CHAPTER ⑤

SXF Follower

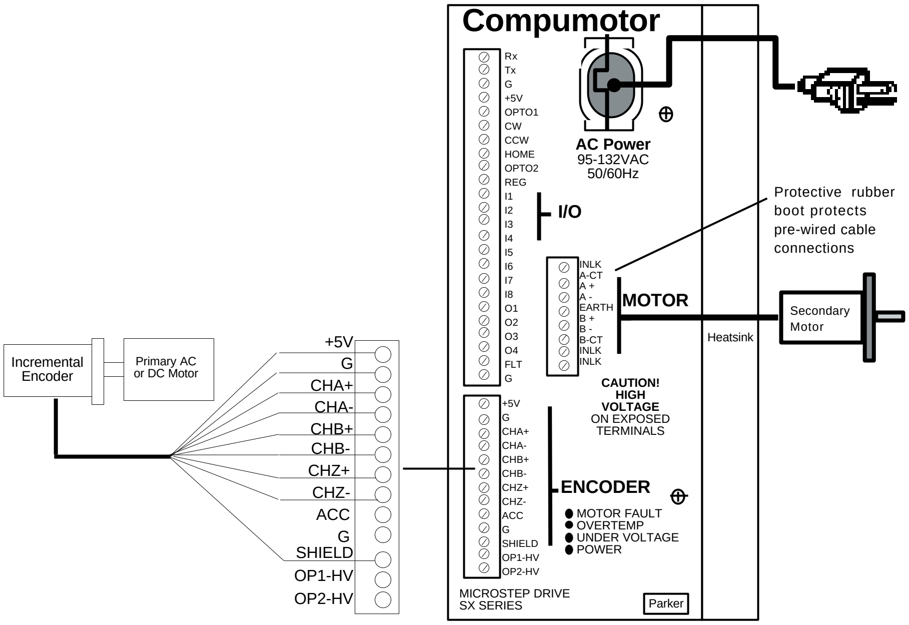
Chapter Objectives

The information in this chapter will enable you to:

- ❏ Understand basic following concepts
- ❏ Understand the basic types of following and their common applications
- ❏ Become familiar with the SXF commands associated with following

What is Following?

The SXF can perform velocity following and distance following moves. The SXF can follow from an incremental or absolute encoder input. *Unless otherwise noted,* ***all the features that will be presented or have been discussed are also valid in Following mode****.* ***The only difference is that you replace the velocity command with a speed ratio and the acceleration with a following acceleration for distance following.*** *Instead of specifying the speed with the* **`V`** *command, you will specify the speed with respect to the primary using the* **`FOL`** *command*. The following figure shows the basic following system configuration.



Typical Following System Configuration

In the previous figure, the primary axis is an AC or DC motor. The encoder is mounted on the shaft of the primary motor. The encoder provides the SXF with data on the primary motor's position and velocity. The SXF uses the position and velocity data to move the secondary axis. Therefore, the secondary motor is *following* the motion of the primary motor. The SXF controls the motion of the secondary motor according to the motion of the primary. This concept of following can be used in different forms to satisfy many different applications.

Types of Following

There are four types of Following application motion.

❑ Velocity following
❑ Velocity and position following
❑ Recede and advance while following (includes electronic cam)
❑ Phase error correction (synchronization)

The order in which the categories are presented are roughly in order of increasing complexity. By identifying how the secondary axis motion relates to the primary axis, you will be able to determine your application's type and the applicable programming commands. The SXF also provides several addition features that have been added to address more specific attributes of the various applications. The SXF has several additional features that enhance the functionality for an application.

❑ Registration while following
❑ Jogging in the following mode
❑ Following a pulse train and a direction
❑ Following encoder pulses and the encoder direction
❑ Entering following ratio via thumbwheels
❑ Capability of loading the primary encoder position into a variable

The SXF can perform all of the SX's programming functions. The only difference is that the motion profiles are now following moves and motion is based on a primary axis. The SXF can also be used in its Indexer mode. You can easily switch back and forth from follower and Indexer functionality and create motion programs in which the SXF serves as both a follower and an Indexer. The SXF has the following programming capabilities:

❑ Variables, general-purpose & read-only (position, primary encoder position, etc.)
❑ Math
❑ Complex branching—**IF ELSE**, **REPEAT UNTIL**, **WHILE**
❑ On-the-fly changes—**MPP** mode
❑ Flexible I/O

For an explanation of the features that are common to both the SX and the SXF, refer to *Chapter 4, Application Design*. You should understand how to set the SXF for following. You can use one command (**FSI**) to enter and exit the Indexer and Following modes.

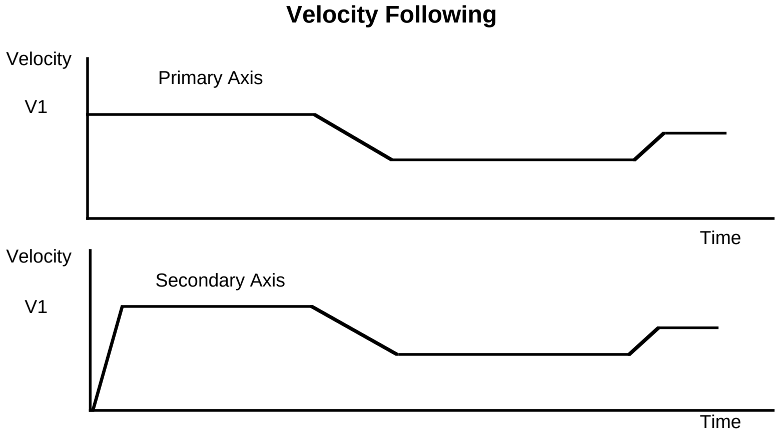
Command	Description
FSI1	Enters the following mode
FSIØ	Exits the Following mode(Indexer mode)

The rest of this chapter explains the four following types. Additional features of SXF following will also be covered.

Velocity Following

In ***velocity following***, the secondary axis is concerned only with the primary axis' speed. The relationship of the primary axis position with respect to the secondary axis is irrelevant. In this type of application, the secondary axis accelerates at a specified acceleration up to a ratio of the primary axis' speed. Preset or continuous moves are performed in the same manner as in Indexer mode. Exact distances on the secondary axis can be moved in Preset mode. However, instead of moving at a velocity specified by the **V** command, the secondary axis moves at a ratio of the primary axis' velocity specified by the **FOR** and **FOL** commands. The acceleration is independent of the primary axis encoder and is specified by **A** and **AD** (as with an Indexer).

The following figure shows velocity following. Once the secondary axis accelerates to the specified following ratio, it tracks the primary axis' speed and position at the specified ratio. During acceleration, the primary encoder speed and position are not followed. The acceleration ramp is independent of the primary axis. Once the secondary axis has accelerated to the specified following ratio, it will be following the primary axis (if the primary axis slows down, the secondary axis slows down).

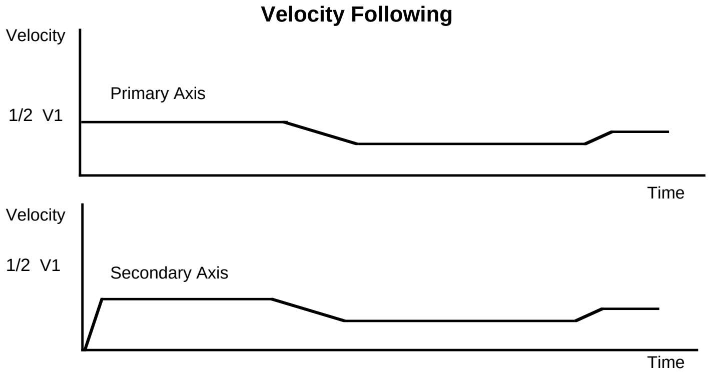


Velocity Following

Velocity following is used in dispensing and on-the-fly cutting applications.

In each of the following categories described, the secondary axis only moves in the direction specified by either the **D** command, the **V** command, or the **H** command. If the primary axis changes direction, the secondary axis will still move in the same direction. Only the number of pulses and the rate of the pulses determines the secondary axis' motion. With position and direction tracking enabled, the secondary axis will follow the primary axis' direction. ***To illustrate velocity following, detach the encoder from the primary axis and manually move the encoder***.

In the previous figure, if the primary velocity changed after the following ratio was achieved, position and velocity were tracked exactly. If the primary axis' velocity were half as fast as the example in next figure, a different phase or positional relationship would result during acceleration from rest. This is why it is velocity following. ***Again, once the secondary axis accelerates to the specified following percentage, it will follow velocity and position exactly***.



Primary Axis as Half Speed with Velocity Following

Setting Up Velocity Following

To perform velocity following, you must use two commands.

Command	Description
FOR	Relates the number of secondary motor steps per unit of travel to the corresponding primary encoder steps per unit of travel.
FOL	Relates the speed of the secondary axis to the speed of the primary axis as long as **FOR** is set correctly. The value is entered as a percentage of the primary axis speed.

The SXF uses the equation below to determine the number of motor steps.

$$\text{Motor Steps} = \mathbf{FOR} * \frac{\mathbf{FOL}}{100} * \text{Encoder Steps}$$

These two commands, (**FOR** and **FOL**) remove the complication of having a different resolution for measurement of distance on the primary and secondary axes. Once you relate the number of primary encoder pulses per unit of travel to the number of secondary motor pulses per that same unit of travel, you can relate their speeds. You can specify the secondary axis to move at 50% of the speed of the primary axis (1:2 ratio) or 100% of the speed (1:1 ratio). The encoder is usually mounted on the primary axis or through gearing and will measure a certain number of pulses per inch. The secondary axis would also have some certain number of pulses per inch. The ratio of the secondary axis steps to the primary axis' steps is entered in the **FOR** command. If the primary axis moves at 10 inches per second (ips) and the secondary axis is also set to move at 10 ips, a speed percentage of 100 is all that needs to be programmed (**FOL1ØØ**). ***The remaining commands are standard SX commands.*** Use the following commands to implement the velocity following feature:

Step ① Enter the number of secondary motor steps per unit of travel per number of primary encoder step per the same unit of travel. In this example, the motor step resolution is 25,000 steps/rev and the encoder resolution is 4,000 steps/rev. The unit of travel is 1 revolution—the ratio is 25,000/4,000 or 6.25.

Command	Description
> FOR6.25	Sets the motor step to encoder step ratio of 6.25
> OFF	Turns the SXF off
> CMR25ØØØ	Sets motor resolution to 25000 steps/rev
> ON	Turns the SXF on

Step ② Enter the following mode with the FSI command.

Command	Description
> FSI1	Enters the Following mode

Step ③ Set the speed ratio. If the secondary axis is to move at the same speed as the primary axis (100% of the primary axis' speed), enter **FOL1ØØ**.

Command	Description
> FOL1ØØ	The secondary axis will move at 100% of the primary axis' speed

Step ④ Set the SXF to .i. continuous mode and begin motion with the G command.

Command	Description
> MC	Enters Continuous mode
> A5ØØ	Sets acceleration
> AD5ØØ	Sets deceleration
> G	Initiates motion

Turn the encoder. The SXF should begin moving at the same speed that you are turning the encoder. Change the direction that you turn the encoder and note the motion of the secondary axis. Follow the steps below.

Step ① Execute the following commands.

Command	Description
> S	Stops the continuous motion command issued above
> 1FSP1	Enables the Position and Direction Tracking mode
> G	Initiates motion

Step ② Now turn the encoder or move the primary so that the encoder moves. The SXF will begin moving. Change the direction that the encoder is moving. The SXF will change direction. To change the relative direction between the SXF and the encoder, use the .i.change direction; command (H±). Now turn the encoder and note that the SXF's relative direction has changed. Tracking the direction as well as pulses can only be used in continuous mode. Preset moves can also be performed in velocity following.

Preset Moves In Velocity Following

A preset move is performed as in the standard Indexer mode. Issue the mode normal command (**MN**) and specify a distance in terms of secondary motor steps. In a preset move:

① The secondary axis accelerates (**A**) to the desired speed percentage.
② It decelerates at the rate set with the **AD** command.
③ It moves the specified distance (**D**) and stop.

If the primary axis' speed varies, the acceleration ramp will be the same, but the distance that the secondary axis travels to reach its following ratio will be different. The positional relationship or phase relationship is not maintained during acceleration. Once the secondary axis achieves the following percentage speed, it will track both velocity and position exactly.

Step ① Attach the encoder to the primary axis. Start the primary axis moving.

Step ② Enter the following set of commands.

Command	Description
> MN	Enters Normal mode
> FSPØ	Exits the direction Tracking mode
> A5ØØ	Sets acceleration
> AD5ØØ	Sets deceleration
> FOL1ØØ	Sets the following speed percentage to 100
> D125ØØØ	Sets the distance to 125000 steps
> G	Initiates motion

The secondary axis will move 125000 motor steps at the same speed as the primary axis. Repeat the example, but vary the speed of the primary axis. ***The secondary still moves 125000 steps, and its speed varies with the primary axis***.

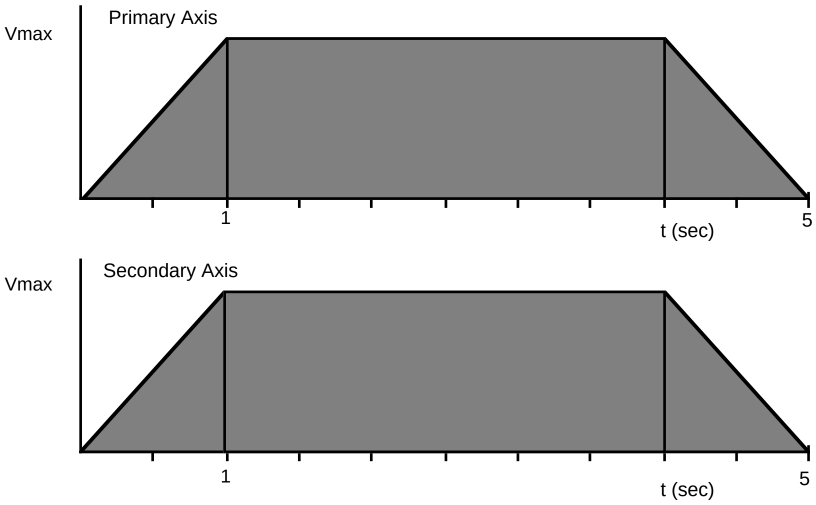
Velocity following is the simplest form of following. All motion in the SXF is programmed in the same manner as the SX. The only exception is that the velocity command is replaced by a following percentage.

Position and Velocity Following

Position and velocity following is the most common form of following. In this case, the secondary axis must maintain a specific positional relationship with the primary axis. The primary axis moves a specific number of primary encoder pulses while the secondary axis must move a specified number of secondary motor steps. The secondary axis moves at the same speed as the primary axis (a 1:1 speed ratio) in most of these applications. ***Coil winding applications, however, are an exception***. In coil winding applications, the relationship between the primary (usually the spindle axis) and secondary axes (usually the traverse axis) is based on a desired pitch. This pitch defines a positional and velocity relationship that will have an arbitrary speed relationship based on the particular application. The positional relationship is usually defined by a single traverse corresponding to a specific number of spindle axis turns, thus defining a specific number of primary encoder pulses while a specific number of secondary pulses must be traveled.

Primary Axis at Rest

There are two variations of ***position and velocity following***. The difference is in the motion of the primary axis. The less common form starts the primary axis motion from rest. The secondary axis must follow the primary axis pulse for pulse. In this type of application, the secondary axis will be in a continuous move and will track the motion of the primary axis exactly. A *web positioning system* where two axes are guiding the web (one edge is the primary axis and the other edge is the secondary axis), is an example of such an application. The web can be positioned based on the motion relationship between the primary and secondary axes. The following figure shows the profiles of a secondary axis following a primary axis pulse for pulse.



Pulse-for-Pulse Following

If the primary axis starts from rest, the secondary axis must track it pulse for pulse. **FSA** enables this capability. If position tracking is disabled, the SXF will follow the pulse count only and will not change direction if the primary axis changes direction. This is acceptable in many applications. In some applications, however, the primary axis may overshoot when it comes to a rest and the encoder will change directions.

Since the SXF is following the primary axis' pulse count only, it will actually move the secondary axis too many pulses. By enabling Position Tracking (**FSP1**), you can you can track the primary encoder's direction and pulse count and therefore not accumulate excess pulses caused by overshoot.

