under the cutting area.

The leading edge of the trim stock is detected with a sensor connected to trigger #1 which is located 4 inches from the home position of the saw. Axis #1 will be following the trim based on an encoder mounted on the trim via a friction wheel. The encoder is a 1000-line encoder and the wheel is geared to give 2 revolutions per inch of trim, resulting in 8,000 post-quadrature steps per inch of trim. Axis #1 has a resolution of 4,000 steps per rev and is connected to a 2-pitch leadscrew 24 inches in length. Axis #2 is similar in mechanics but its length is 10 inches. The travel on Axis #1 will be controlled by the speed at which axis #2 makes its cut. The travel on axis #2 is a fixed speed of 5 inches per second, with a fixed cross stroke of 5 inches. Limit switches are in place for safety.







Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The master cycle length will be set equal to the desired cut length (36" in the example below), which the operator can change by modifying variable VAR1. The cut cycle will be a continuous loop, but the first cut will be made 0.2 inches from the end of the stock to ensure an even first edge. Axis #1 will accelerate to the desired tracking ratio over 1 inch of master travel for all cuts. Assume that the home position of both axes is at position 0 inches.

The Cut-to-Length example takes advantage of being able to change master cycle length, while being careful to change it only at the beginning of a current cycle. This ensures that the current master cycle position will be less than the new master cycle length, and will not change as a result of a change in cycle length. In this example, the master cycle length and corresponding waits are redefined every cycle to the current value of VAR1. The value of VAR1 becomes the cut length, and can be changed via remote command during program execution. With minor modifications, the cut lengths and the number of iterations could be read from DATA commands (e.g., in a teach mode application).

Program

Command	Description
> SCALE1	Enable scaling
> SCLD8ØØØ,8ØØØ	Set axes #1 & #2 scale factors for programming in inches
> SCLV,8ØØØ	Axis #2 velocity scale factor for inches/sec
> SCLA,8ØØØ	Axis #2 accel scale factor for inches/sec/sec
> SCLMAS8ØØØ	Master scale factor for programming in inches
> DEF CUTLEN	Start definition of Cut-to-Length program
- COMEXC1	Enable continuous command execution mode
- COMEXS1	Continue execution if stop is issued
- INFEN1	Enable input functions
- INFNC25-H	Enable trigger input A for TRGFN use
- OUTFEN1	Enable output functions
- A2Ø,2Ø	Set acceleration
- V5,5	Set velocity
- MA11	Absolute positioning mode for non-Following moves
- VAR1=36	Desired cut length is 36"
- VAR2=4	Sensor is 4" from home position of axis #1
- VAR2=VAR2+Ø.2	1st cut to be 0.2" from end of stock
- VAR3=1	FOLMD to be set to 1"
- VAR4=VAR3/2	Slave will travel 1/2" when accelerating to 1:1 ratio while master travels 1"
- VAR2=VAR4-VAR2	Take distance slave travels during accel into account so we'll be up to speed at position = 0.2" from end of stock. Then the cut will be made. Initial master cycle position will be the negative of the distance traveled during slave wait and accel.
- FOLMAS31	Encoder input #3 is the master for slave axis #1
- FMCP(VAR2)	Set initial master cycle position to wait length
- FOLMD(VAR3)	Acceleration to constant ratio will take place over VAR3" (1" here) of master travel
- FOLRN1	
- FOLRD1	Following ratio is 1 to 1
- TRGFNA X1	Define a new master cycle on trigger input #1 (TRG-A)
- WAIT(IN.25=b1)	Suspend program execution until stock is sensed
- OUT.6-1	Turn on output for saw blade to move down into position
- GOWHEN(1PMAS>=Ø)	Wait on first move for start of master cycle. This will ensure being at 1:1 ratio at exactly 0.2" from end of stock.
- $NEWCUT	Subroutine DEF for continuous cutting
- D,5	Axis #2 will move 5 inches across the stock
- MC1	Axis 1 into continuous move mode.
- FOLEN1	Enable Following on axis #1
- VAR5=VAR1	Set VAR5 = VAR1 (Snapshot of VAR1)
- FMCLEN(VAR5)	New master cycle length is cut length
- GO1	Start Following move on slave axis #1

(Continued)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Continuous Cut-to-Length Program *(Continued)*

- WAIT(1FS.4=b1)	Wait for axis #1 to be in sync with the moving stock
- GOx1	Once axis #1 is up to speed, move axis #2 across stock to make the cut
- WAIT(2AS.1=bØ)	Wait for axis #2 cut to finish
- S1	Stop Following move on axis #1
- COMEXCØ	Disable continuous command execution mode
- FOLENØ	Exit Following mode
- OUT.6-Ø	Raise the saw blade
- MCØ	Axis 1 into preset move mode.
- DØ,Ø	Move both axes back to home positions
- GO11	Execute moves on axes 1 and 2 (execution will not occur until motion from previous move is complete)
- COMEXC1	Enable continuous command execution mode
- OUT.6-1	Move saw blade into position for next cut
- GOWHEN(1PMAS>=VAR5)	Synchronize next move with next master cycle
- GOTO NEWCUT	Repeat the cut cycle
- END	End program definition

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ④

Technical Considerations for Following

Overview

In the Technical Overview section at the start of this document (see page 1), the algorithm for 6000 controller Following was briefly discussed. Here we will address some of the more technical aspects of Following:

- Master Position Prediction
- Master Position Filtering
- Following error
- Maximum acceleration and velocity **(steppers only)**
- Dynamic position maintenance **(steppers only)**
- Factors affecting Following accuracy
- Preset vs. Continuous Following moves
- Master and slave distance calculations
- Using other features with Following

Keep in mind that in all cases, the slave position is calculated from a sampled master position.

Following Status (`TFS` and `FS`) bits 17 and 18 indicate the status of master position measurement features:

Bit #	Function (YES =1; NO = ø)
17	Mas Pos Prediction On ..... Master position prediction has been enabled (`FPPEN`).
18	Mas Filtering On..... A non-zero value for master position filtering (`FFILT`) is in effect.

Master Position Prediction

Master Position Prediction is a technique used to compensate for the fact a slave's position command cannot be calculated and implemented infinitely fast.

The master position prediction mode is enabled by default (`FPPEN1`) in the Following algorithm, but can be turned off as desired with the `FPPENØ` command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Position Sampling Period
Steppers – 2 milliseconds.
Servos – *system update period* (depends on SSFR and INDAX command values – see SSFR command description).

The 6000 controller measures master position once per *position sampling period*, and calculates a corresponding slave position command. This calculation and achieving the subsequent slave commanded position requires 2 sample periods.

If master position prediction mode is disabled (FPPENØ), waiting 2 sample periods results in a slave position lag. That is, by the time the slave reaches the position that corresponds to the sampled master position, 2 sample periods have gone by, and the master may be at a new position. Measured in time, the lag is 2 sample periods. Measured in position, the lag is 2 sample periods * current slave velocity.

For example (stepper controller), suppose the slave is traveling at a speed of 25000 counts per second. If master position prediction mode is disabled (FPPENØ), the slave will lag the master by 100 counts (25000 counts/sec * 4 ms = 100 counts).

By measuring the change in master position over sequential sample periods, the master's present velocity is calculated. The present master velocity and position are used to predict future master position. If master position prediction mode is enabled (FPPEN1, the predicted future master position is used to determine the slave's position command. In this case the slave has no velocity-dependent phase delay. The slave's velocity for a given sample will always be the velocity required to move from its current position to the next calculated position command.

If the master motion is fairly smooth and velocity is not very slow, the measurement of its recent velocity will be very accurate, and a good way of predicting future position. But the master motion may be rough, or the measurements may be inaccurate if there is no filtering (see *Master Position Filtering* below). In this case, the predicted master position and the corresponding slave position command will have some error, which may vary in sign and magnitude from one sample to the next. This random variation in slave position command error results in rough motion. The problem is particularly pronounced if there is vibration on the master.

It may be desirable to disable the master position prediction mode (FPPENØ) when maximum slave smoothness is important and minor phase delays can be accommodated.

If master filtering is enabled (FFILT≠Ø), then the prediction algorithm would be used on the filtered master position, resulting in a smoother slave position command. However, due to the delay introduced by the filtering, the prediction algorithm would not compensate for the total delay in the slave's tracking command. (See also *Master Position Filtering* below.)

Master Position Filtering

Position Sample Period
Steppers – 2 milliseconds.
Servos – *system update period* (depends on SSFR and INDAX command values – see SSFR command description).

The slave axis' position command is calculated at each *position sample period*. This calculation is a function of the master position and the master velocity estimated from the change in master position over 2 position sample periods.

The *Master Position Filter* feature allows you to apply a low-pass filter to the measurement of master position. Master position filtering is used in these situations:

- Measurement of master position is contaminated by either electrical noise (when analog input is the master) or mechanical vibration.
- Measurement noise is minimal, but the motion that occurs on the master input is oscillatory. In this case, using the filter can prevent the oscillatory signal from propagating into the slave axis
  (i.e., ensuring smoother motion on the slave axis).

The bandwidth of the low-pass filter is controlled with the FFILT command:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FFILT Setting	Low pass Filter Bandwidth
Ø	∞ (no filtering) – *default setting*
1	120 Hz
2	80 Hz
3	50 Hz
4	20 Hz

NOTE

Increasing the FFILT command value increases the filtering effect (lowers the bandwidth), but at the expense of increasing the phase tracking error (phase lag) of the slave axis. For more information on phase tracking, refer to *Factors Affecting Following Accuracy* below.

When considering whether or how much master position filtering to use, consider the application requirement itself. The application requirements related to filtering can be categorized into these three types:

Type I: If an application requires smooth motion but also high slave tracking accuracy, then a heavy filtering should **not** be used. It should not be used because it may introduce too much phase lag, although the motion may be smooth. In other words, the master axis in the first place should produce very smooth motion and low sensor measurement noise such that a higher level of master filtering is not needed.

Type II: If slave tracking error is not critical but smooth slave axis motion is desired, then you can use a higher level of master filtering to deal with sensor noise or master vibration problems.

Type III: If it is determined that under certain dynamic conditions the master position's oscillatory measurement is purely caused by its vibration motion (noise is insignificant), and it is necessary for the slave to follow such motion, then the filter command should not be used or only use the highest bandwidth (FFILT1).

Following Error

As soon as an axis becomes configured as a slave, the slave's position command is continuously updated and maintained. At each update, the position command is calculated from the current master position and velocity, and the current ratio or velocity of the slave.

Steppers only: This continuously updated position command is used as the target position for the *Dynamic Position Maintenance* feature, described later in this chapter.

Whenever the commanded position is not equal to the actual slave position, a *Following error* exists. This error, if any, may be positive or negative, depending on both the reason for the error and the direction of slave travel. Following error is defined as the difference between the commanded position and the actual position.

```
Following Error = Commanded position - Actual position
```

If the slave is traveling in the positive direction and the actual position lags the commanded position, the error will be positive. If the slave is traveling in the negative direction and the actual position lags the commanded position, the error will be negative. This error is always monitored, and may be read into a numeric variable (VAR) at any time using the PER command. The error value in slave steps is scaled by SCLD for the axis. This value may be used for subsequent decision making, or simply storing the error corresponding to some other event.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Maximum Velocity and Acceleration (Steppers Only)

The slave's attempt to faithfully follow the master may command velocities and accelerations that the slave axis is physically not able to complete. Therefore, the FMAXV and FMAXA commands are provided to set the maximum velocity and acceleration at which the slave will be allowed to move.