Command	Description
FSA1	Enables Pulse Tracking mode where secondary instantaneously accelerates between commanded velocities
FSP1	Secondary tracks both the direction & pulse count from the primary encoder

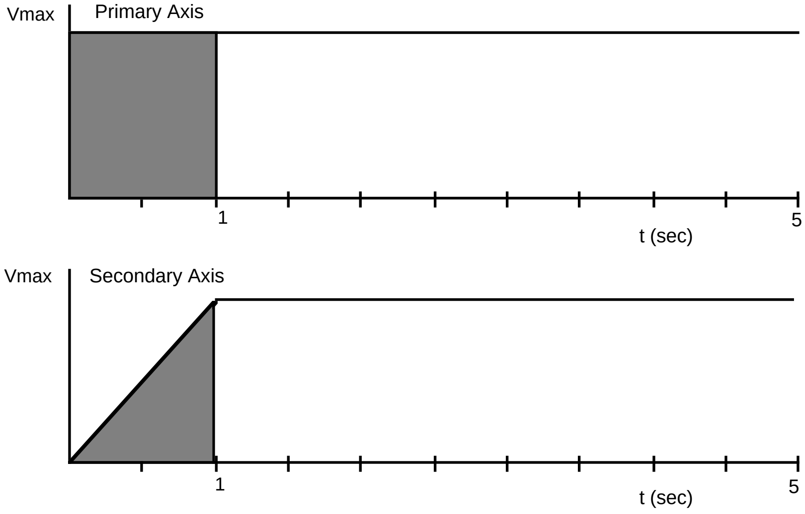
You must disable the Following Synchronized Acceleration mode (**FSFØ**) and activate Continuous Move mode to use Pulse Tracking mode (**FSA1**). **FSAØ** disables Pulse Tracking mode.

Primary Axis in Motion

The most common form of ***position and velocity following*** begins with the primary axis already in motion. The secondary axis must accelerate up to the specified speed ratio of the primary axis. In this type of application, the secondary axis must accelerate to a known positional or phase relationship with the primary axis. The primary axis is usually a conveyer or web and the secondary axis is performing an operation on the web or parts on the conveyer. The primary axis is always moving so the secondary axis must be moving at the same speed and with the correct orientation to perform some operation on the moving primary axis. To maintain positional and velocity relationships, the secondary axis must be able to accelerate over a known distance with respect to the primary axis (i.e., following acceleration is needed). The Set Following Synchronization Rate (**FAC**) and Set Following Synchronization Count (**FEN**) commands are used in conjunction with the velocity following commands.

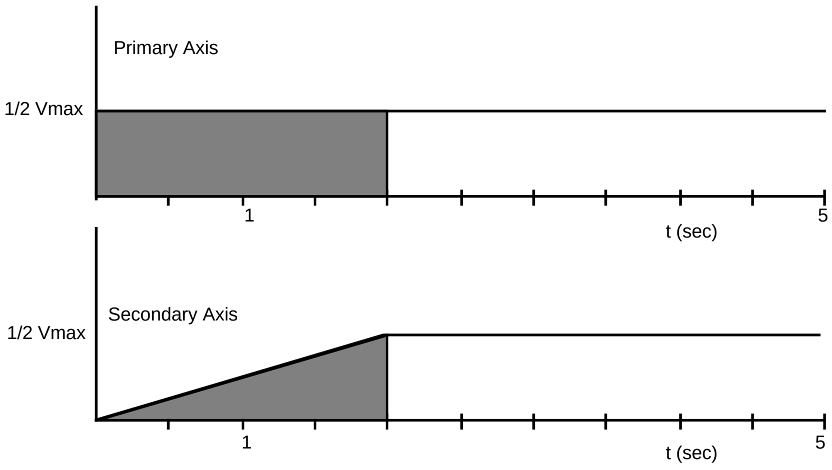
The **A** and **AD** commands used in velocity following are replaced by a following acceleration. A following acceleration is accomplished by stepping through subsequent ratios from the present following ratio to the final following ratio (**FOL**). The increment between ratios is set by the **FAC** command. Incrementing from one ratio to the next is based on the primary encoder changing by a set number of primary encoder pulses. This number of pulses is set by the **FEN** command. For example, if the value for **FAC** is 1, the value for **FEN** is 10, and the final following speed percentage is 100 (**FOL**), the secondary axis must accelerate from zero speed to 100% of the primary speed. From rest, every 10 primary encoder steps (**FEN**) the following percentage will change by an increment of 1% (**FAC**) until it is equal to the final following percentage of 100% (**FOL**). Therefore, the secondary axis will accelerate over 1000 primary encoder steps. It does not matter if the speed of the primary axis varies, the secondary axis' acceleration is based on primary encoder steps (not time).

The following figures illustrate position and velocity following. The shaded area indicates the distance moved by both the primary and secondary axes while the secondary axis is accelerating. ***The secondary axis moves 1/2 the distance that the primary axis moves***. This will always be the case when the secondary axis accelerates from rest to the same speed as the primary axis, when the primary axis is already moving.



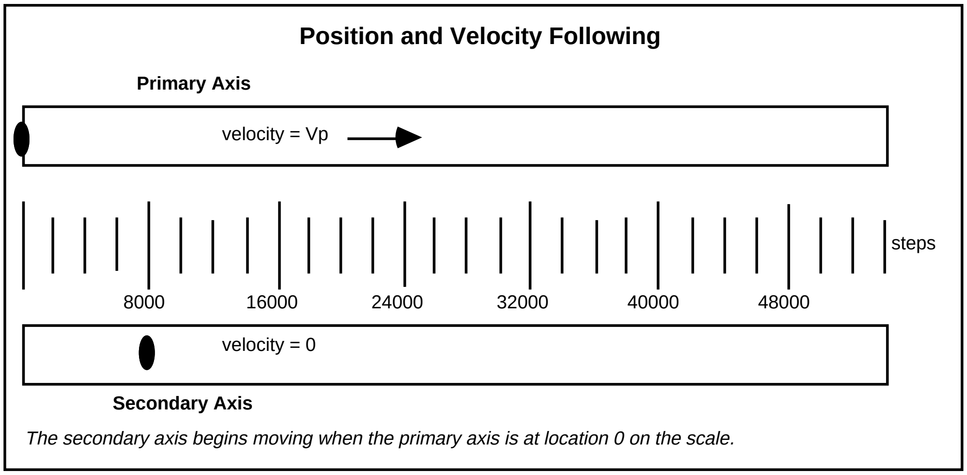
Following Acceleration

This figure shows how the acceleration ramp varies if the primary axis' speed varies. If the primary axis' speed is 1/2 the maximum velocity, the acceleration ramp will be twice as long. The shaded portions in the following figure are equivalent to the shaded portions in the previous figure. ***The primary axis' velocity can change at any time, even during the secondary axis' acceleration ramp, without changing the positional relationship***.



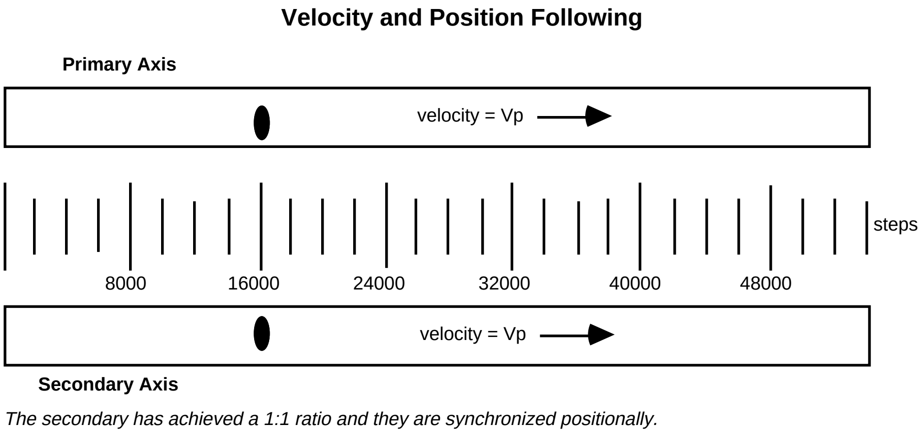
Following Acceleration With Primary Axis Velocity Change

The shaded area for the secondary axis is 1/2 of the shaded area under the primary axis curve. When the primary axis is moving and the secondary axis must start from rest and accelerate to the primary axis' velocity, the secondary axis will always move 1/2 the distance. Using following acceleration, the secondary axis adjusts its acceleration according to the primary axis' velocity so that it will always accelerate over the same distance while the primary axis moves a specified distance. To synchronize secondary and primary axes' positions, the secondary axis must start ahead of the primary axis (Refer to the figure titled *Starting a Velocity & Position Following Move—1:1 Ratio)* to compensate for the fact that the primary axis is already moving (velocity = **Vp**).


Position and Velocity Following
Primary Axis
velocity = Vp
steps
8000
16000
24000
32000
40000
48000
velocity = 0
Secondary Axis
The secondary axis begins moving when the primary axis is at location 0 on the scale.

Starting a Velocity & Position Following Move—1:1 Ratio

The key to position and velocity following is that the **Vp** value does not matter (previous figure). Assuming the application in the previous figure is programmed with following acceleration, accelerating over 8000 primary encoder steps, the spots will always be at a 1:1 speed ratio after the primary axis moves the 16000 steps (see the following figure).


Velocity and Position Following
Primary Axis
velocity = Vp
steps
8000
16000
24000
32000
40000
48000
velocity = Vp
Secondary Axis
The secondary has achieved a 1:1 ratio and they are synchronized positionally.

Velocity & Position Following After Acceleration Is Done

Following Acceleration

Accelerating over a known distance with respect to a known primary axis distance allows you to synchronize the exact phase relationship you want between the primary and secondary axes. Following acceleration enables this synchronization.

Command	Description
FAC	The increment of following ratio by which the following ratio changes during acceleration
FEN	The number of encoder pulses that cause the following ratio to increment to the next value
FSF1	Enables Following Acceleration mode

Determining FAC and FEN with Primary Axis Data

To determine what **FAC** *and* **FEN** *are, you must know*:

- □ The maximum velocity that the primary axis can travel (the velocity must be in units of primary encoder steps per second)
- □ The distance in primary encoder steps that the primary axis will move during which time the secondary axis will accelerate ***or*** the maximum acceleration that the secondary axis can accelerate

These parameters will be denoted as follows:

V_{Pmax} = Maximum primary velocity in encoder steps per second

D_{Pacc} = Distance primary axis travels while secondary axis accelerates in primary encoder steps

OR

As_{MAX} = Maximum acceleration of the secondary axis

Based on these equations, the values for **FAC** and **FEN** can be determined:

Equation 5-1. FEN

$$FEN = V_{PMAX} * \frac{TF}{1000} \qquad V_{Pmax} = \frac{\text{Primary Encoder Counts}}{\text{Second}}$$

TF = Primary Encoder Sample Period in ms

Equation 5-2. FAC

$$FAC = \frac{FOL * V_{Pmax}}{D_{Pacc}} * \frac{TF}{1000} \qquad V_{Pmax} = \frac{\text{Primary Encoder Counts}}{\text{Second}}$$

TF = Primary Encoder Sample Period in ms

FOL = Following percentage in units of percent

$\mathbf{Dp_{acc}}$ = Distance primary moves during secondary axis accel in units of primary axis enc counts

Typically, the application will have the secondary axis start from rest and accelerate up to an **FOL** value of 100 (1:1 ratio). However, the **FOL** value can be any value that is within the limits of the motor/drive system. ***D_{acc} is in units of primary encoder steps***. Determining the **FAC** and **FEN** values sets the number of primary axis encoder steps over which the secondary axis will accelerate (independent of the primary axis' speed). The secondary axis always travels the same number of motor steps during acceleration while the primary encoder moves Dp_{acc}.

The **TF** command allows you to set the sample period of the primary axis' encoder. It is programmable from 1 - 32 ms. ***The default is 4 ms***. **TF** simply scales the **FAC** and **FEN** values. In Equations 5.1 and 5.2, **TF** is used in units of ms—the constant of 1000 converts it to seconds so that the units cancel properly. Typically you will want **TF** to be as fast as your system will allow (1 ms). If the primary encoder is moving slowly, you may need to increase the sample rate to more than 1 ms because the actual encoder count does not change by much in a sample period and thus you have coarser resolution on the changes in encoder counts. For example, if your sample period is 1 ms and the maximum speed is 1 rps, the encoder count only changes by 4 counts each sample period. If there is a slight variation in speed and you read a change of 3 counts, there is a 25% variation. This may cause choppier secondary axis motion. Changing the encoder sample period can have a smoothing effect.

Determining FAC and FEN with Secondary Axis Data

Instead of knowing the distance that you want the secondary axis to accelerate over, you may know the maximum acceleration that your secondary axis can accelerate at. You can determine Dp_{acc} using As_{max} and the following speed percentage that you are accelerating to. Use Equation 5-3 to determine Dp_{acc} from As_{max}.

Equation 5-3.

Dp_{acc}

$$DP_{acc} = \frac{VP_{max}2}{2AS_{max}} * FOR * \frac{FOL}{100} = \frac{VP2_{MAX}}{2AS_{MAX}} * FOR * \frac{FOL}{100}$$

The acceleration is in units of secondary motor steps/sec^2. The maximum velocity of the primary axis is in primary encoder steps/sec.

Using this value for Dp_{acc}, you can use Equations 5-1 and 5-2 to determine **`FAC`** and **`FEN`**. Remember to enable the Following Synchronized Acceleration mode (**`FSF1`**) to enable following acceleration.

How Following Acceleration Works

The concept of accelerating the secondary axis over a known distance with respect to a known primary axis distance (independent of the primary axis' speed) is developed on the analogies drawn between *following* and *time-based motion*. In ***time-based motion***, the velocity describes the rate of change of the position with respect to a change in time. In ***following-based motion***, the secondary axis moves at a ratio of the primary axis' velocity. This following ratio is of the same units as the velocity it is following, but is simply scaled. Therefore, the following ratio is analogous to a velocity. In the same manner, an acceleration in the time domain is defined as the rate of velocity change. The analogy in following would be to have a following acceleration that is a rate of change of the following ratio. ***Time-based motion is based on sampled time whereas following is based on the sampled primary axis encoder pulses (for digital systems)***. Examine the following example.

- ❏ Primary axis encoder → 4000 counts/revolution
- ❏ Secondary axis → 4000 steps/revolution.
- ❏ Primary axis speed = 1 rps
- ❏ Secondary axis following speed percentage = 100%
- ❏ Distance that secondary axis accelerates = 2000 primary axis encoder steps
- ❏ Primary axis encoder sample period = 1 ms

Therefore, the secondary axis must now accelerate over 2000 primary encoder steps to a following percentage of 100% or a speed of 1 rps. Specifying the number of primary encoder pulses and the final speed that the secondary axis must attain after acceleration defines the ***acceleration ramp***. If the secondary axis' acceleration is based on time, you can calculate an acceleration ramp that would accelerate the secondary axis in the desired fashion.

$$\text{Acceleration time} = \frac{2000 \text{ steps}}{4000 \frac{\text{steps}}{\text{sec}}} = 0.5 \text{ seconds}$$

$$\text{Change in velocity} = 4000 \text{ steps/sec} — 0 \text{ steps/sec} = 4000 \text{ steps/sec}$$

$$\text{Acceleration} = \frac{4000 \frac{\text{steps}}{\text{sec}}}{0.5 \text{ sec}} = 8000 \frac{\text{steps}}{\text{sec}^2} = \frac{8 \frac{\text{steps}}{\text{sec}}}{\text{ms}}$$

If you changed the velocity by 8 steps/sec every sample period (**`TF`** - 1 ms), you would achieve the desired acceleration ramp. The problem is that application is dependent on time. If the primary axis' speed changes, the secondary axis would no longer be accelerating over 2000 primary encoder steps. Therefore, the application requires a following acceleration that is based on encoder pulses rather than time.

$$\text{Change in Following Percentage} = 100\% - 0\% = 100\%$$

$$\text{Following Acceleration time} = \frac{2000 \text{ steps}}{4000 \frac{\text{steps}}{\text{sec}}} = 500 \text{ ms} = 500 \text{ sample periods}$$

$$\text{Following Acceleration} = \frac{100}{500} = 0.2\ (\texttt{FAC})$$

The application is still time dependent. To remove the time dependency and make the acceleration dependent on the encoder pulses ***replace the time sample period by an encoder period***. In the above example, following percentage was based on changing the following percentage by 0.2 every sample period. It will take 500 sample periods to achieve a 100% following percentage. At 4000 steps/sec, the primary encoder is changing at a rate of 4 steps per ms or 4 steps/sample period.

1 sample period = 4 steps (encoder period—**FEN**)

Instead of changing the following percentage every sample period, change it every time the encoder count changes by 4 steps. If the primary encoder is moving at the maximum velocity, the acceleration ramp will be equal to the maximum acceleration provided. However, if the primary encoder velocity is less than the maximum velocity, the acceleration will also be reduced. The distance that the secondary axis accelerates and travels (with respect to the primary axis' moves during this acceleration) will remain unchanged. If the primary axis exceeds the maximum velocity, the acceleration ramp would also increase and exceed the maximum acceleration, which may cause a motor stall.

In summary, following acceleration uses the analogy that a change in velocity per change in time (normal acceleration) is the same concept as a following acceleration being a change in following percentage per change in encoder steps. In the SXF, the following percentage changes by the **FAC** value for each change in encoder steps of **FEN** steps. Several different combinations of **FAC** and **FEN** can achieve the same acceleration ramp. However, only unique **FAC** and **FEN** values will satisfy a specific maximum velocity and maximum acceleration. **FAC** and **FEN** can easily be determined with the equations and examples discussed earlier.

Decelerating

The SXF decelerates to zero speed using the **AD** value. Although you may expect that this will diminish the positional relationship, this is usually not a concern at the endpoint in a profile. The following move is usually started by a trigger, which indicates that the primary axis is at a particular location. The move could also be started based on the primary axis' encoder position. The move is typically of a preset distance. ***The important point is that the secondary axis is at a known position with respect to the primary axis when the profile begins***. This positional relationship is maintained during acceleration. After the secondary axis moves the appropriate distance, it will normally return the same distance it just traveled at a high speed to prepare for a repeat move. This is why the deceleration is not important. However, in cases where the SXF must perform electronic cam profiles, a deceleration's positional relationship may be necessary. In this case, the SXF can decelerate to a stop using the **FAC** and **FEN** values by setting the following ratio to 0 (**FOLØ**) while in Position Profile mode. To terminate this move, a stop command must be issued after the secondary axis reaches zero speed.

Position and Velocity Following Example

In this example, you will perform a preset move using following acceleration (with the parameters and motion requirements listed below).

❏ Primary axis encoder resolution = 4000 counts/rev

❏ Secondary axis motor resolution = 25000 steps/rev

❏ Maximum primary encoder speed = 2 rps

❏ Distance in primary encoder steps that the secondary axis must accelerate over = 2000 steps

❏ Desired speed ratio—**FOL** = 100% (1:1 ratio)

❏ Preset secondary axis move distance = 3 motor revolutions

❏ Encoder sample period—**TF** = 4 ms

A preset move of 75000 steps will be made. The secondary axis will accelerate over 2000 primary encoder steps up to the same speed as the primary encoder. A trigger will initiate motion. Follow these steps to perform the move profile. The encoder sampling period is set to the default of 4 ms.

Step ①

Determine the values for **FAC** and **FEN** from Equations 5-1 and 5-2.

$$\textbf{FEN} = Vp_{max} * \frac{\textbf{TF}}{1000} = 2\ \frac{revs}{sec} * 4000\ \frac{counts}{rev} * \frac{4}{1000}\ seconds = 32\ encoder\ counts$$

$$\textbf{FAC} = \frac{\textbf{FOL} * Vp_{max}}{Dp_{acc}} * \frac{\textbf{TF}}{1000} = \frac{100 * 8000\ \frac{counts}{sec}}{2000\ counts} * \frac{4}{1000}\ seconds = 1.6\ percent$$

Step ②

Enter the values for **FEN** and **FAC**.

Command	Description
> FEN32	Number of encoder counts of change required to increment the following percentage by **FAC**
> FAC1.6	The amount the following percentage increments for each **FEN** change in encoder counts

If you want to change the sample period of the primary encoder to 1 ms, scale both **FEN** and **FAC** by the change in the **TF** value. For example, if you go from 4 ms to 1 ms, divide **FAC** and **FEN** by 4 to get the following values.

FEN = 8

FAC = 0.4

If we went from **TF4** to **TF8**, multiply **FAC** and **FEN** by 2. ***Remember to change* FAC *and* FEN *if you change* TF**.

Step ③

Enable Following Acceleration mode.

Command	Description
> FSF1	Enables following acceleration

Step ④

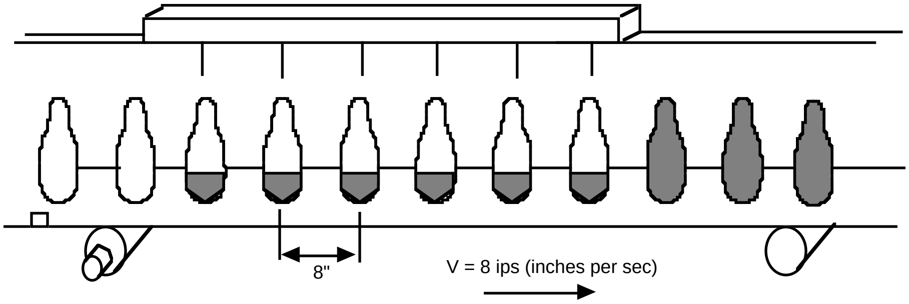
Start the primary axis into motion, then enter the commands below to perform the following acceleration move. If the primary axis' speed exceeds 2 rps, the following acceleration will not work properly.

Command	Description
> MN	Sets SXF to Normal mode
> FSI1	Enables following
> D75ØØØ	Sets the preset move distance to 75000 steps
> FOL1ØØ	Sets the following percentage to 100%
> G	Starts the following move

The secondary axis will accelerate over 2000 primary encoder steps. The secondary axis will move 1000 * **FOR** or 6250 motor steps over this acceleration ramp.

Following Acceleration Example: Bottle Filling

Typically, an application that requires position and velocity following will start the secondary axis from rest and accelerate it to 100% of the primary axis' speed (a 1:1 speed ratio). A trigger initiates motion on the secondary axis when a part or product is in a particular location on the primary axis. In this example, a conveyer belt moves bottles on a production line. The secondary axis is a filler that accelerates up to the conveyer line speed and fills the bottles. It fills six bottles at a time and then returns to the start point to fill six more.



Position & Velocity Following—Bottle Filling Application

One cycle of operation consists of the following steps.

① The secondary axis accelerates to the conveyer line speed.

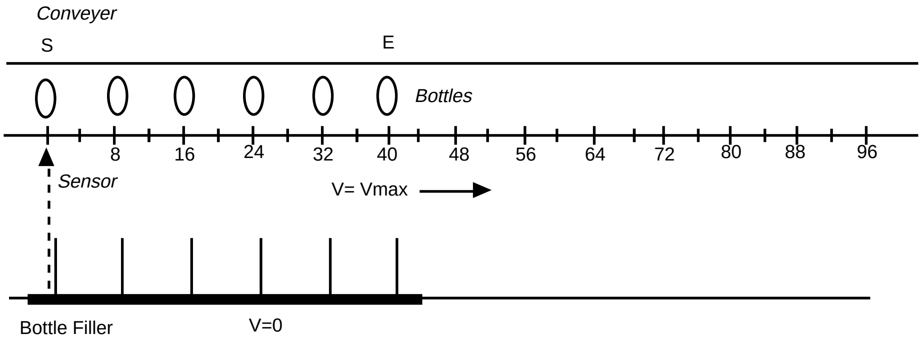
② The secondary axis enables an output that tells the dispenser to begin filling the bottles.

③ The secondary axis decelerates to a stop and returns to the starting point (at a high speed) to begin filling the next set of bottles.

In this application, the rate at which the bottles can be filled determines the maximum rate of the entire dispensing cycle. *Note the following information about the application.*

Maximum conveyer speed: 8 ips
Time to bill bottles: 2.5 seconds
Primary encoder resolution: 4000 counts/rev
Primary encoder linear resolution: 2 revs/inch = 8000 counts/inch
Distance between bottles: 8 inches
Bottle filler motor resolution: 25000 steps/revolution
Bottle filler linear resolution: 1 revolution/inch = 25000 steps/inch
Distance over which bottle filler accelerates: 2 inches

An output must be activated at the point that the bottle filler is moving at the same speed as the conveyer to initiate the dispensing of fluid into the bottles. A photoelectric sensor detects a bottle and begins the filling cycle. The SXF waits for the sensor as a trigger. When a bottle is detected, the SXF accelerates to the line speed, turns on an output, fills the bottle stops, and returns to the starting point to wait for the next trigger. The following figure depicts the conveyer and the filler axis at the start of a filling cycle.



Start of A Filling Cycle

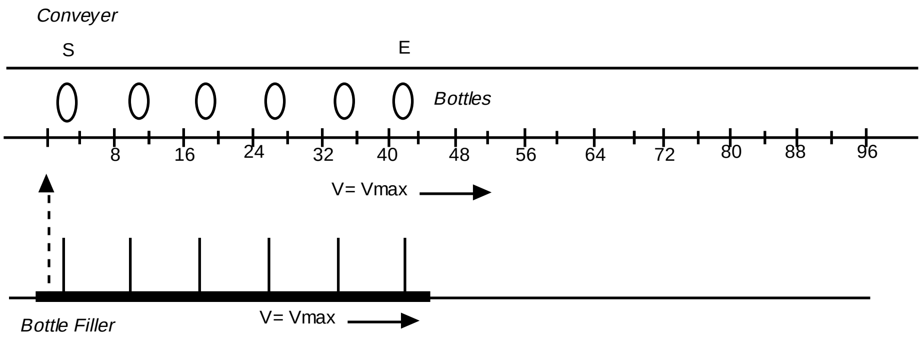
When the bottle marked **S** crosses the trigger sensor, the cycle begins.

① The secondary axis accelerates to the conveyer line speed.

② The secondary axis enables an output that tells the dispenser to begin filling the bottles.

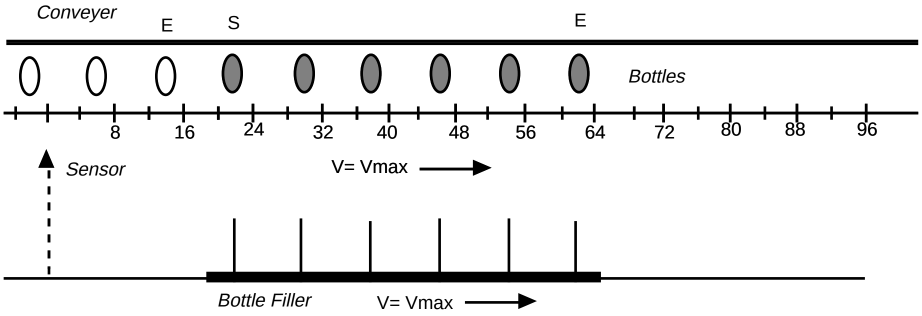
③ The secondary axis decelerates to a stop and returns to the starting point (at a high speed) to begin filling the next set of bottles.

This figure shows the two axis after the bottle filler reaches line speed and is ready to begin dispensing.



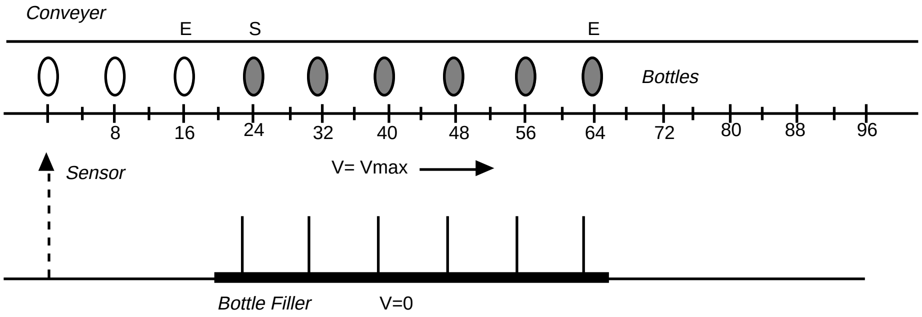
Bottle Filler Has Accelerated to Line Speed

A bottle can be filled in 2.5 seconds. At a maximum conveyer speed of 8 ips, the conveyer will move the bottles 20 inches. The next figure shows the bottle's location after the first six bottles have been filled. ***Note the location of the next set of bottles that will be filled***.



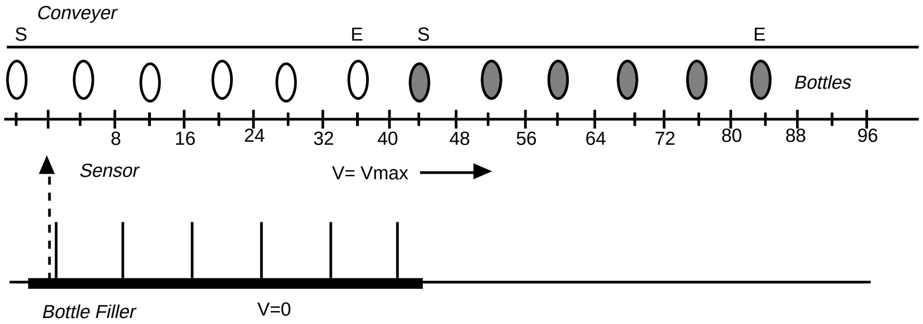
Dispensing is Completed

The bottle filler must now stop and return to the start location before the bottle marked **S** crosses the trigger point. The following figure shows the location of the bottle filler after it has stopped. It must now return 22 inches to the start before the bottles have moved 20 inches. The bottle filler will be in place, ready for the next trigger from the next set of bottles. It will arrive in place 4 inches before the next bottle. At 8 ips, 500 ms will elapse before the next bottle.



Bottle Filler Stops

To get back to the starting point before the conveyer has moved 20 inches, the bottle filler must return at a speed faster than the conveyer. It will accelerate at the same following acceleration set for the first part of the cycle. The following figure shows the bottle filler after it returns to the starting point. After the conveyer travels 4 more inches, the cycle will resume.



Bottle Filler Ready to Start a New Cycle

To program the application, use the following steps.

Step ①

Determine the **FOR** value.

Primary conveyer axis steps per inch = 8000

Secondary bottle filler axis steps per inch = 25000

$$\mathbf{FOR} = \frac{25000}{8000} = 3.125$$

Step ②

Determine the **FAC** and **FEN** following acceleration parameters. The encoder sample period time (**TF**) is 1 ms.

$$\mathbf{FEN} = 8000\,\frac{\text{steps}}{\text{inch}} * 8\,\frac{\text{inches}}{\text{second}} * \frac{1}{1000} = 64\text{ steps}$$

$$\mathbf{FAC} = \frac{100\% * 8000\,\frac{\text{steps}}{\text{inch}} * 8\,\frac{\text{inches}}{\text{second}}}{16000\ \text{steps}} * \frac{1}{1000} = 0.4$$

Step ③

Determine how far the primary and secondary axes will move during the dispensing part of the cycle. This includes the acceleration and deceleration parts of the move for the bottle filler when no fluid is dispensed.

$$Vp_{max} = 64000\,\frac{\text{steps}}{\text{sec}}$$

The dispensing takes 2.5 seconds. During the time that a constant following percentage occurs, the conveyer will have moved:

$$Dp_{con} = 64000\,\frac{\text{steps}}{\text{sec}} * 2.5\text{ sec} = 160000$$

The secondary axis will move this distance * **FOR**:

$$Ds_{con} = 160000\text{ primary encoder steps} * 3.125\,\frac{\text{secondary motor steps}}{\text{primary encoder steps}} = 500000\text{ secondary motor steps}$$

The secondary axis or bottle filler axis will accelerate over 2 inches of the conveyer at maximum speed and decelerate over two inches. If the conveyer moves slower, the dispensing part of the process will become a smaller percentage of the total cycle. The distance the secondary axis travels during acceleration will be one half of the distance the conveyer or primary axis travels. The secondary axis starts from rest and accelerates to match the primary axis' speed. The secondary axis will move 1 inch during acceleration and 1 inch during deceleration. The total distance of the bottle filler move is:

$$D_{sec} = 500000 + 2 * (1\,\text{inch} * 25000 \frac{\text{steps}}{\text{inch}}) = 550000\ \text{steps (secondary motor steps)}$$

The value to be entered for the **AD** command for deceleration is:

$$\mathbf{AD} = \frac{V^2}{2*D} = \frac{\left(8\frac{\text{inches}}{\text{sec}}\right)^2}{2*1\,\text{inch}} = 32\frac{\text{inches}}{\text{sec}^2} * 1\frac{\text{revolution}}{\text{inch}} = 32\frac{\text{revs}}{\text{sec}^2}$$

If the primary axis' speed changes, the SXF will still decelerate at this rate. The distance that the bottle filler moves will be identical, regardless of the primary axis' speed, so the bottle filler will still have to make a 22-inch move ***back*** while the primary axis moves 20 inches.