If the slave is commanded to move at rates beyond the defined maximums, the slave will begin falling behind it's commanded position. If this happens, a correction velocity will be applied to correct the position error as soon as the commanded velocity and acceleration fall within the limitation of FMAXV and FMAXA.

The FMAXA and FMAXV commands should be used only to protect against worst case conditions, and should be avoided altogether if they are not needed. If an axis is not able to follow its profile because of limitations imposed by these commands, some correction motion will occur. This is due to the Following error and the resulting *Dynamic Position Maintenance* (see below). If the maximum acceleration (FMAXA value) is set very low, some oscillation about the commanded position may occur because the slave is not allowed to decelerate fast enough to prevent overshoot.

Dynamic Position Maintenance (Steppers Only)

Even while following a master, a slave axis can be in encoder mode with stall detection, position maintenance, and/or deadband wait enabled. In this mode of operation, a difference between actual slave position and the desired slave position may arise just as it may with normal encoder mode moves. Systems with an encoder mounted on a load where mechanical backlash or product stretching is present will be most prone to these desired vs. actual position differences.

An axis becomes a slave by specifying a master with the FOLMAS command. When an axis in encoder step positioning mode (ENC1) becomes a slave axis, it automatically begins the equivalent of position maintenance. This remains true even if the EPMØ (disable position maintenance mode) command is given, and even while the axis is in motion.

There is a very important reason for the continuous dynamic position maintenance. In regular time based moves, the axis has no defined position relationship with anything while it is moving, but it does have a position goal while at rest. Therefore, position maintenance is only meaningful when the axis is not moving. When a slave axis is following a master, there is always a defined position command, calculated from the position of the master. Therefore, position maintenance occurs even while the axis is moving.

The position error each sample period will usually be very small, so only a small correction velocity is added to that required for the slave to follow the master at the commanded ratio. The 6000 controller limits the maximum correction velocity to that specified with the EPMV command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Factors Affecting Following Accuracy

There are additional accuracy requirements of Following applications beyond those of standard positioning. The slave must maintain positioning accuracy while in motion, not just at the end of moves, because it is trying to stay synchronized with the master.

Assuming parameters such as master and slave scaling and ratios have been specified correctly, the overall positioning accuracy for an application depends on several factors:

- Resolution of the master
- Resolution of the slave
- Position sampling accuracy
- Accuracy of the slave motor and drive
- Accuracy of load mechanics
- Master position prediction
- Master velocity relative to master position prediction & master position filtering
- Tuning (servos only)
- Dynamic position maintenance (steppers only)
- Repeatability of the trigger inputs and sensors

Just as with a mechanical arrangement, the accuracy errors can build up with every link from the beginning to the end. The overall worst case accuracy error will be the sum of all the sources of error listed below. The errors fall into two broad categories, namely, master measurement errors and slave errors. These both ultimately affect slave accuracy, because the commanded slave position is based on the measured master position.

It is important to understand how master measurement errors result in slave position errors. In many applications, master and slave units will be the same (e.g., inches, millimeters, degrees). These applications will require linear speeds or surface speeds to be matched (i.e., a 1:1 ratio). For example, suppose that in a rotary knife application, there were 500 master steps per inch of material, so an error in master measurement of one encoder step would result in 0.002 inches of slave position error.

If the master and slave units are not the same, or the ratio is not 1:1, the master error times the ratio of the application gives the slave error. An example would be a rotary master and a linear slave. For instance, suppose one revolution of a wheel gives 4000 master counts, and results in 10 inches of travel on the slave. The ratio is then 10 inches/revolution. The slave error which results from one step of master measurement error is (1/4000) * 10 inches/revolution = 0.0025 inches.

Resolution of the Master

The best case master measurement precision is the inverse of the number of master steps per user's master unit. For example, if there are 100 master steps/inch, then the master measurement precision is 0.01 inches. Even if all other sources of error are eliminated, slave accuracy will only be that which corresponds to 1 step of the master (e.g., 0.01 inches in the previous example).

Resolution of the Slave

The best case slave precision is the inverse of the number of slave steps per user's position unit. For example, if there are 1000 slave steps/inch, then the slave resolution is 0.001 inches. Even if all other sources of error are eliminated, slave positioning accuracy will only be that which corresponds to 1 step of the slave. This must be at least as great as the precision required by the application.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Position Sampling Accuracy

The position sampling rate for the 6000 controller depends on whether it is a servo or a stepper. The sample period for a stepper is 2 ms. The sample period for a servo is the system update period, which is affected by the current SSFR and INDAX command settings (see SSFR description in the *6000 Series Software Reference Guide*).

The repeatability of the sampling rate, from one sample to the next, may vary by as much as 20 µs for servos and 100 µs for steppers. This affect may be eliminated by using non-zero master position filter (FFILT) command values. Otherwise, measurement of master position may be off by as much as (20 to 100 microseconds * master speed). This may appear to be a significant value at high master speeds, but it should be noted that this error changes in value (and usually sign) every sample period. It is effectively like a noise of 200-600 Hz; if the mechanical frequency response of the motor and load is much less than this frequency, the load cannot respond to this error.

Accuracy of the Slave Motor and Drive

The precision also depends on how accurately the drive follows its commanded position while moving. Even if master measurement were perfect, if the drive accuracy is poor, the precision will be poor.

In the case of stepper drives, this amounts to the specified motor/drive accuracy.

In the case of servo drives, the better the drive is tuned for smoothness and zero Following error, the better the precision of the positioning. Often, this really only matters for a specific portion of the profile, so the drive should be tuned for zero Following error at that portion.

Accuracy of the Load Mechanics

The accuracy (not repeatability) of the load mechanics must be added to the overall build up of accuracy error. This includes backlash for applications which involve motion in both directions.

Master Position Prediction

The master position prediction mode may be enabled or disabled with the FPPEN command, but each state contributes a different error.

Disabled (FPPENØ): The slave position command is based on a master position that is 2 sample periods old. This means that master measurement error due to disabling the position prediction mode will be (2 sample periods * master speed).