Step ④

You have determined the parameters necessary for the first part of the move. Next, you must determine the following percentage required to move the secondary axis back 22 inches while the primary axis moves 20 inches. Use Equation 5-4 to determine **FOL**.

Equation 5-4. FOL

$$\mathbf{FOL} = \frac{D_{prim}}{200*K} - \sqrt{\left(\frac{D_{prim}}{200*K}\right)^2 - \frac{D_{sec}}{\mathbf{FOR}*K}} \quad \text{Where } K = \frac{\mathbf{FEN}}{100*\mathbf{FAC}}$$

$$\mathbf{FOL} = \frac{160000}{200*1.6} - \sqrt{\left(\frac{160000}{200*1.6}\right)^2 - \frac{550000}{3.125*1.6}} = 500 - 374.2 = 125.8\%$$

125.8% is the return following percentage.

Step ⑤

Enter the sequence below to implement the motion.

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
FOR3.125	Sets motor to encoder steps per unit travel ratio
FAC.4	Sets the following acceleration increment to .4 % per encoder period
FEN64	Sets the encoder period, which increases the percentage to 64 steps
D55ØØØØ	Sets the secondary axis move for the cycle to 550000 motor steps
FSI1	Enables following mode
AD32	Sets the deceleration to 32 rps^2
IN1A	Defines input 1 as a trigger input
IN2D	Defines input 2 as a stop input
L	Starts a continuous loop
FOL1ØØ	Sets the initial following percentage to 100%
TR1	Waits on the input trigger
G	Starts motion
FOL125.8	Sets return move following percentage to 125.8%
H	Changes the direction
G	Starts the return following move
H	Changes direction again
N	Ends the loop—the following cycle will repeat
>XT	Ends the sequence

To decelerate using the **FAC** and **FEN** values, modify the program as follows:

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
SSH1	Sets save buffer on stop
FOR3.125	Sets motor to encoder steps per unit travel ratio
FAC.4	Sets the following acceleration increment to 0.4 % per encoder period
FEN64	Sets the encoder period, which increases the percentage to 64 steps
TF1	Sets the following encoder sample period to 1 ms
D55ØØØØ	Sets the secondary axis move for the cycle to 550000 motor steps
FSI1	Enables Following mode
AD32	Sets the deceleration to 32 rps^2
IN1A	Defines input 1 as a trigger input
IN2D	Defines input 2 as a stop input
L	Starts a continuous loop
FOL1ØØ	Sets the initial following percentage to 100%
TR1	Waits on the input trigger
MPP	Enters the Motion Profiling mode
G	Starts motion
FP176ØØØ	Waits until 176000 encoder pulses have passed
FOLØ	Stops the motion of the secondary
FP16ØØØ	Waits for the decel ramp distance
STOP	Stops the move itself
FOL125.8	Sets return move following percentage to 125.8%
H	Changes the direction
G	Starts motion
FP139782	Waits until 139782 encoder pulses have passed
FOLØ	Stops the motion of the secondary
FP2Ø218	Waits for the decel ramp distance
STOP	Stops the move itself
H	Changes direction again
N	Ends the loop—the following cycle will repeat
> XT	Ends the sequence

The value for **FP** during the return move is determined by calculating the distance the primary axis will move during acceleration and the constant following percentage portion and determining the distance it moves during the deceleration portion. This is determined from:

$$Dp_{acc} = \mathbf{FOL} * \frac{\mathbf{FEN}}{\mathbf{FAC}} = 125.8\% * \frac{64}{0.4} = 20218 \text{ encoder steps}$$

$$Dp_{dec} = Dp_{acc} = 20218 \text{ inches}$$

$$Dp_{con} = D_{prim} - Dp_{dec} - Dp_{acc} = 119564$$

$$D_{prim} = 20 \text{ inches} = 160000$$

The *first* **FP** = Dp$_{con}$ + Dp$_{acc}$ = 119564 + 20218 = 139782. The *second* **FP** = Dp$_{dec}$ = 20218. The second **FP** (20218 steps) measures the deceleration ramp. After deceleration, the move stops. The acceleration and deceleration ramps are based on the primary axis’ speed.

Recede and Advance While Following

Receding and advancing while following requires position and velocity following. In this type of application, the secondary axis follows the primary encoder at a 1:1 ratio or at the same speed. The secondary axis has a specific positional or phase relationship with the primary encoder. This type of application is used when multiple operations (such as welds) must be performed on one moving part. The operations are performed at various places on the part, requiring the secondary axis to advance or recede.

In an ***advance application***, the secondary axis must accelerate and move a specific distance beyond the primary axis, then decelerate to a 1:1 ratio. The secondary axis moves a specific distance with respect to the primary axis while both axes are moving. In a ***recede application***, the secondary axis decelerates until it recedes a specific distance behind the primary axis and then resumes a 1:1 speed ratio with the primary axis.

The point at which the advance move or recede move occurs is based on a specific position on the primary axis or on an input trigger. This type of application requires that the following ratio be changed on-the-fly while based on either an input or the primary encoder's position. It also requires that the secondary axis be able to move a specific distance while the primary axis moves a corresponding specific distance. In this manner, the secondary can advance or recede a specific distance with respect to the primary axis. *To change the following ratio on-the-fly*, you must use Motion Profiling (**MPP**) mode. You will need to measure the distance traveled by the primary encoder. Use the set of commands below.

Command	Description
> FPn	Delays processing for n primary encoder steps
> FPAn	Delays processing until the absolute count of the primary encoder reaches n
> VAR1=FEP	Allows you to read the value of the **F**ollowing **E**ncoder **P**osition into variable 1.

Advance Following Example

In this example, the primary axis has a 4000 count per revolution encoder. The secondary motor has a 4000 step per revolution motor. Therefore, the value for **FOR** is 1. The application requires that the secondary axis:

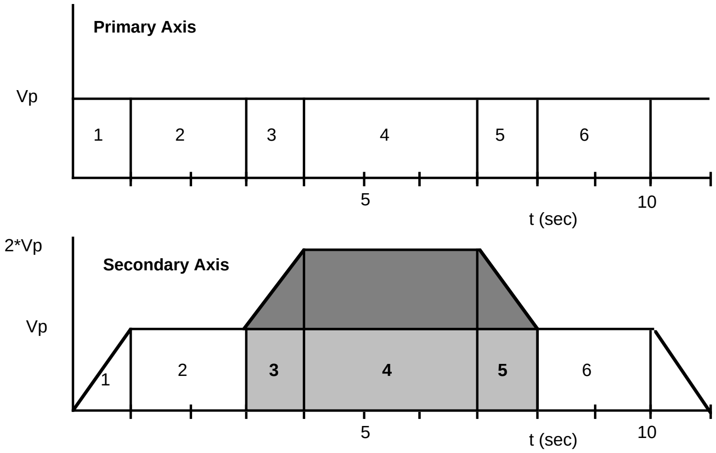
① Accelerate over 4000 primary encoder steps

② Move at a 1:1 ratio for 2 primary encoder revolutions

③ Advance (with respect to primary axis) 12000 primary encoder steps

④ Move 2 more primary encoder revolutions at 1:1 after advancing

⑤ Stop. The move profiles are shown below.



Advancing with Respect to the Primary While Following

The darkly shaded area in the previous figure represents the distance that the secondary axis advances with respect to the primary axis. The lightly shaded area represents the distance that both the primary and the secondary axes move during the ***advance portion*** of the profile. In these types of applications, the phase relationship or positional relationship is set with the **FAC** and **FEN** values (for as long as the following acceleration is performed). When the secondary axis accelerates from the 1:1 ratio to the 2:1 ratio, it will again accelerate at the following acceleration set. When it decelerates back to the 1:1 ratio, it decelerates in the same manner as it accelerates (using the **FAC** and **FEN** following acceleration rate). When the secondary axis decelerates back to *rest or zero speed*, it will decelerate at the **AD** rate. If the secondary axis must decelerate to zero speed at the FAC and FEN values, you must use **FOLØ** and set the following ratio to zero to make it stop.

Before programming this move profile, we will completely analyze the motion of the secondary and primary axes and then describe the sequence of commands necessary for performing the move.

In the previous figure, assume that the primary axis is moving at a maximum speed of 4000 steps/second. The area of each 1-second block represents 4000 primary encoder counts. Starting with Section #1, the primary encoder begins the section at a speed of $Vp_{max} = V_p$ = 4000 steps/sec. Therefore, the primary encoder moves 4000 counts during this section. From Section #1 of the secondary profile's plot, you can see that the secondary motor starts from rest and accelerates to a 1:1 speed ratio over 4000 counts of the primary encoder, Dp_{acc} = 4000 steps. You can calculate the desired **FAC** and **FEN** values to create such an acceleration ramp. This following acceleration will be used throughout the profile when changing from one following ratio to another. Set the encoder sample rate, **TF**, to 1 ms.

$$Vp_{max} = 4000 \text{ steps/sec}$$

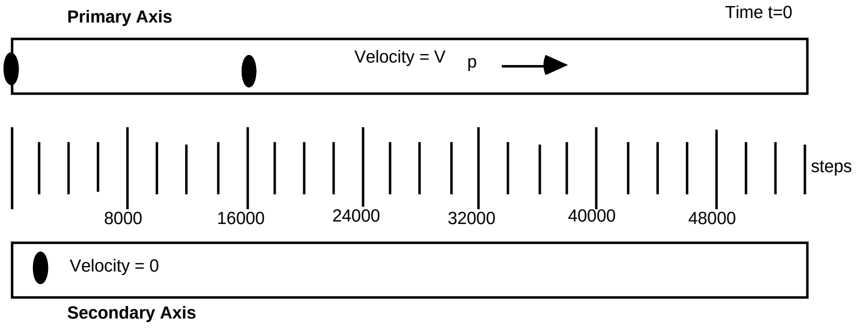
$$Dp_{acc} = 4000 \text{ steps}$$

$$\textbf{FEN} = Vp_{max} * \frac{\textbf{TF}}{1000} = 4000 \frac{\text{steps}}{\text{sec}} * \frac{1}{1000} \text{ sec} = 4 \text{ encoder counts}$$

$$\textbf{FAC} = \frac{FOL * Vp_{max}}{Dp_{acc}} * \frac{\textbf{TF}}{1000} = \frac{100\,\% * 4000 \frac{\text{steps}}{\text{sec}}}{4000 \text{ steps}} * \frac{1}{1000} \text{ sec} = 0.1 \text{ percent}$$

By calculating the area under the secondary axis profile curve, you can determine that the secondary motor has moved 2000 motor steps. If the secondary axis is accelerating to the same speed as the primary axis, it will always travel half of the primary axis' distance. In this manner, the secondary axis will be physically aligned with this point when it reaches a 1:1 speed ratio. The next figure shows two conveyer belts—primary and secondary axes. The primary and secondary axes' spots are at different locations at different times in the profile.

Advance Example



Advance Example at the Start of Section # 1

Usually, you will use an equation, not a graphic, to determine the distance traveled. Equation 5-5 calculates the distance the secondary axis travels for any acceleration, even if the secondary axis starts acceleration while it is already moving at a given ratio to the primary axis.

Equation 5-5 Ds_{acc}

$$Ds_{acc} = \textbf{FOR} * \left(\frac{1}{2} * \Delta\textbf{FOL}^2 * \frac{\textbf{FEN}}{100*FAC} + \Delta\textbf{FOL}*\textbf{FOL}_I * \frac{\textbf{FEN}}{100*\textbf{FAC}}\right)$$

FOL = The change in following percentage

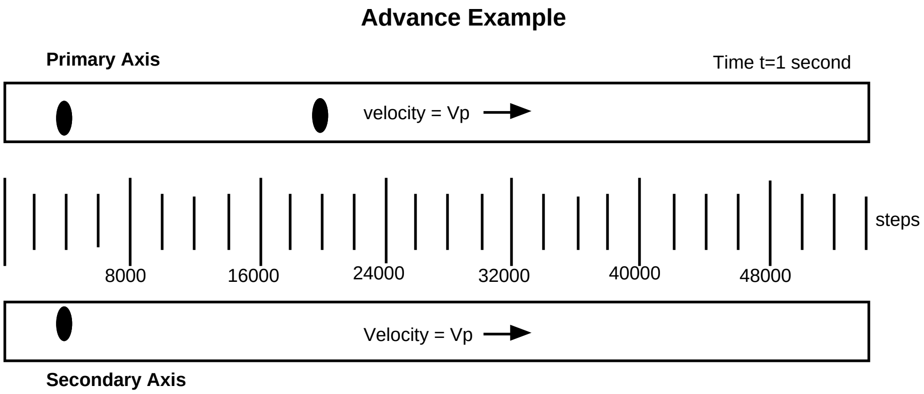
$\textbf{FOL}_I$ = The initial following percentage

In Equation 5-5, the change in following percentage is the difference between the final following percentage that you are accelerating to and the following percentage you are starting from. If you are starting from rest, the initial following percentage is Ø and the change in following percentage is **FOLØ** or **FOL**. If you are at **FOL1ØØ** and want to accelerate to a following percentage of **FOL2ØØ**, the change in following percentage is 100 and the initial following percentage is 100. **FAC** and **FEN** are the values calculated from the equations for **FAC** and **FEN** using the V_{max} and primary encoder acceleration distance.

The equation has two parts. The first part has the square of the following percentage change and the second part has a single following percentage change term. The first term determines the distance that the secondary axis travels due to the acceleration ramp portion of the curve. The second term gives the distance that the secondary axis travels due to its initial velocity. In this case, the secondary axis starts from rest (the second term contributes zero to the distance traveled). From the plot of the profile (Section # 1), the initial following percentage is Ø and the final percentage is 100, or the same speed as the primary axis. Using the general equation above, you should get 2000 steps, which is the area under the curve.

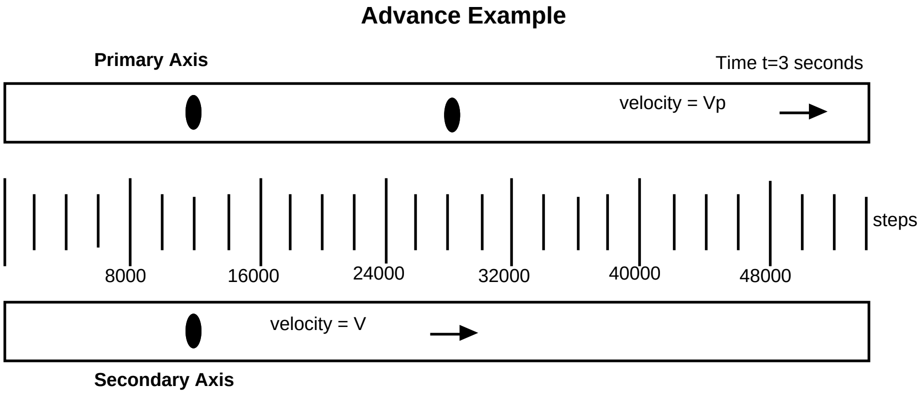
$$Ds_{acc} = 1* \quad \left(\frac{100^2}{2} * \frac{4}{100*0.1} + 100 * 0 * \frac{4}{100*0.1} \right) = \text{2000 secondary motor steps}$$

Therefore, in Section #1, the primary axis moved 4000 encoder counts and the secondary axis moved 2000 motor steps. We have determined the values for **`FAC`** and **`FEN`** based on the plots of the primary and secondary axes' profiles and the fact that V_{max} is 4000 steps/sec. At the end of Section #1, the spots will be in the locations shown in the following figure (t = 1 second).



Advance Example—End of Section #1

In Section #2, the primary axis is moving at V_{max} and the secondary axis is moving at the same speed because the ratio is 1:1 (**`FOL1ØØ`**). Section #2 lasts for 2 seconds. The primary axis travels 8000 steps during this section. The secondary axis travels 8000 steps too. This can be determined from the profile plot by calculating the area beneath the curve for the section. The primary and secondary axes are lined up at the start of Section #2 and they travel at a 1:1 ratio for the duration of the section. At the end of Section #2, the primary axis has moved 12000 steps and the secondary axis has moved 10000 steps (thru Sections #1 and #2). The next figure shows the relative location of the spots at the end of Section #2 (t = 3 seconds).

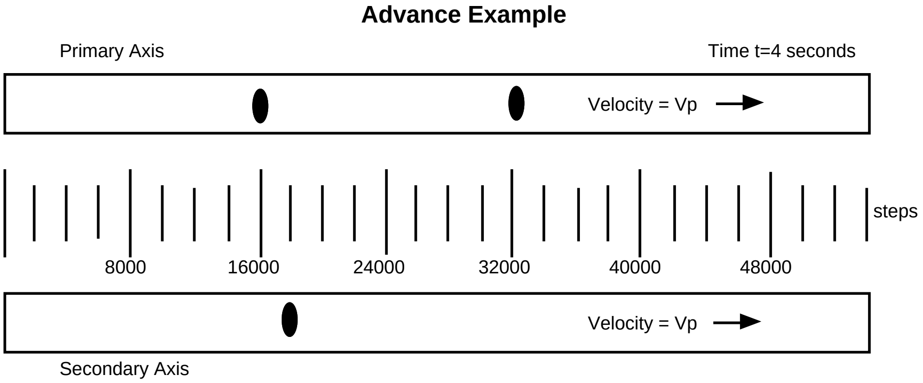


Advance Example—End of Section #2

In Section #3, the advance portion of the secondary profile begins. The secondary axis accelerates from a following percentage of 100% to 200% (a 2:1 ratio). Look at the plot of the profiles to graphically determine the distance that the primary and secondary axes have traveled. The primary axis moves 4000 steps and the secondary axis moves 6000 steps. The secondary axis' distance can also be determined from the previous equation..

$$Ds_{acc} = 1* \left(\frac{1}{2} * 100^2 * \frac{4}{100*0.1} + 100 * 100 * \frac{4}{100*0.1}\right) = 6000 \text{ steps}$$

At the end of Section #3, the primary axis has moved 4000 steps since the beginning of the advance portion of the profile. The secondary axis has advanced 2000 steps with respect to the primary axis. The following figure shows the location of the spots at the end of Section #3.



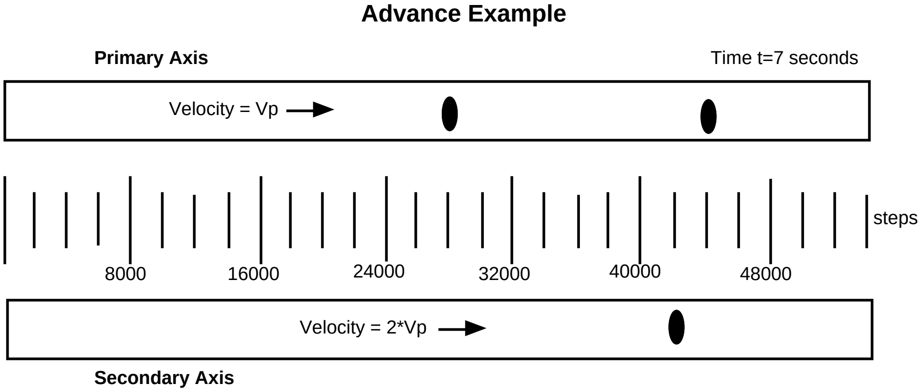
Advance Example—End of Section #3

At the start of Section #4, the secondary axis is at a following percentage of 200% and is moving at 2 * Vp (a 2:1 ratio). Section #4 lasts 3 seconds. The primary axis moves 12000 steps, while the secondary axis moves 24000 steps. The distance the secondary axis traveled can be determined from the equation below.

Equation 5-6. Ds_{con}

$$Ds_{con} = \frac{FOL}{100} * D_{prim} = 1 * \frac{200}{100} * 12000 = 24000$$

Since the start of the advance portion, the primary axis has moved a total of 16000 steps and the secondary axis has moved 30000 steps. The secondary axis has advanced 14000 steps with respect to the primary axis. The next figure shows the location of the spots at the end of Section #4.



Advance Example—End of Section #4

During Section #5, the secondary axis decelerates to a following percentage of 100% (a 1:1 ratio). After it decelerates, it will have completed the advance portion of the profile. The primary axis travels 4000 steps in Section #5. The secondary axis travels 6000 steps. When the secondary axis accelerates from one following percentage to another, then decelerates to the original following percentage (as in this example), ***the secondary axis' acceleration distance will always equal the deceleration distance***. However, the deceleration distance can also be calculated from the following equation.

Equation 5-7. Ds_{dec}

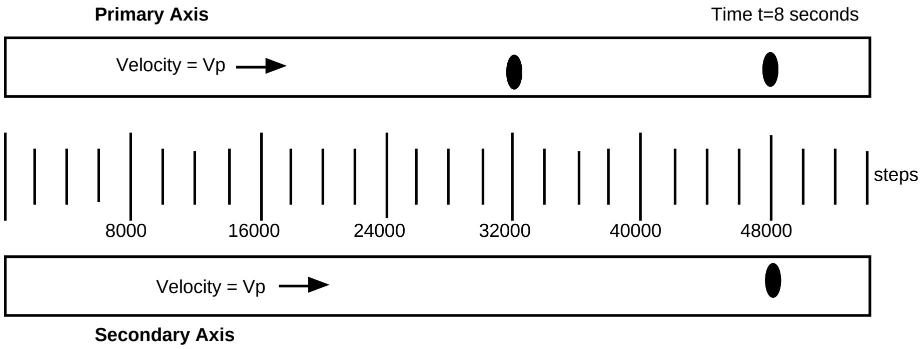
$$Ds_{dec} = \mathbf{FOR} * \left(\frac{1}{2} * \Delta\mathbf{FOL}^2 * \frac{\mathbf{FEN}}{100 * \mathbf{FAC}} + \mathbf{FOL} * \mathbf{FOL}_I * \frac{\mathbf{FEN}}{100 * \mathbf{FAC}}\right)$$

When decelerating, the initial following percentage is the percentage you are at when you begin deceleration. In this example, it is 200%. The final following percentage is 100%. Therefore, the change in following percentage is a negative number (-100%).

$$Ds_{dec} = 1 * \left(\frac{1}{2} * (-100)^2 * \frac{4}{100*0.1} + -100 * 200 * \frac{4}{100*0.1}\right) = -2000 + 8000 = 6000 \text{ steps}$$

At the end of Section #5, the advance portion of the move profile is complete. The secondary axis is moving at a 1:1 speed ratio with the primary axis. The secondary axis has moved 36000 steps and the primary axis has moved 20000 steps during the advance portion. The secondary axis has advanced 16000 steps with respect to the primary axis. The following figure shows the locations of the axes at the end of Section #5.

Advance Example



Advance Example—End of Section #5

During Section #5, the secondary axis travels at a 1:1 ratio until the deceleration ramp begins. If the move is a preset move, it will begin the deceleration ramp at the appropriate time based on the **AD** command. It will be at rest at the exact distance of the preset move. If the move is a continuous move, it will decelerate according to the **AD** value when a Stop (**S**) or Kill (**K**) command is encountered. From within a sequence, you can use the buffered Stop command. If you want the deceleration ramp to use the following acceleration value to decelerate, use the **FOLØ** command. After the secondary axis rests, a Stop command must be used to terminate the move. As depicted in the move profiles, the secondary axis moves at a 1:1 ratio for 8000 more steps, then decelerates to zero. During this time, the primary axis moves 8000 steps too.

Programming this profile requires the Following Encoder Distance Point (**FP**) or Following Encoder Absolute Point (**FPA**) command. **FP** is a delay-based on an incremental encoder distance. **FPA** is a delay-based on an absolute encoder distance.

FP Delay Example

To perform the advance move profile, breakpoints are needed to indicate when the secondary axis should accelerate to new following percentages. The **FP** and **FPA** commands define these breakpoints. These changes are performed on-the-fly and require the Motion Profiling mode. The breakpoints are the points at which acceleration or deceleration begin. This example will show the program using the **FP** command.

Step ① Set up the SXF with the appropriate encoder interface and enable the Following mode.

Command	Description
> FSI1	Enables SXF's Following mode

Step ② Set up the velocity following portion of the application. The number of secondary motor steps per unit of travel is 4000. The number of primary encoder steps per unit of travel is also 4000. Therefore, the **FOR** value is 1. The **FOL** command will be set up in the sequence for running the profile.

Command	Description
> FOR1	Relates the number of secondary motor steps for a distance to the number of primary encoder pulses for the same distance

Step ③ Set-up the following acceleration value and enable Following Acceleration mode. The values chosen are the same as were calculated in the explanation of the example above.

Command	Description
> FAC.1	Increases the following percentage to 0.1 for every change in encoder pulses by FEN
> FEN4	Sets the number of encoder pulses required before the following percentage is incremented by FAC.
> FSF1	Enables the use of following acceleration

Step ④ The breakpoints in the desired profile occur at the end of Section #2, the end of Section #4, and if **FAC** and **FEN** are used for deceleration, a breakpoint is set at the end of Section #6. If your **FOR** command differs from this example, the distance command is entered in terms of your secondary motor. The following sequence performs the desired profile. Each step of the sequence is explained. Enter the sequence.

Command	Description
> XE1	Erases Sequence #1
> XD1	Begins the definition of Sequence #1
D56ØØØ	The total distance the secondary moves is 56000 steps.
FOL1ØØ	The first following percentage to accelerate to is 100%.
MPP	Enters Motion Profiling mode so changes can be made on the fly.
G	The secondary motion begins.
FP12ØØØ	The first breakpoint occurs after the **primary axis** moves 12000 steps. **FP** causes command processing in the sequence to delay until 12000 primary encoder steps have been counted.
FOL2ØØ	After 12000 primary encoder steps the following percentage is changed to 200%. The **secondary axis** begins to accelerate at the following acceleration. Distance is known from previous analysis.
FP16ØØØ	Right after the command to begin acceleration to an **FOL** of 200%, the command processing is delayed until the **primary axis** has moved 16000 more pulses (from when **FP** is encountered and thus is an incremental encoder distance).
FOL1ØØ	After 16000 more primary encoder pulses, the following percentage changes to 100% and the **secondary axis** decelerates to a 1:1 ratio.
FP12ØØØ	Command processing is delayed 12000 more steps.
FOLØ	The secondary axis decelerates to an FOL of 0.
FP4ØØØ	It takes 4000 primary encoder steps to decelerate.
STOP	A Stop command is needed because the SXF would still think it was in a move and that the current following ratio was set to zero.
NG	Exits the Motion Profiling mode.
XT	Ends the definition of Sequence #1

Step ⑤ Typically, in an application that requires velocity and position following, it does not matter how you decelerate to a stop (you are moving from a synchronized state to rest). At the end of such a move, you will need to reverse direction and return to the starting location to repeat the profile. In this case, the optimal profile is as follows:

① Accelerate to a known positional relationship

② Perform the operation required at the synchronized speed

③ When it is complete, decelerate as fast as possible to repeat the cycle

You must decelerate at a rate unrelated to the primary encoder speed. The SXF allows you to do this. This will also simplify the programming. The same profile is programmed below using the **AD** deceleration value.

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
D56ØØØ	The total distance the secondary moves is 56000 steps
FOL1ØØ	The first following percentage to accelerate to is 100%
MPP	Enters Motion Profiling mode so changes can be made on the fly
G	The secondary axis' motion begins
FP12ØØØ	Delays processing until primary axis moves 12000 encoder pulses
FOL2ØØ	Change to 200% following percentage
FPA16ØØØ	Delays processing until primary axis moves 16000 encoder pulses
FOL1ØØ	Change ratio back to 100%
NG	Exits the Motion Profiling mode
XT	Ends sequence #1 definition

In this sequence, only two breakpoints are needed, the breakpoint to accelerate to 200% and then to decelerate back to 100%. Since the SXF will decelerate at the value in **AD**, it will automatically decelerate to a distance of exactly 56000 secondary motor steps at the appropriate time.

Step ⑥ This step uses **FPA** instead of **FP**. **FPA** delays processing until the absolute value of the following encoder counter exceeds the **FPA** value.

Command	Description
> XE1	Erases Sequence #1
> XD1	Defines Sequence #1
D56ØØØ	The total distance the secondary moves is 56000 steps.
PFZ	Zero the following encoder counter.
FOL1ØØ	The first following percentage to accelerate to is 100%.
MPP	Enters Motion Profiling mode so changes can be made on the fly.
G	The secondary motion begins.
FPA12ØØØ	Delay command processing until primary encoder count exceeds 12000 encoder pulses.
FOL2ØØ	Change to 200% following percentage.
FPA28ØØØ	Delay command processing until primary encoder counter exceeds 28000 encoder pulses or an incremental change of 16000 pulses.
FOL1ØØ	Change ratio back to 100%.
NG	Exits the profiling mode.
XT	Ends Sequence #1 definition

In Steps 5 and 6, the encoder is counting in the positive direction. If the encoder is counting in the negative direction, a negative sign is required for both the **FP** and **FPA** commands. Use either the **FP** or the **FPA** command, depending on your application.

*Use **FPA** when repetitive cycles of the same move profile are done without a trigger to start each cycle*. By making the delays dependent on an absolute encoder position, there is no accumulative error. In many cases, a trigger input from a sensor is used to start the move profile that is repeated.

*If you use a trigger to start the move each time, use **FP** and the trigger will remove any* accumulative *error*. The following sequence illustrates the uses of the **FPA** command and a variable to perform a repetitive move that does not use a trigger to start it. This case is more like a cam cycle and the position relationship must be maintained while the cycle repeats.

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
VAR1=12ØØØ	Sets variable 1 equal to the first breakpoint.
VAR2=28ØØØ	Sets variable 2 equal to the second breakpoint
VAR3=Ø	Sets variable 3 equal to the primary reference point
D56ØØØ	Sets distance of secondary axis move to 56000 steps
PFZ	Zeroes the following encoder counter
L	Begins the continuous loop of the profile cycle
FOL1ØØ	The first following percentage to accelerate to is 100%
MPP	Enters Motion Profiling mode so changes can be made on-the-fly
FPA(VAR3)	Variable 3 synchronizes the move start with the primary axis. For the first 40000 primary steps, the secondary axis is moving, then it moves back during the next 40000 steps of the primary then it repeats.
G	Begins secondary axis motion.
VAR3=VAR3+8ØØØØ	Sets variable 3 equal to the start of the next cycle
FPA(VAR1)	Delays command processing until primary encoder count exceeds 12000 + 56000*n* encoder pulses, *n* = # of times through the loop.
FOL2ØØ	Changes following percentage to 200%.
VAR1=VAR1+8ØØØØ	Sets variable 1 equal to the first breakpoint for the next cycle
FPA(VAR2)	Delays command processing until primary encoder count exceeds 28000 + 56000*n* encoder pulses, *n* = the # of times through the loop.
FOL1ØØ	Changes ratio back to 100%.
VAR2=VAR2+8ØØØØ	Sets variable 2 equal to the second breakpoint for the next cycle
NG	Exits the profiling mode and complete the 56000 step move
FOL2ØØ	Sets the following ratio to a higher speed to move back to the starting point at a fast speed
H	Changes direction
G	Moves back to the starting point.
H	Changes direction
N	Repeats the cycle
XT	Ends sequence #1 definition

In this example, the move profile is repeated. One cycle consists of the following events:

① Secondary axis moves 56000 steps while primary axis moves 40000 steps.

② The secondary axis retreats to the start and after another 40000 primary encoder steps the cycle is repeated.

③ No operation during the secondary axis' retreat.

④ The retreat is set to a high following ratio to get the secondary axis back to the start before the primary axis moves 40000 steps.

This cycle is very similar to a cam cycle (which will be described in the next section). You can load the **FP** command with a variable (like the **FPA** command). You can check the following encoder counter value at any time by loading it into a variable.

Command	Description
> VAR1=FEP	Loads variable 1 with the value of the following encoder counter

Calculating FOL, FP, or FPA For An Advance or Recede Application

The advance example explained how an advance move is made and how the different commands (**FOL**$_I$, **FAC** and **FEN**) contribute to the move. This section provides some simple formulas that you can use to set up such a profile. To do position and velocity following, you must be able to accelerate the secondary axis to a known position with respect to the primary axis. This is what determines the **FAC** and **FEN** following acceleration values. Once you determine these values, you will use them to calculate the acceleration ratio that you must use to make the advance move. The following information will help you understand the move profile in the example.