Enabled (FPPEN1): If the position prediction mode is enabled (default setting), its accuracy is also affected by position sampling accuracy, master speed, and master position filtering. The error due to enabling the position prediction mode is about twice that due to sampling accuracy (i.e., 40 to 200 microseconds * master speed). As with the error due to position sampling accuracy, the error due to the position prediction mode being enabled is like a noise on the order of 200-600 Hz, which is not noticed by large loads.

Master Velocity Relative to Master Position Prediction & Master Position Filtering

Variation in Master Velocity

Although increasing master position filtering (increasing the FFILT command value) eliminates the error due to sampling accuracy, it increases the error due to variations in master speed when the master position prediction mode is enabled (FPPEN1).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Most applications maintain a constant master speed, or change very slowly, so this effect is minimal. But if the master is changing rapidly, there may be a significant master speed measurement error. Because predicted master positions are in part based on master speed measurement, the can result in an error in master position prediction mode (FPPEN1). This effect will always be smaller than that due to the master position prediction mode being disabled (FPPENØ).

Phase Tracking

The cost of using the low pass filter (increasing the FFILT command value) is the increase in phase tracking error of the slave's position command. This is an intrinsic characteristic of a low-pass filter. Increasing the filtering (lowering the bandwidth) increases the phase tracking error, or phase lag.

If the master axis is moving at constant velocity, then the delay in the filtered position in terms of time is $\frac{2}{bw}$ sec , where "bw" is the bandwidth in radians/sec. Therefore, at constant velocity, the tracking error of the filtered master position is *velocity* $*$ $\frac{2}{bw}$.

For example, suppose the master is moving at 4000 counts/sec and the master filter bandwidth is 80 Hz, then the filtered master position delay = $\frac{2}{80 * 2\pi}$ = 3.98 msec, and the slave position command tracking error = 4000 $*$ 0.00398 = 15.915 counts.

One important note is that the slave tracking error discussed here refers to the error of the slave's **position command** calculated from the history of the master position signal. The actual slave tracking error also depends on the tuning and the dynamics of the system. For example, one way to eliminate the tracking error introduced by the low pass filter is to use the velocity feedforward gain (SGVF) in the servo loop.

Tuning **(Servos Only)**

A servo system's tuning has a direct impact on how well the slave can track the master input. Overshoot, lag, oscillation, etc., can be devastating to Following performance. The best tool to use for tuning the 6000 series controller is Motion Architect's Servo Tuner Module. For tuning instructions, refer to the *Tuning* chapter in the servo controller's user guide and to the *Motion Architect User Guide*.

Dynamic Position Maintenance **(Steppers Only)**

Even when the slave is in motor step mode (ENCØ), there may be one slave step of error inherent to the algorithm. When the slave is in encoder step mode, the user specified position maintenance gain is used to correct position. This gain is expressed as motor steps per second per encoder step error, and has a maximum value of 250 (limited internally). If a value lower than this is used, the position error in encoder steps due to low gain is given by: error = (250/gain) * (encoder resolution/motor resolution).

Repeatability of the Trigger Inputs and Sensors

Some applications may use the trigger inputs for functions like registration moves, GOWHENs, or new cycles. For these applications, the repeatability of the trigger inputs and sensors add to the overall position error.

In the 6000 controller, the position capture from the trigger inputs have approximately 100 µs repeatability, and the sensor repeatability (SR) should be determined, too. Velocity $*$ time = distance, so the error due to repeatability is (SR + 0.0001 seconds) $*$ speed = error. If the sensor repeatability is given in terms of distance, that value can be added directly.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Preset vs. Continuous Following Moves

When a slave performs a preset (MCØ) move in Following mode (FOLEN1), the commanded position is either incremental or absolute in nature, but it does have a commanded endpoint. The direction traveled by the slave will be determined by the commanded endpoint position, and the direction the master is counting.

Let's illustrate this with an example. Assume all necessary set-up commands have been previously issued for our slave (axis #1) and master so that distances specified are in revolutions:

Command	Description
> FOLRN3	Set Following slave-to-master ratio numerator to 3
> FOLRD4	Set Following slave-to-master ratio denominator to 4 (ratio is 3 revs on the slave to 4 revs on the master)
> FOLEN1	Enable Following on axis #1
> FOLMD1Ø	Set preset move to take place over 10 master revolutions
> MCØ	Set slave to preset positioning mode
> MA1	Set slave to absolute positioning mode
> PSETØ	Set current absolute position reference to zero
> D5	Set move distance to absolute position 5 revolutions
> GO1	Initiate move to absolute position 5

If the master is stationary when the GO1 command is executed, the slave will remain stationary also. If the master begins to move and master pulses are positive in direction, the slave will begin the preset move in the positive direction. If the master pulses stop arriving before 10 master revolutions have been traveled, the slave will also stop moving, but that GO1 command will not be completed. If the master then starts to count in the negative direction, the slave will follow in the negative direction, but only as far as it's starting position. If the master continues to count negative, the slave will remain stationary. The GO1 command will not actually be completed until the master has traveled at least 10 revolutions in the positive direction from where it was at the time the GO1 command was executed. If the master oscillates back and forth between it's position at the start of the GO1 command to just under 10 revolutions, the slave will oscillate back and forth as well.

The master must be counting in the positive direction for any preset (MCØ) GO1s commanded on the slave to be completed. If mechanics of the system dictate that the count on the source of the master pulses is negative, a minus (-) sign should be entered in the FOLMAS command so that 6000 controller sees the master counts as positive.

Continuous slave moves (MC1) react much differently to master pulse direction. Whereas a preset move will only start the profile if the master counts are counting in the positive direction, a continuous move will begin the ramp to its new ratio following the master in either direction. As long as the master is counting in the positive direction, the direction towards which the slave starts in a continuous move is determined by the argument (sign) of the D command. The slave direction is positive for D+ and negative for D–.