❑ **FAC**: Following speed percentage increment

❑ **FEN**: Change in primary encoder pulses to cause an increment of **FAC**

- **D**$_{\mathbf{prim}}$: The distance the primary axis will travel during the advance portion of the secondary move profile. This is 20000 primary encoder steps in the example above.
- **D**$_{\mathbf{sec}}$: The distance the secondary must advance with respect to the moving primary, measured in primary encoder steps. In the example above this is 16000 primary encoder steps.
- **FOL**$_{\mathbf{I}}$: The initial following percentage that you will accelerate from to the new following percentage.

In applications that require an advance move, you will usually know the distance that you want to advance with respect to the primary axis and the distance the primary axis will move during the advance. The distance that you want the secondary axis to advance with respect to the primary axis is given in terms of primary encoder steps. The distance can be converted from secondary motor steps to primary motor steps (and vice versa) with the **FOR** command. After determining the parameters listed above, you can use the following formula to determine the following percentage you must accelerate to. First, determine a following acceleration constant (**K**) to simplify the equations.

Equation 5-8. Following Constant

$$\mathbf{K} = \frac{\mathbf{FEN}}{100*\mathbf{FAC}}$$

The constant **K** is used in Equation 5-9 to determine **FOL**.

Equation 5-9. FOL

$$\mathbf{FOL} = \mathbf{FOL_I} + \frac{D_{prim}}{200*K} - \sqrt{\left(\frac{D_{prim}}{200*K}\right)^2 - \left(\frac{D_{sec}}{K}\right)}$$

With an advance move, the value of **FOL**$_{\mathbf{I}}$ will always be 100. Apply the formula to the example above (the following percentage should be 200%). This is the following percentage that you must attain to advance 16000 steps with respect to the primary axis, while the primary axis moves 20000 steps.

D_{prim} = 20000 primary encoder steps

D_{sec} = 16000 primary encoder steps

FAC = 0.1

FEN = 4

FOL$_{\mathbf{I}}$ = 100

We will first determine the following acceleration constant K.

$$K = \frac{4}{100*0.1} = 0.4$$

We will now determine **FOL**.

FOL is the same as $\mathbf{FOL}_F$ in the equations used earlier for determining the distances traveled by the primary and secondary axes.

$$\mathbf{FOL} = 100 + \frac{20000}{200*0.4} - \sqrt{\left(\frac{20000}{200*0.4}\right)^2 - \left(\frac{16000}{0.4}\right)}$$

$$= 100 + 250 - \sqrt{(250)^2 - 40000} = 350 - \sqrt{22500} = 350 - 150 = 200 = \mathbf{FOL}_F$$

The following percentage that must be accelerated to is 200%. Now calculate what the breakpoint is for decelerating back to a 1:1 ratio or 100%. The **FP** value will be determined from the Equation 5-10.

Equation 5-10.
Following Breakpoint

$$\mathbf{FP} = D_{prim} - \frac{\mathbf{FEN}}{\mathbf{FAC}} * \left(\mathbf{FOL}_F - \mathbf{FOL}_I\right) = 20000 - \frac{4}{0.1} * (200 - 100) = 16000 \text{ steps}$$

The value you would use for **FP** is 16000. The breakpoint at which you begin the advance portion of the move profile was not calculated. This value varies from application to application and you may want to use a trigger to begin the advance move rather than **FP**. An example of using a trigger to begin an advance move is described below.

Configure an input as a trigger input with the **IN** command.

Command	Description
> IN1A	Configures input #1 as a trigger input.
> IN2D	Configures input #2 as a stop input.
> XD1	Defines sequence #1.
MC	We will make this a continuous move.
FOL1ØØ	The first following percentage to accelerate to is 100%.
MPP	Enters Motion Profiling mode so changes can be made on-the-fly.
G	Initiates motion.
TR1	Command processing pauses until input #1 (trigger input) is activated. The secondary axis will move continuously at a speed percentage of 100% (with respect to the primary axis).
FOL2ØØ	The following percentage is changed to 200%. Acceleration begins.
FP16ØØØØ	Command processing will delay 16000 primary encoder steps.
FOL1ØØ	The following percentage is changed to 100%. Deceleration begins.
NG	Exits Motion Profiling mode.
XT	Ends sequence #1 definition.

In this sequence, the secondary axis will begin moving at a 100% speed percentage. When trigger input #1 is activated, the secondary axis will advance 16000 steps with respect to the primary axis. It will then decelerate to a 100% or 1:1 ratio and continue until the stop input (input #2), is activated (or a Stop [**S**] command is issued).

Recede vs Advance

Recede moves are similar to advance moves. In the illustration of the spots for the advance example, the secondary axis synchronized with the first primary axis, then receded while the primary axis moved. This motion can be analyzed in the same manner as the advance move with the exception that a different equation is used to determine the required value for **FOL**. Again, you will have to provide the distance that the primary axis will move during the recede move and the distance with respect to the primary axis that the secondary axis must recede (measured in terms of primary encoder steps). For example:

❏ A primary encoder has a resolution of 4000 steps/rev (1 rev = 1 inch)

❏ The secondary motor has 25000 steps/revolution and also has 1 rev = 1 inch

❏ The **FOR** command is set to 6.25.

Typically you will know what distance you want the secondary axis to recede. If the secondary is to recede 1.5 inches with respect to the primary axis while the primary axis moves 3.5 inches, set D_{prim} and D_{sec} equal to:

$$D_{prim} = 3.5'' = 3.5'' * 4000 \frac{\text{steps}}{\text{inch}} = 14000 \text{ primary steps}$$

$$D_{sec} = 1.5'' = 1.5'' * 25000 \frac{\text{steps}}{\text{inch}} = 37500 \text{ secondary motor steps} = \frac{37500}{6.25} = 6000 \text{ primary encoder steps}$$

Both distances are provided in primary encoder steps. The two terms are used to determine the required **FOL** in Equation 5-11.

Equation 5-11. FOL

$$FOL = FOL_I - \frac{D_{prim}}{200*K} + \sqrt{\left(\frac{D_{prim}}{200*K}\right)^2 - \frac{D_{sec}}{K}}$$

Where K is the following acceleration constant (as in Equation 5-8).

$$K = \frac{FEN}{100 * FAC}$$

To calculate the **FOL** value, determine a value for **FAC** and **FEN**. This will depend on your application’s maximum velocity and either the maximum acceleration for the secondary axis or the distance the primary axis travels while the secondary axis must accelerate. Use the same **FAC** and **FEN** values from the advance example:

FAC = 0.1

FEN = 4

In an advance or recede application, the initial following percentage $\mathtt{FOL}_I$ will always be 100.

The value for **FOL** produced by the equation above is:

$$\mathbf{K} = \frac{4}{100*0.1} = 0.4$$

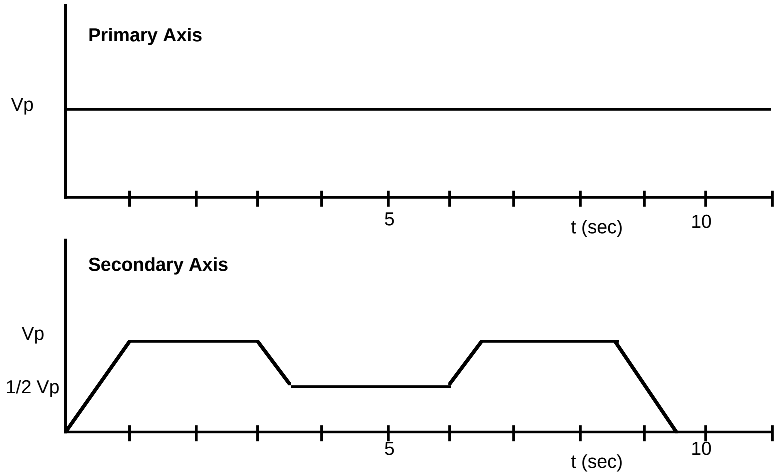
$$\mathbf{FOL} = 100 - \frac{14000}{200*0.4} + \sqrt{\left(\frac{14000}{200*0.4}\right)^2 - \frac{6000}{0.4}} = 100 - 175 + \sqrt{(175)^2 - 15000} = 50\%$$

The value needed for **FP** can be determined from Equation 5-12.

Equation 5-12. FP

$$\mathbf{FP} = D_{prim} + \frac{\mathbf{FEN}}{\mathbf{FAC}} * \left(\mathbf{FOL}_F - \mathbf{FOL}_I\right) = 14000 + \frac{4}{0.1} * (50 - 100) = 12000 \text{ primary enc. steps}$$

The move profile is shown below.



Recede While Following Profile

The sequence that will execute this profile is provided below:

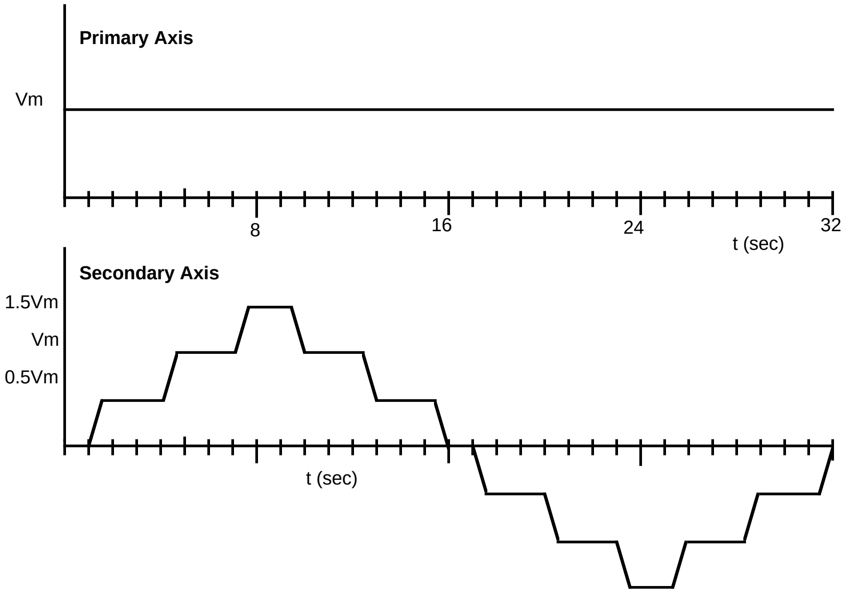
Command	Description
> FACØ.1	Sets the percentage increment to 0.1
> FEN4	Sets the number of encoder counts for an increment to 4
> FSF1	Enables following acceleration
> FOR6.25	Sets the secondary motor steps per unit distance to primary encoder steps per unit distance ratio
> XD1	Defines sequence #1
MN	Enters Normal mode
D175ØØØ	Sets distance to 175000 steps
FOL1ØØ	The first following percentage to accelerate to is 100%
MPP	Enters Motion Profiling mode so changes can be made on-the-fly
G	Initiates motion
FP12ØØØ	The command processing will pause here until the primary encoder has moved 12000 steps. The secondary will then decelerate to 50%
FOL5Ø	Changes following percentage to 50%—the recede portion begins
FP12ØØØ	Command processing will delay 12000 primary encoder steps.
FOL1ØØ	The following percentage is changed to 100%—deceleration begins
NG	Ends the Motion Profiling mode
XT	End sequence definition

Cam Following

A common application that requires velocity and position following is the simulation of a cam or an electronic cam. To simulate the motion produced by a cam, you must satisfy the following requirements:

- ❏ Follow both the position and the velocity of a primary encoder.
- ❏ You must also be able to change following ratios during motion and still maintain a positional relationship.
- ❏ Change ratios based on primary encoder distance.
- ❏ Must be able to keep track of the primary encoder position even if the secondary axis is not moving.
- ❏ Must be able to continuously repeat a cam cycle without developing accumulative error.

You can simulate a cam profile electronically using the commands and equations developed earlier. Motion Profiling mode (**MPP**) is required for cam following. For more information on Motion Profiling mode, refer to *Chapter 4, Application Design*. The next figure shows a typical cam profile.



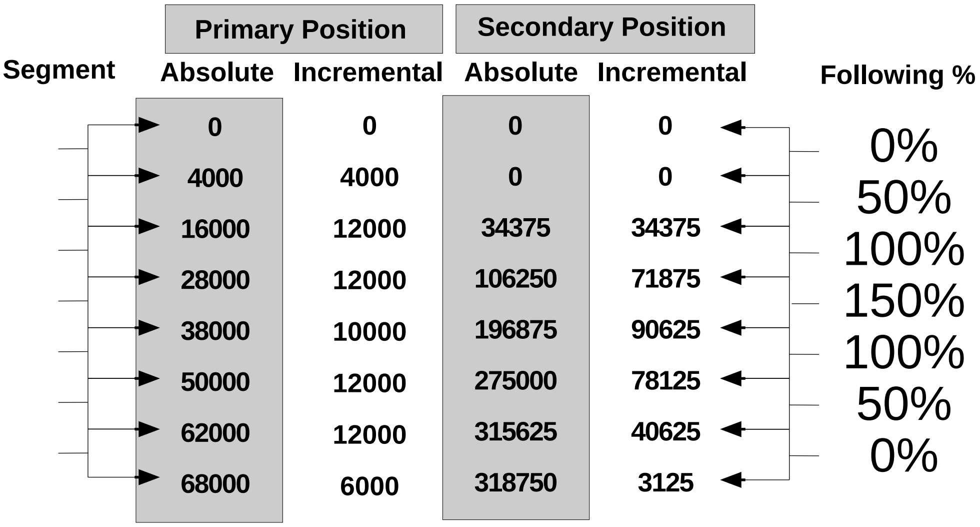
Cam Following

In this example, the encoder is a 4000 pulse per revolution encoder and it is mounted on the primary axis. The ***secondary axis*** will perform an electronic cam cycle, which consists of the following steps:

① Sitting at rest for one primary encoder revolution, then executing a step profile of 3 following percentages from 50% to 100%, then to 150%, and back down to 0%.

② It will then delay 1 primary encoder revolution and perform the same profile in the opposite direction.

③ This cycle is to be repeated until a stop is issued. The secondary axis will accelerate at its maximum rate when the primary axis is at its maximum velocity. Table 5-1 defines the cam cycle for this profile.



Segment	Primary Position		Secondary Position		Following %
	Absolute	Incremental	Absolute	Incremental	
	0	0	0	0	0%
	4000	4000	0	0	50%
	16000	12000	34375	34375	100%
	28000	12000	106250	71875	150%
	38000	10000	196875	90625	100%
	50000	12000	275000	78125	50%
	62000	12000	315625	40625	0%
	68000	6000	318750	3125	

Cam Cycle

The following percentage is given for each segment. Each of the distance points is a breakpoint where the following percentage changes. Use the following acceleration to change from one following percentage to the next. The data needed to program this type of profile is listed below.

Vp_{max}: Maximum velocity of the primary axis

As_{max}: Maximum acceleration of the secondary axis

FOR: Relative resolutions per unit of distance for the primary and secondary axes

FAC: Following acceleration value, percentage increments

FEN: Following acceleration value, encoder counts for an increment

For each ratio segment, you must know the distance the primary axis will travel and the corresponding distance the secondary axis will travel. The breakpoints can be determined from the graph. Usually, you will not be able to graphically describe the motion relationship, and will simply know that you want the primary axis to move ***x*** steps and the secondary axis to move ***y*** steps in the same time frame.

After you determine **FAC** and **FEN** from your maximum acceleration and maximum velocity, or from the distance you want the secondary axis to accelerate over and the maximum velocity, you can use **FAC** and **FEN** and the primary and secondary axes' travel distances to determine the following percentages and the values for the breakpoints to change to new following percentages. When accelerating to a higher following percentage, you can use Equation 5-13 to determine the required **FOL** percentage that you must accelerate to using the **FAC** and **FEN** values you have determined.

Equation 5-13.
FOL ACCEL

$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{100*K} - \sqrt{\left(\frac{D_{prim}}{100*K}\right)^2 + \frac{2*\mathbf{FOL}_I*D_{prim}}{100*K} - \frac{2*D_{sec}}{\mathbf{FOR}*K}}$$

Where K is the following acceleration constant determined by **FAC** and **FEN** in Equation 5-8.

$$K = \frac{\mathbf{FEN}}{100*\mathbf{FAC}}$$

If you are decelerating to a lower following percentage, use Equation 5-14 to determine the following percentage you must use to move the secondary axis the specified number of steps for the corresponding motor steps.

Equation 5-14.
FOL DECEL

$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{100*K} + \sqrt{\left(\frac{D_{prim}}{100*K}\right)^2 - \frac{2*\mathbf{FOL}_I*D_{prim}}{100*K} + \frac{2*D_{sec}}{\mathbf{FOR}*K}}$$

In both the accelerating and decelerating equations, the terms listed below for the primary encoder steps and the secondary motor steps are required.

D_{prim} = The number of primary encoder steps that the motor will move in the segment.

D_{sec} = The number of corresponding secondary motor steps that the secondary will move during which time the primary encoder will move D_{prim}.

To illustrate the programming of the profile above, we will assign values to the application's requirements. The maximum velocity, acceleration and the **FOR** value are to be determined by the application. The **FAC** and **FEN** values are calculated. The maximum velocity and acceleration are given below. The example has the following parameters:

Primary encoder resolution	4000 steps/revolution
Secondary motor resolution	25000 steps/revolution
1 encoder rev	1 motor rev

FOR 6.25

V_{pmax} = 4000 steps/second

As_{max} = 1 rev/second ^{2}

Encoder sample period TF = 1 ms

From the equations in the *Velocity and Position Following* section **FAC** and **FEN** are determined:

$$\mathbf{FEN} = V_{pmax} * \frac{\mathbf{TF}}{1000}$$

$$\mathbf{FAC} = \frac{\mathbf{FOL}*Vp_{max}}{Dp_{acc}} * \frac{\mathbf{TF}}{1000}$$

Since we are starting with Vp_{max} and As_{max}, we must determine D_{acc} for the equations above.

$$Dp_{acc} = \frac{Vp_{max}{}^2 * \mathbf{FOR} * \mathbf{FOL}}{As_{max} * 100} = \frac{\left(4000\,\frac{steps}{sec}\right)^2 * 6.25 * 100}{25000\,\frac{steps}{sec^2} * 100} = 4000 \text{ primary encoder steps}$$

Vp_{max} is in primary enc. steps/sec

As_{max} is in secondary motor steps/sec^2

The **FOR** term converts the acceleration units to primary encoder steps units. **FAC** and **FEN** can now be calculated.

$$\textbf{FEN} = 4000 \ \frac{\text{steps}}{\text{sec}} * \frac{1 \ \text{ms}}{1000} = 4 \text{ steps}$$

$$\textbf{FAC} = \frac{100 \ * \ 4000\frac{\text{steps}}{\text{sec}}}{4000 \ \text{steps}} * \frac{1 \ \text{ms}}{1000} = 0.1$$

You can now use these equations to determine the **FOL** value for each segment of primary encoder distance and secondary motor distance. The **FOL** values are already given in the *Cam Cycle* table, but it will be illustrative to determine the **FOL** required for some of the segments. Evaluate **FOL** for Segments #4 and #6. Typically, you will know the distance you want the primary axis to move and the corresponding distance that you want the secondary axis to move. You will have to enter the **FOL** values for the table from the equations given in this chapter.

Segment 4

D_{prim} = 10000 primary encoder steps

D_{sec} = 90625 secondary motor steps

Using the equation for accelerating, we can evaluate for **FOL** in Segment #4.

The acceleration constant $K = \frac{4}{100*0.1} = 0.4$

$$\textbf{FOL} = 100 + \frac{10000}{100 \ * \ 0.4} - \sqrt{\frac{2*100*10000}{100*0.4} + \left(\frac{10000}{100 \ * \ 0.4}\right)^2 - \frac{2*90625}{6.25*0.4}}$$

$$= 100 + 250 - \sqrt{50000 + (250)^2 - 72500} = 150\%$$

The breakpoint is given automatically by the table and is 10000 for FP and is 38000n for **FPA** where n is the number of cycles completed thus far.

Segment 6

D_{prim} = 12000 primary encoder steps

D_{sec} = 40625 secondary motor steps

Using the equation for decelerating to a lower percentage we can calculate the value for **FOL** for Segment 6.

$$\textbf{FOL} = 100 - \frac{12000}{100*0.4} + \sqrt{\left(\frac{12000}{100*0.4}\right)^2 - \frac{2*12000*100}{100*0.4} + \frac{2*40625}{6.25*0.4}}$$

$$= 100 - 300 + \sqrt{(300)^2 - 60000 + 32500} = 50\%$$

The breakpoints for **FP** and **FPA** are 12000 and 62000***n*** steps respectively. In many cases involving a cam cycle, a trigger is not used to start each cycle and the repetition of the cycle is based on the primary encoder. In these situations, use the **FPA** command (it is based on the following encoder's absolute count). The absolute count comes from a hardware counter that can be accessed by assigning it to a variable:

```
> VAR1=FEP            FEP is the value in the hardware counter, it is a read only value
```

The following sequence will perform the cam profile. The secondary axis will be put in a continuous move. Two parts occur in a cycle. The first part moves the stepped profile in one direction. The second part reverses direction and returns to the start to repeat the cycle.

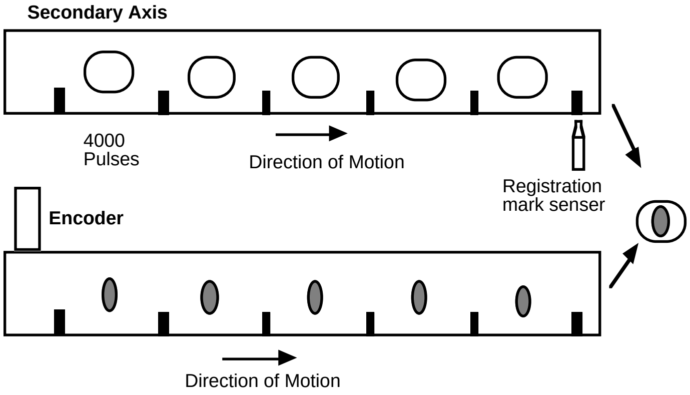
Command	Description
> FOR6.25	Sets the secondary motor steps to primary encoder steps ratio
> FAC0.1	Sets the change in following percentage for following acceleration
> FEN4	Sets the number of encoder pulses required to change by **FAC**
> FSF1	Enables following acceleration
> FSI1	Enables following
> SSH1	Saves buffer on stop
> VAR1=64ØØØ	Variable for the incrementing the cycle
> VAR2=4ØØØ	Breakpoint 1
> VAR3=16ØØØ	Breakpoint 2
> VAR4=28ØØØ	Breakpoint 3
> VAR5=38ØØØ	Breakpoint 4
> VAR6=5ØØØØ	Breakpoint 5
> VAR7=62ØØØ	Breakpoint 6
> VAR8=64ØØØ	Breakpoint 7
> 1XE1	Erases sequence #1
> 1XD1	Defines sequence #1
MC	Enables continuous mode
FOLØ	Sets the current following percentage to 0%
L	Begins the loop cycle
MPP	Enters the Motion Profiling Mode
G	Initiates motion
FPA(VAR2)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 1
FOL5Ø	Following % is changed to 50% or 1/2 as fast as the primary motor
VAR2=VAR2+VAR1	Set variable 2 to the breakpoint value for the next cycle
FPA(VAR3)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 2
FOL1ØØ	Speed ratio is changed to 1:1
VAR3=VAR3+VAR1	Set variable 3 to the breakpoint value for the next cycle
FPA(VAR4)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 3
FOL15Ø	Speed ratio is changed to 1.5:1
VAR4=VAR4+VAR1	Set variable 4 to the breakpoint value for the next cycle
FPA(VAR5)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 4
FOL1ØØ	Speed ratio is changed to 1:1
VAR5=VAR5+VAR1	Set variable 5 to the breakpoint value for the next cycle
FPA(VAR6)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 5
FOL5Ø	Speed ratio is changed to 5:1
VAR6=VAR6+VAR1	Set variable 6 to the breakpoint value for the next cycle
FPA(VAR7)	Pauses execution until absolute value of the primary encoder counter exceeds breakpoint 6
FOLØ	Speed ratio is changed to 0
VAR7=VAR7+VAR1	Set variable 7 to the breakpoint value for the next cycle
FPA(VAR8)	Pauses execution the absolute value of the primary encoder counter exceeds breakpoint 7
STOP	Ends the move (this is required)
NG	Ends Motion Profiling mode
VAR8=VAR8+VAR1	Set variable 8 to the breakpoint value for the next cycle
H	change the direction
N	Ends the loop cycle
XT	Ends sequence #1 definition

This sequence is an example of a complex following profile. Position and velocity are synchronized and the positional relationship is maintained.

Synchronization

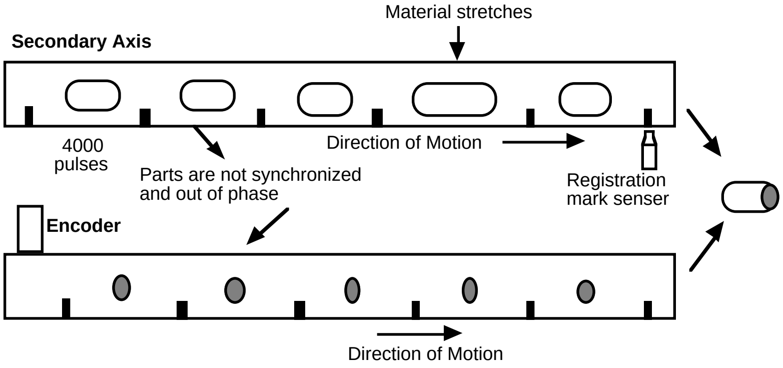
The SXF can synchronize its speed and phase with respect to a primary axis. In many applications, it is necessary to have the position and speed of a secondary axis synchronized with the speed and position of the primary axis with registration marks on the secondary axis parts or material. These marks must be evenly spaced so that at a constant speed (with respect to the primary axis) the number of primary axis encoder steps recorded between registration marks is an expected constant number. If these marks should come further apart (e.g., the material stretches) the SXF will adjust the speed ratio to correct for the error between the registration marks. The following figures illustrate this process.

In the figure below, a secondary axis has parts that are to be synchronized to the primary axis' parts. The registration sensor detects the location of the parts with respect to the primary axis. This sensor goes to the SXF. It indicates the start of the part. The SXF then counts the encoder pulses from the primary axis that occur between registration marks.



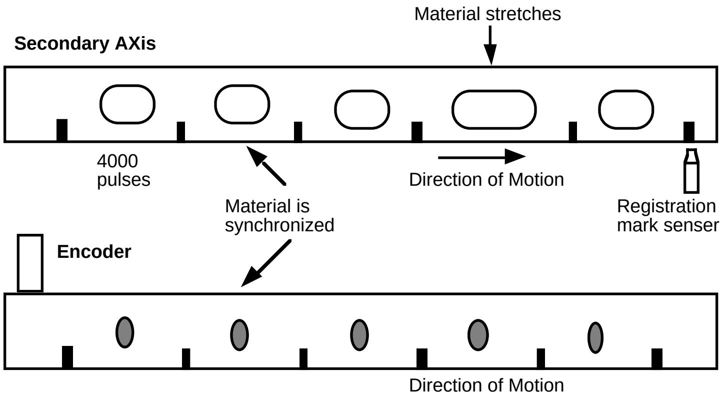
Secondary Parts Synchronized With the Primary Axis

If the material on the secondary axis stretches, as indicated in the following figure, all parts after the stretched material will no longer be synchronized with the primary axis' parts. The next figure depicts the result of not using Synchronization mode.



Material Stretches—Parts Out of Sync

With the SXF's Synchronization mode, the secondary axis accelerates to re-synchronize with the primary axis. It removes the phase shift between the two axes. The SXF detects that the number of pulses between the registration marks has increased (due to stretching). The speed ratio is increased, so the secondary axis speeds up. The material after the stretched portion is good material, so the pulses between the next two registration marks will be slightly less than 4000 because the speed ratio is now higher. The speed will now be reduced. The secondary and primary axes will be synchronized again. Every time a registration mark is encountered, a new actual count is latched and the speed ratio is adjusted to synchronize the axis. Corrections will continue until the secondary axis again has the expected number of pulses between registration marks.



SXF Synchronizes Parts (After Stretching)

You can determine and program the amount of correction that is applied to the speed ratio between each registration mark to fit your application. Use the following commands to program a synchronization application.

Command	Description
> FC	Expected encoder count between registration marks
> FBS	Normalizing count for determining new speed ratio to move at
> FIN	Increment used to determine the new following speed
> FSL	Enables the Synchronization mode
> FSK	Enables the expected encoder count teach mode
> INnI	Defines an input as the registration mark synchronization input.
> TF	Sets the sample rate of the encoder input
> FOR	Motor to encoder count ratio
> FOL	Primary to secondary axis speed ratio
> FSI	Enables Following mode

The **FOL** and **FOR** commands determine the number of secondary motor steps that will be commanded for the encoder pulses that are received. The **FBS** and **FIN** commands determine the amount of correction that will be applied to the **FOL** and **FOR** motor to encoder ratio. **FC** is the number of encoder pulses that are expected between the registration marks if the speeds are synchronized properly. This number is compared to the actual number of pulses that are counted between each registration mark. The difference between these two values represents the error. The **FIN** and **FBS** numbers are applied to this error to determine the new speed ratio.

The **INnI** command configures an input to the SXF for accepting the signal from the sensor that detects the registration marks on the parts or material. The **TF** command is the rate at which the encoder interface is sampled. The **FSL** command enables the Synchronization mode. When Synchronization mode is enabled the first time, the Synchronization input is toggled, the encoder counter is started. The next time it is toggled, the count is latched and the counter is reset to zero. The latched count is the actual number of encoder pulses counted between registration inputs. This is compared to the expected value (**FC**). The difference is multiplied by the correction factor (**FIN** and **FBS**) to determine the speed ratio to run at until the synchronization input is toggled again. This process is continuous as long as the synchronization mode (**FSL**) is enabled.

If you do not know the expected encoder count between registration marks, use the Self Learn mode (**FSK**) to determine the expected count (**FC**). To determine the expected count (**FC**) using the Self Learn mode (**FSK**) you must start the process at the speed ratio that you want to run at, turn on the Self Learn mode. The SXF will count the pulses between the registration marks. When the secondary axis stops, the last recorded number will be placed in the expected count (**FC**) number. This will be used when you are in Synchronization mode.

As an example, the process in the previous figures use an encoder that has 4000 pulses per revolution and the secondary motor has 25000 steps per revolution. The **FOR** command is set to 6.25. The motor is mounted on the secondary conveyer belt so that one revolution is 4 inches. The encoder is mounted on the primary axis so that one encoder revolution is 4 inches. The **FOL** command must be set to 100% (**FOL1ØØ**) for the secondary axis to move at the same speed as the primary axis. If the primary axis moves at 1 rps, the secondary axis must move at 1 rps. In one Figure, the two axes start moving at the same speed. The registration marks are 4 inches apart on the secondary axis. Use the Self Learn mode to determine how many encoder pulses are between the registration marks on the material on the secondary axis. The following steps show how to program the application.

Step ① Set up Self-Learn mode.

Command	Description
> **FOR6.25**	Set the motor to encoder ratio to 6.25
> **FOL1ØØ**	Set the motor to encoder ratio speed percentage to 100%
> **FSI1**	Enable the Following mode
> **FSK1**	Enable the Self Learn mode
> **TF1**	Set the encoder sample rate to 1 ms
> **A5ØØ**	Set the acceleration of 500 rps^2
> **AD5ØØ**	Set the deceleration of 500 rps^2
> **MC**	Place the SXF in the continuous mode
> **IN1I**	Defines input #1 as a synchronization input

Step ② The registration sensor should be wired to the synchronization input. The primary and secondary axes should now be started. After the SXF has passed more than 3 registration marks, it can be stopped. The number of encoder pulses between registration marks can be checked with the **FC** command. In this example, the number that is determined is 4000 counts.

Command	Response
> **1FC**	*4000

If you know the number of encoder pulses you expect to record between registration marks, this number can be entered directly for the **FC** command and will override the number determined in Self Learn mode.

Command	Description
> **1FC4ØØØ**	Manually entering the expected count

Step ③

The SXF now has the number of counts expected between registration marks. The next step is to determine the correction gain desired. The correction will be applied to the difference between the expected count that was just determined and the actual counts that will be counted during actual operation. The equation for the determination of the correction is;

Equation 5-15.
Correction

$$\text{Correction} = \frac{(\text{Actual encoder count - Expected encoder count}) * \textbf{FIN}}{\textbf{FBS}}$$

This motor-to-encoder step ratio is determined by the following equation.

Equation 5-16.
Speed Ratio

$$\text{New Speed Ratio} = \textbf{FOR} * \textbf{FOL} + \text{Correction}$$

To determine the number of motor steps that will be commanded for the number of encoder pulses received, use the following equation.

Equation 5-17.
`Motor Step Correction`

Motor Steps = Encoder Steps * (**FOR*FOL** + Correction)

You must determine the amount of correction you want to have for a given amount of error. Once this has been determined, you can enter the **FIN** and **FOR** commands.