If the master is counting in the negative direction when slave begins a continuous move, the direction towards which the slave moves is opposite to that commanded with the D command. The slave direction is positive for D– and negative for D+.

If the master changes direction during a continuous slave move, the slave will also reverse direction. As with standard continuous moves, the GO1 will continue until terminated by S, K, end-of-travel limits, stall condition, or command to go to zero velocity. As with preset moves, the sign on the FOLMAS command determines the direction of master counting with respect to the direction of actual counting on the master input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Master and Slave Distance Calculations

The formulas below show the relationship between master move distances and the corresponding slave move distances. These relationships may be used to assist in the design of Following mode moves in which both the position and duration of constant ratio are important.

In such calculations, it is helpful to use SCLMAS and SCLD values which allow the master and slave distances to be expressed in the same units (e.g., inches or millimeters). In this case, many applications will be designed to reach a final ratio of 1:1, and the distances in these figures can be easily calculated.

☞ **HINT**: For a trapezoidal preset slave move with a maximum ratio of 1:1, the master and slave distances during the constant ratio portion will be the same. The slave travel during acceleration will be exactly half of the corresponding master travel, and it will also occur during deceleration.

Master Distance (FOLMD)

If the slave is in continuous positioning mode (MC1), FOLMD is the master distance over which the slave is to accel or decel from the current ratio to the new ratio. If the slave is in preset positioning mode (MCØ), FOLMD is the master distance over which the slave's entire move will take place.

When the Slave is in Continuous Positioning Mode (MC1)

$$D = \frac{FOLMD * (R_2 + R_1)}{2}$$

where:

- FOLMD = Master distance
- R_2 = New ratio (FOLRN ÷ FOLRD)
- R_1 = Current ratio (FOLRN ÷ FOLRD)
- D = Slave distance traveled during ramp

When the Slave is in Preset Positioning Mode (MCØ)

Trapezoidal Slave Moves

$$D_{max} = (FOLMD * R_{max})$$

$$D_1 = \frac{(D_{max} - D)}{2}$$

$$MD_1 = \frac{2 * D_1}{R_{max}}$$

$$D_2 = D - (2 * D_1)$$

$$MD_2 = \frac{D_2}{R_{max}}$$

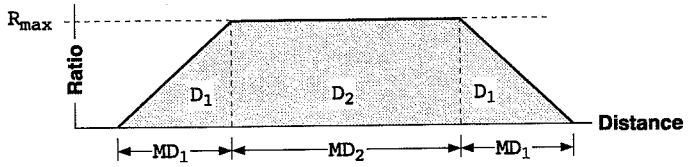
$$FOLMD = (2 * MD_1) + MD_2$$

where:

- FOLMD = Master distance
- MD_1 = Master distance during accel & decel ramps
- MD_2 = Master distance during constant ratio
- D = Total slave preset distance commanded
- D_1 = Slave travel during accel and decel ramps
- D_2 = Slave travel during constant ratio
- D_{max} = Maximum slave distance possible (assuming no accel or decel)
- R_{max} = Maximum ratio = FOLRN ÷ FOLRD

Trapezoidal profile if:

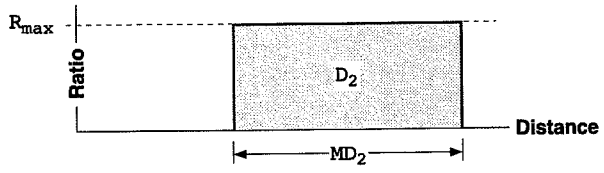
$D = (D_2 + 2 * D_1) < D_{max}$



NOTE ☞
Profiles depict ratio vs. distance.

Rectangular profile if:

D_2 = D
D_1 = zero
MD_2 = FOLMD
MD_1 = zero



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Triangular Slave Moves

$$D = \frac{R_{peak} * FOLMD}{2}$$

where:

FOLMD = Master distance

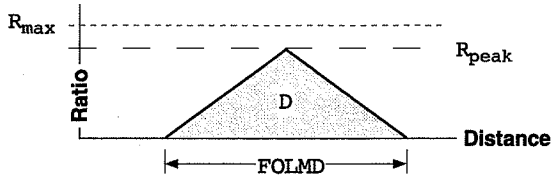
R_{peak} = Peak ratio reached during move

D = Total slave distance

NOTE ☞ *Profile depicts ratio vs. distance.*

Triangular profile if:

$D < 1/2\ D_{MAX}$



Distance Calculation Example

In the example below, the desired travel during constant ratio is already contained in numeric variable #1 (VAR1), and may have been read from thumbwheels or a DATA command. The corresponding slave move distance (D) and FOLMD are calculated as shown:

Command	Description
> VAR2=2	Desired slave travel during accel and decel combined
> VAR3=2 * VAR2	Required master travel during these ramps
> VAR4=VAR1 + VAR2	Move distance is constant ratio portion plus ramps
> VAR5=VAR1 + VAR3	Master travel for entire slave move
> FOLMD(VAR5)	Establish calculated master travel
> D(VAR4)	Establish calculated slave travel
> GO1	Make desired slave move

Similar calculations may be done for a series of continuous move ramps to ratios, separated by GOWHEN for master cycle positions. These ramps may be repeated in a loop to create a continuous cyclical slave profile.

Beware of Roundoff Error (Scaling only)

Some potential for roundoff error exists if the scaling of a move distance or master distance by SCLD and SCLMAS does not result in an integer number of steps. Some additional care must be taken in the segment by segment construction of profiles using ramps to continuous ratio. The 6000 controller maintains a slave position command which is calculated from the commanded constant ratios, the ramps to the new ratios, and the master travel over which these take place. At the end of each ramp or constant ratio portion, this commanded position is calculated to the nearest integer slave step. If the ratios and master travel result in a non-integer slave travel for a segment, the fractional part of that segment's calculated travel will be lost. In a cyclical application, repeated truncations could build up to a significant error. This may be avoided through careful attention to design of the profile.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Other Features with Following

The 6000 controller has many features that may be used in the same application as its Following features. In some cases, having configured an axis as a slave with the FOLMAS command will affect, or be affected by, the operation of other features, as described below.

Setups used by FOLMAS (Steppers Only)

The FOLMAS command uses the drive resolution (DRES), the velocity range (PULSE), and the choice of motor step positioning or encoder step positioning (ENC) data to configure an axis as a slave. If encoder step positioning is chosen (ENC1) , the encoder resolution (ERES) data is also used. **In order for this data to be used correctly, these commands must be given before FOLMAS. These commands are not allowed after FOLMAS is given.** After FOLMAS is executed, the MC command may be used to change from incremental to absolute positioning and back, but a change to encoder or motor step positioning (ENC) is not allowed.

S (Stop) & K (Kill) Commands

Stop (S) and Kill (K) commands cause the slave to do a non-Following decel, even if the slave is in Following mode. A stop or a kill, buffered or immediate, will clear a pending GOWHEN condition (clears AS bit #26).

GO Command

If a slave axis is in Following mode (FOLEN1), moves will ramp to a ratio (set with FOLRN and FOLRD). If it is not in Following mode, moves will ramp to a velocity (V). Switching in and out of Following mode does not change the value for final ratio or final velocity goals, but simply changes which parameter is used as the goal.

Registration Moves (Steppers Only)

Registration inputs may be enabled with the INFNCi-H command while an axis is a slave, and registration moves may interrupt either a Following mode move or a time-based move. The registration move itself, however, is always a time based move, and implements the registration velocity with respect to a stationary reference. **Any trigger input may be used as a registration input or for any Following feature which uses triggers, even at the same time.**

Enter/Exit Following Mode While Moving

The FOLENØ command may be executed while a slave is moving at constant ratio. In this case, the current velocity becomes the constant velocity, and the slave may accelerate or decelerate to other velocities. The FOLEN1 command may not be executed while the slave is moving in the non-Following mode. Attempting to do so will result in the error response "MOTION IN PROGRESS".

Pause and Continue

A program may be paused and continued, even if one or more axes is configured as a slave. Those axes do not lose track of the master input, even though motion is stopped. As usual, if a program finishes normally, or if COMEXS is set to zero (COMEXSØ), the program may not be continued. If a program is resumed, partially completed Following moves will be completed; however, the remainder of the move is completed over the entire original master distance (FOLMD value).

TVEL and TVELA Commands

The TVEL and VEL commands have always reported an unsigned value (i.e. magnitude), consistent with the fact that the V command is always positive, even if the move direction is negative. When Following is enabled (FOLEN1), TVEL and VEL report the net magnitude of commanded velocity due to following and/or shifting. For example, if the slave is following in the positive direction at 1 rps, TVEL reports 1.000. If a shift in the negative direction of 1.5 rps is then commanded, TVEL will report 0.5 – still positive, even though the net direction is negative.

By contrast, TVELA and VELA have always been signed. In the above example, TVELA will report -0.5.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TAS and AS Axis Status Bits

Axis Status (TAS and AS) bits 1-4 represent the motion status of an axis. Bits 1 and 2 represent the moving and direction status of the net motion of an axis, including the combination of following and shifting.

Bits 3 and 4 represent acceleration and velocity, respectively. With Following disabled (FOLENØ), the accel and velocity can only be due to a commanded time-based profile. With Following enabled (FOLEN1), the net commanded velocity and accel could be due to following the master and/or a simultaneous shift (time-based profile). Bits 3 and 4 only reflect the acceleration and velocity status of the time-based profile, not the combined command.

For example, if the slave is following in the positive direction at 1 rps, TAS reports 1ØØØ. If a continuous shift (FSHFC2) of 0.8 rps in the negative direction is then commanded, TAS will report 1Ø1Ø, while the shaft is decelerating to 0.2 rps. When the continuous shift velocity is reached, TAS will report 1ØØ1.

Changing Feedback Sources (Servos only)

Changing feedback sources (SFB) may result in a slave following its own feedback. For this reason, changing feedback sources is not allowed while a master is specified (FOLMAS).

Following and Other Motion

Following motion is initiated with the GO command, just like normal motion. Other motion, which does not depend on the motion of an external master, may not be used while a slave is in Following mode (FOLEN1). These motion types include jog mode, joystick mode, contouring, homing, streaming mode, and linear interpolation (GOL). To use these motion types, Following must be disabled (FOLENØ) — see *Enter/Exit Following Mode While Moving* above for precautions.

Conditional Statements Using PMAS

The master cycle position (PMAS) value may be used in the comparison argument of these commands:

- WAIT & GOWHEN: If it is desired to WAIT or GOWHEN on a master cycle position of the next master cycle, one master cycle length (value of FMCLEN) should be added to the master cycle position specified in the argument. This allows commands that sequence slave events through a master cycle to be placed in a loop. The WAIT or GOWHEN command at the top of the loop could execute, even though the actual master travel had not finished the previous cycle. This is done to allow a PMAS value which is equal to the master cycle length to be specified and reliably detected.
- IF, UNTIL, & WHILE: These arguments use the instantaneous PMAS value. Be careful to avoid specifying PMAS values that are nearly equal to the master cycle length (FMCLEN), because rollover may occur before a PMAS sample is read.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C H A P T E R ⑤

Troubleshooting

Symptoms and Causes

The table below offers some possible reasons for troubles which may be encountered in achieving the desired slave motion.