Command	Description
> **1FIN3.12**	The following increment is 3.12
> **1FBS1ØØ**	The following base number is 100

Step ④ Disable Self Learn mode and enable Synchronization mode.

Command	Description
> **1FSKØ**	Disable Self Learn mode
> **1FSL1**	Enable Synchronization mode

Step ⑤ Orient the primary and secondary axes to attain the desired phase relationship. Start both axes at the same time, or start the SXF first and the primary axis second.

The SXF will now correct any errors in the phase relationship between the two axes and maintain a synchronized speed. The new speed ratio that is determined will be applied for the entire period between registration marks. The time between the registration marks is effectively the sample period. A correction is made for each sample period.

Another method for synchronization is to use the inputs to the SXF for increasing and decreasing the following speed ratio. The SXF inputs can be defined to increase or decrease the following ratio. By setting one input for increasing the following ratio and one input for decreasing the following ratio, synchronization can be achieved. In this case, use an external circuit to determine whether the secondary axis should accelerate (increase ratio) or decelerate (decrease ratio) the secondary axis.

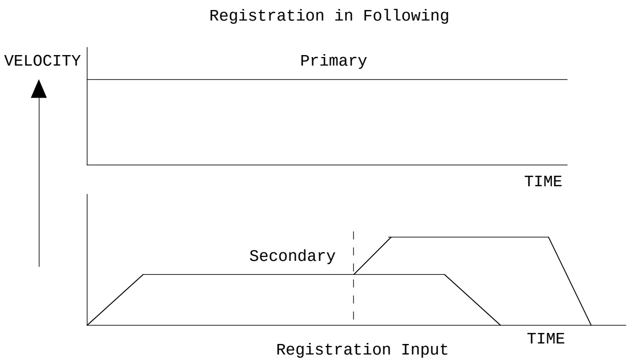
Define the input with the **IN** command. **INnX** defines the input for increasing following ratio, **INnY** defines the input for decreasing the following ratio. The following ratio will be increased or decreased while the input is active. During the SXF's sample periods, the ratio will increase or decrease while the inputs are active. The inputs have a 2 ms debounce time. If the input remains active for 4 ms, the following ratio will be increased or decreased twice. The amount that the following ratio is increased or decreased is determined by the **FIN** command. If **FIN** is 1, **FOL** is increased or decreased by 1 during each sample period.

Other Following Features

This section discusses following features that the SXF provides for special following requirements.

Registration in Following Mode

With the SXF, registration can be performed in Following mode. It is like programming registration in the Indexer version, *but the velocity term is replaced by* **FOL** *for the desired speed*. This figure illustrates registration in Following mode.



Registration in the Following Mode

Registration can only be performed if the Following Synchronized Acceleration mode is disabled (**FSFØ**). For this example, re-attach the encoder to the primary axis.

Step ① Repeat steps 1 - 5 of the *Velocity Following Example*.

Step ② Use the dedicated registration input to start a registration move

Step ③ Define registration move #1 as follows:

```
> REG1,A1Ø,AD1Ø,FOL2ØØ,D25ØØØ
```

Step ④ Begin motion on the primary axis, then begin motion with the SXF.

Command	Description
> MC	Change to continuous mode
> G	Initiate motion

Step ⑤ Toggle the registration input. The motor will begin following at a speed ratio of 2:1 for 25,000 motor steps.

Jogging in Following Mode

In some applications, you may want to move the motor manually while in Following mode. This allows you to follow the primary axis at a following ratio by toggling a switch. You can configure the SXF to allow you to follow the primary motor manually with the Configure Input (**IN**) command. Jogging in Following mode does not require the **JHV** or **JVL** commands. In this case, you will jog at whatever speed ratio the **FOL** command is set to. To use the inputs, you can either configure the input as a CW or a CCW jog as with the preset Indexer jog. However, you cannot use the high-speed/low-speed jog input because you can only jog at the speed set by the **FOL** speed ratio.

Therefore, you use the two jog input functions: CW Jog input (**IN#J**) and CCW Jog Input (**IN#K**). You must also enable the jogging feature with the **OSE1** command. Once you set these parameters, you can attach a switch to the jog inputs (predefined) and jog the motor(s). The **#** character represents digits 1 - 8, which you enter. You must have the SXF in Following mode to jog at a speed ratio of the primary. The following example shows you how to define power-up sequence #100 to jog.

Step ① Define a power-up sequence. ***Position Tracking mode must be disabled* (FSPØ) *to enable direction jogging.***

Command	Description
> XE1ØØ	Erases sequence #100
> XD1ØØ	Defines sequence #100
> LD3	Disables the limits(*not needed if you have limit switches installed*)
> JA25	Sets Jog Acceleration to 25 rps^2
> JAD25	Sets Jog Deceleration to 25 rps^2
> OSE1	Enables Jog function
> IN1J	Sets **IN 1** as a CW jog input
> IN2K	Sets **IN 2** as a CCW jog input
> FOR6.25	Sets the motor to encoder steps
> FOL75	Secondary moves at 75% of the primary speed
> FSI1	Enables Following mode
> XT	Ends sequence definition

Step ②

Command	Description
> Z	Resets the SXF

Reset the SXF. Move the primary or primary axis.

Step ③ Turn on **IN 1** to move the motor CW at 75% of the primary axis' speed (until you turn **IN 1** off).

Step ④ Turn on **IN 2** to move the motor CCW at 75% of the primary axis' speed (until you turn **IN 2** off).

Following a Step and/or Direction Signal

The SXF can follow a step (or pulse) and a direction signal rather than quadrature encoder pulses. The same incremental encoder interfaces used for quadrature following are used for the step and direction following. The **Phase A+** and **Phase A-** inputs are now the **Step+** and **Step-** inputs. The **Phase B+** and **Phase B-** inputs are now the **Direction+** and **Direction-** inputs. The only other requirement is to put the SXF in the pulse and direction mode. This is accomplished with the **FSN** command. By typing **FSN1** the pulse and direction capability is enabled and the **Phase A** and **Phase B** inputs are now step and direction inputs. Once the pulse and direction capability is added, any following applications are performed exactly as if the input were quadrature signals. If your application requires pulse and direction, enable the Pulse and Direction mode and repeat the procedures in this following section.

Following Equation and Command Summary

This section provides a reference for the following equations and the SXF software commands that are associated with following.

Following Command Summary

For a complete explanation of these following commands, refer to the ***SX Software Reference Guide***. *Set-up commands are required for any following application.*

FSA Followed by a 1—enables instant acceleration between commanded velocities for each resulting velocity change after sampling the encoder and determining the ratio.

FSF Followed by a 1—enables the use of following acceleration as determined by **FAC** and **FEN**.

FSI Followed by a 1—enables following mode versus Indexer mode

FSK Followed by a 1—enables the calculation of **FC** for Synchronization mode.

FSL Followed by a 1—enables Synchronization mode

FSN Followed by a 1—enables a step & direction signal to be followed or just a pulse train if direction is not used in conjunction with **FSP**.

FSP Followed by a 1—enables position tracking.

FOR The number of secondary motor pulses per unit of travel divided by the primary encoder pulses per the same unit of travel.

FOL The % of the primary encoder speed that the secondary axis moves at.

FAC The change in following % for each change in encoder pulse count of **FEN** during following acceleration.

FEN The number of encoder pulses that the encoder count must change by to increment the following percentage by **FAC.**

FP In Motion Profiling (MPP) mode, command execution is paused for the number of following encoder steps entered in the **FP** command.

FPA In Motion Profiling (MPP) mode, the execution of commands pauses until the value in the encoder counter exceeds the **FPA** value.

FIN The amount by which the following % changes when changed by inputs.

FBS In synchronization, used with **FIN** to determine the amount of following percentage correction.

FC In synchronization, the expected number of encoder counts between registration marks.

TF The following encoder sample period

VARn=FEP **FEP** is a read only variable of the actual encoder count. Set it equal to the variable to get the current value of the encoder counter.

PF Gives you a report back of the encoder count.

PFZ Clears the encoder counter.

INnX An input function that lets you increase the following percentage by **FIN**.

INnY An input function that lets you to decrease the following percentage by **FIN**.

INnI An input function used for the registration mark sensor of the synchronization mode.

The commands are categorized according to the applications the support.

Velocity Following	***Velocity and Position Following***	
FOR	FOR	FSF
FOL	FOL	FEN
FSI	FSI	FAC
TF	TF	FSA
		FSP

Advance and Recede		***Synchronization***		***Special Function***	
FOR	FSF	FOR	FSF	FBS	FSN
FOL	FAC	FOL	FAC	FSK	FSM
FSI	FEN	FSI	FEN	FSL	PF
TF	FP	TF	FC	NnI	PFZ
	FPA	INnY	FIN	INnX	

Following Equation Summary

The following equations were discussed throughout this chapter. They are provided again here for reference and convenience

Velocity Following

$$\text{Secondary Motor steps} = \text{Primary encoder count} * \mathbf{FOR} * \frac{\mathbf{FOL}}{100}$$

Velocity and Position Following

Given V_{max} and D_{acc}:

$$\mathbf{FEN} = V_{max} * \frac{\mathbf{TF}}{1000} \qquad V_{max} = \frac{\text{Primary encoder counts}}{\text{second}}$$

TF = Primary Encoder Sample Period in ms

$$\mathbf{FAC} = \frac{\mathbf{FOL} * V_{max}}{D_{acc}} * \frac{\mathbf{TF}}{1000} \quad V_{max} = \frac{\text{Primary encoder counts}}{\text{second}}$$

TF = Primary Encoder Sample Period in units of ms

FOL = Following percentage in units of percent

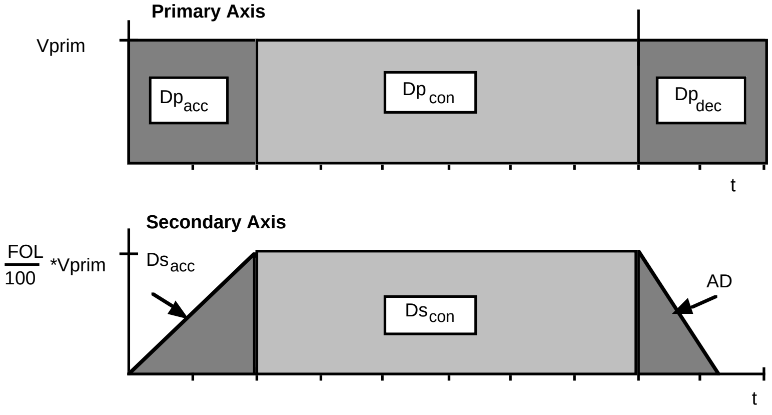
$\mathbf{D_{acc}}$ = Distance primary axis moves during secondary axis acceleration in units of primary encoder counts

Given V_{max} and A_{max} Determine D_{acc} from $D_{acc} = \dfrac{V_{max}{}^2}{A_{max}} * \mathbf{FOR} * \dfrac{\mathbf{FOL}}{100}$

The acceleration is in units of secondary motor steps/sec2. The maximum velocity of the primary is in primary motor steps/sec.

The following figure illustrates the motion profiles of a secondary and the primary. The different parameters are shown on the profiles and the equations to determine the parameters are given below. **AD** is used for the deceleration.

Velocity and Position Following With AD Decel



$$Ds_{acc} = \mathrm{FOR} * \left(\frac{1}{2} * \Delta\mathbf{FOL}^2 * \frac{\mathbf{FEN}}{100*\mathbf{FAC}} + \Delta\mathbf{FOL}*\mathbf{FOL}_I * \frac{\mathbf{FEN}}{100*\mathbf{FAC}}\right)$$

FOL= The change in following percentage

$\mathbf{FOL}_I$ = The initial following percentage

The equations above can be simplified by defining a following acceleration constant determined by **FAC** and **FEN**.

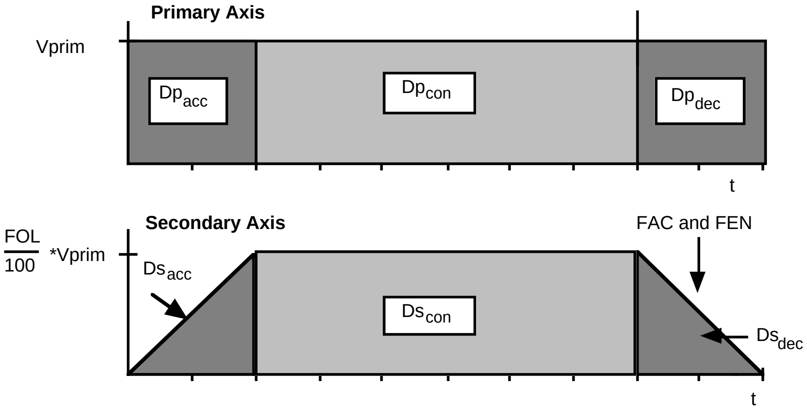
$$K = \frac{\mathbf{FEN}}{100*\mathbf{FAC}}$$

The equations can now be written as:

$$D_{sec} = \mathbf{FOR} * K * \left(\frac{1}{2} * \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL}*FOL_I\right)$$

$$D_{sec} \text{ (Deceleration)} = -\mathbf{FOR} * K * \left(\frac{1}{2} * \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL}*FOL_I\right)$$

Use the equations and parameters in the next figure To make a trapezoidal move (deceleration is done according to the **FAC** and **FEN** commands).



Velocity and Position Following With Following Decel

$$D_{prim} = Dp_{acc} + Dp_{con} + Dp_{dec}$$

$$D_{sec} = Ds_{acc} + Ds_{con} + Ds_{dec}$$

$$Ds_{acc} = \mathbf{FOR} * K * \left(\frac{1}{2} * \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL}*FOL_I\right)$$

$$Ds_{dec} = -\mathbf{FOR} * K * \left(\frac{1}{2} * \Delta\mathbf{FOL}^2 + \Delta\mathbf{FOL}*\mathbf{FOL}_I\right)$$

$$Dp_{acc} = \mathbf{FOL} * \frac{\mathbf{FEN}}{\mathbf{FAC}}$$

$$Ds_{con} = Dp_{con} * \mathbf{FOR} * \frac{\mathbf{FOL}}{100}$$

$$\mathbf{FOL} = \frac{D_{prim}}{200*K} - \sqrt{\left(\frac{D_{prim}}{200*K}\right)^2 - \left(\frac{D_{sec}}{K}\right)}$$

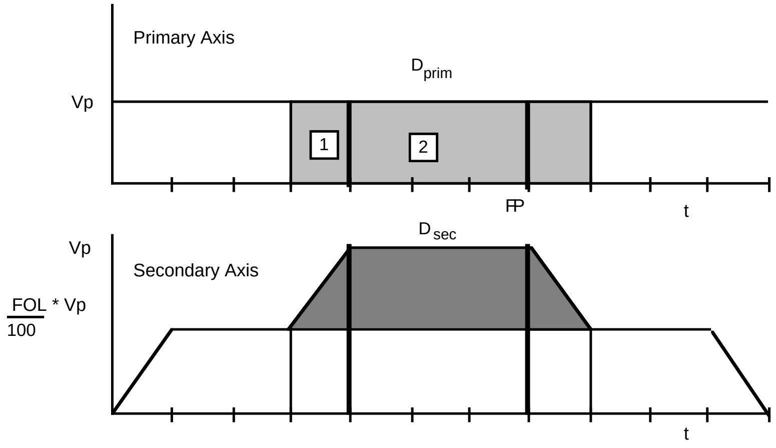
K is the following acceleration constant, D_{prim} and D_{sec} are the distances that the primary axis and secondary axes will move, respectively. $\mathbf{FOL}_I$ is the initial following percentage. In the case of a trapezoidal move, it will always be 0. The required sequence is:

Command	Description
`> XE1`	Erases sequence #1
`> XD1`	Defines sequence #1
`FSI1`	Enables Following mode
`FORn`	Sets motor to encoder steps per unit travel ratio
`FACn`	Sets the following acceleration increment
`FENn`	Sets the encoder period
`Dn`	Sets the secondary axis distance to n motor steps
`FOLn`	Sets the initial following percentage to n
`TR1`	Waits on the input trigger
`MPP`	Enters the Motion Profiling mode
`G`	Starts motion
`FPa`	Waits until a encoder pulses have passed
`FOLØ`	Stops the motion of the secondary
`FPb`	Waits for the decel ramp distance
`STOP`	Stops the move itself
`> XT`	