Symptom	Possible Causes
Slaves do not follow master	• Improper FOLMAS • Poor connection if master is encoder • Master running backward • No encoder power (when the encoder is selected as the master)
Slave motion is rough	• FFILT command value too low • Unnecessary FPPEN amplifies master roughness
Ratio seems wrong	• FOLRN and FOLRD slave-to-master ratio values are inaccurate, possibly reversed • SCLD or SCLMAS wrong • DRES or ERES wrong for encoder step slaves (steppers only) • Following limited by FMAXV or FMAXA (steppers only)
Slave profile wrong, or un-repeatable	• WAIT used where GOWHEN should be • Too little master travel between GOWHEN and GO1, desired PMAS is missed
Master/slave alignment drifts over many cycles	• Roundoff error due to fractional steps resulting from SCLD or SCLMAS and user's parameters • Ratios and master distances specified result in fractional slave steps covered during ramps, constant ratio
Slave lags Following position **(steppers only)**	• Inhibited by FMAXV • FMAXA *clips* acceleration peaks resulting from attempt to follow rough master
Slave dithers, or oscillates about desired position **(steppers only)**	• EPMG too high for encoder step slave • FMAXA too low for EPMV

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Error Messages

If an illegal programming condition is discovered while programming or executing programs, the 6000 controller responds with an error message. If a program execution error is detected, the program is aborted.

The table below lists all the error messages that relate to Following, and indicates the command and cause that may generate them. These error messages are displayed only if the error level is set to level 4 with the ERRLVL4 command (this is the default setting).

Error Message	Cause
FOLMAS NOT SPECIFIED	No FOLMAS for the axis is currently specified. It will occur if FMCNEW, FSHFC, or FSHFD commands are executed and no FOLMAS command was executed, or FOLMASØ was executed.
INCORRECT DATA	Velocity (V), acceleration (A), or deceleration (AD) command is zero. (used by FSHFC & FSHFD)
INVALID CONDITIONS FOR COMMAND	The FOLMD command value is too small to achieve the preset distance and still remain within the FOLRN/FOLRD ratio. A command phase shift cannot be performed: FSHFD .....Error if already shifting or performing other time based move. FSHFC .....Error if currently executing a FSHFD move, or if currently executing another FSHFC move in the opposite direction. The FOLEN1 command was given while a profile was suspended by a GOWHEN.
INVALID DATA	The parameter supplied with the command is invalid. FFILT .....Error if: smooth number is not 0-4 FMCLEN.....Error if: master steps > 999999999 or negative FMCP.....Error if: master steps > 999999999 or <-999999999 FOLMD .....Error if: master steps > 999999999 or negative FOLRD .....Error if: master steps > 999999999 or negative FOLRN .....Error if: slave steps>999999999 or negative FSHFC .....Error if: number is not 0-3 FSHFD .....Error if: slave steps>999999999 or <-999999999 GOWHEN.....Error if: position > 999999999 or <-999999999 WAIT.....Error if: position > 999999999 or <-999999999 Error if a GO command is given in the preset positioning mode (MCØ) and: FOLRN = zero FOLMD = zero, or too small (see *Preset Moves* section in Chapter 2)
INVALID FOLMAS SPECIFIED	An illegal master was specified in FOLMAS. A slave may never use its own commanded position or feedback source as its master.
INVALID RATIO	Error if the FOLRN:FOLRD ratio after scaling is > 127 when a GO is executed.
MOTION IN PROGRESS	The FOLEN1 command was given while that slave was moving in a non-Following mode.
NOT VALID DURING FOLLOWING MOTION	A GO command was given while moving in the Following mode (FOLEN1) and while in the preset positioning mode (MCØ).
NOT VALID DURING RAMP	A GO command was given while moving in a Following ramp and while in the continuous positioning mode (MC1). Following status (FS) bit #3 will be set to 1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A — Following Commands

Detailed information about these commands is provided in the *6000 Series Software Reference Guide.*

ERROR .....Enable bit #14 to check for when a GOWHEN condition is already true when a subsequent GO, GOL, FSHFC, or FSHFD command is given.

ERRORP.....If ERROR bit #14 is enabled, a GOTO branch to the error program occurs if a GOWHEN condition is already true when a subsequent GO, GOL, FSHFC, or FSHFD command is given.

FFILT .....Sets the bandwidth for master position filtering.

FMAXA .....**STEPPERS ONLY:** Sets the max. acceleration a slave may use while Following.

FMAXV .....**STEPPERS ONLY:** Sets the max. velocity at which a slave may travel.

FMCLEN.....Defines the length of the master cycle.

FMCNEW.....Restarts new master cycle counting immediately. To make this a trigger-based operation instead of issuing the FMCNEW command, use the TRGFNcx1 command.

FMCP .....Defines the initial position of a new master cycle.

FOLEN .....Enables or disables Following mode.

FOLMAS.....Defines masters for slave axes.

FOLMD .....Defines the master distance over which ratio changes or moves are to take place.

FOLRD .....Establishes the **denominator only** for the max. slave-to-master ratio for a preset move or the final ratio for a continuous move (use in combination with the FOLRN command).

FOLRN .....Establishes the **numerator only** for the max. slave-to-master ratio for a preset move or the final ratio for a continuous move (use in combination with the FOLRD command).

FPPEN .....Allows master position prediction to be enabled or disabled.

FSHFC .....Allows continuous advance or retard (shift) of slave position during continuous Following moves.

FSHFD .....Allows preset advance or retard (shift) of slave position during continuous Following moves.

GOWHEN.....A GOWHEN command suspends execution of the next slave move until the specified conditional statement (based on PC, PE, PM, PMAS, PSLV, or PSHF) is true. To make this a trigger-based operation instead of issuing the GOWHEN command, use the TRGFNc1x command.

SCLD.....Sets the slave distance scale factor.

SCLMAS.....Sets the master scale factor.

TRGFN .....TRGFNc1x initiates a GOWHEN. Suspends execution of the next slave move until the specified trigger input (c) goes active.
TRGFNcx1 allows master cycle counting to be restarted when the specified trigger input (c) goes active.

Status and Assignment Commands:

TAS and [AS] .....Bit 26 of each axis status is set (1) if a motion profile suspended by a GOWHEN (including TRGFNc1x) is pending on that axis. The bit is cleared when the GOWHEN is true, or when a stop (S) or kill (K) command is executed.

TER and [ER] .....Bit 14 is set if the GOWHEN condition is already true when a subsequent GO, GOL, FSHFC, or FSHFD command is given. (The corresponding error-checking bits must be enabled with the ERROR command before the error will be detectable.)

TFS and [FS] .....Transfers or assigns/compares the Following status or each axis.

TNMCY and [NMCY]...Transfers or assigns the current master cycle number.

TPMAS and [PMAS]...Transfers or assigns the current master cycle position.

TPSHF and [PSHF]...Transfers or assigns the net position shift since constant ratio.

TPSLV and [PSLV]...Transfers or assigns the slave's current commanded position.

TVMAS and [VMAS]...Transfers or assigns the current velocity of the master axis.

WAIT.....A WAIT command suspends program execution until the specified conditional statement (based on PMAS, FS, NMCY, PSFER, PSHF, PSLV, or VMAS) is true;
WAIT(SS.i=b1) suspends program execution until a trigger input is activated ("i" is the input bit number corresponding to the trigger input—refer to the *6000 Series Software Reference Guide*).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

I N D E X

A-B

acceleration
  maximum (steppers) 28
  scaling 5
accuracy, factors affecting 29
application examples
  continuous cut-to-length 22
  continuous phase shift 12
  electronic gearbox 14
  preset phase shift 13
  trackball 14
assignment of master and slave 4
axis status, relative to Following 36

C

change summary i
commanded position, slave
  tracking error 27
commands, Following (list of) 39
conditional statements using PMAS 20
continue, effect on Following motion 35
continuous cut-to-length application 22
continuous positioning mode moves 9, 32
  distance calculations 33
contouring 36
correction, velocity 28
counting, master cycle, *see master, master cycle*

D

deceleration, scaling 5
distance
  calculations 33
  fractional step truncation 6
  scaling, *see scaling*
drive resolution (DRES) 35
dynamic position maintenance 28
  effect on accuracy 31

E

electronic gearbox application 14
enable or disable Following 8
  while moving 35
error
  error messages 38
  Following (PER) 27
  position (steppers) 28

F

feedback source, changing (servos) 36
filtering, master position, *see master, master position filtering*
Following
  commands, list of 39
  enable or disable 8
    while moving 35
  error 27
  performance considerations 2
  prerequisites to Following motion 3
  Ratio Following introduction 1
  set-up parameters 3
  status 2
  technical considerations 25
fractional step truncation 6

G-J

GOWHEN
  cleared by stop or kill 35
  using PMAS 20, 36
homing 36
IF, using PMAS 20, 36
initial master cycle position 19
jog mode 36
joystick mode 36

K-L

kill (K) command, effect on Following motion 35
length, master cycle 18
linear interpolation 36

M-O

master
  definition of 4
  distance
    move calculations 33
    programming (FOLMD) 8
  master cycle
    concept 17
    counting 17
      restart 18
      status 18
    length 17, 18
    number 19
    position 17, 18
      assignment/comparison (PMAS) 19
      initial 19
      rollover 17, 36
      synchronizing 21
      transfer (TPMAS) 19
  master input (A.K.A.) 1
  master position filtering 2, 26
    effect on accuracy 30
    effect on phase tracking 31
    effect on position accuracy 30
    status 25
  master position prediction 2, 25
    effect on accuracy 30
    status 25
  move profiles 8
  ratio to slave 7
  resolution 29
  scaling 5
  velocity 2, 26
maximum acceleration (steppers) 28
maximum velocity (steppers) 28
motion
  Following motion status 3
  restrictions while in Following 36
  rough 37
  synchronized with PMAS 21

P-Q

pause, effect on Following motion 35
performance considerations 2
phase, shift 11
  status 11
phase, tracking 31
position
  accuracy 30
  commanded position calculation 27
  dynamic position maintenance 28
    effect on accuracy 31
  error 28
  master, *see master*
  sampling period 2
    effect on accuracy 30
  slave 11
power-up program (STARTP) 5
prediction of master position, *see master, master position prediction*
preset positioning mode moves 10, 32
  distance calculations 33

R

ratio of slave to master 7
registration 35
repeatability
  position sampling rate 30
  sensors 31
  trigger inputs 31
resolution
  drive 35
  master 29
  slave 29
restart master cycle counting 18
rollover of master cycle position 17, 19, 36
rough, motion 37

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

S

scaling
  master 5
  slave 5
shift
  continuous 11
  preset 11
  status 11
slave
  commanded position 11
    Following error 27
  definition of 4
  distance
    move calculations 33
    scaling 5
  motor/drive accuracy 30
  move profiles 8
  ratio to master 7
  resolution 29
  scaling 5
  shift, *see shift*
status
  axis, relative to Following 36
  Following 2
stop (S) command, effect on
  Following motion 35
streaming mode 36
synchronizing Following moves with
  master positions 21
system update period (servos) 2

T

technical considerations for
  Following 25
trackball application 14
transfer
  master cycle number (TNMCY) 19
  master cycle position (TPMAS) 19
trigger input functions
  repeatability 31
  restart master cycle counting 18
troubleshooting 37
truncation, distance 6
tuning, effect on accuracy (servos)
  31

U-Z

UNTIL, using PMAS 20, 36
velocity
  correction 28
  maximum (steppers) 28
  range due to PULSE (steppers) 35
  scaling 5
  TVEL & TVELA responses relative
    to Following 35
WAIT
  using PMAS 20
WAIT, using PMAS 36
WHILE, using PMAS 20, 36

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Parker Hannifin Corporation
Hydraulic Valve Division
520 Ternes Avenue
Elyria, Ohio 44035
Tel: (216) 366-5200
Fax: (216) 366-5253

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP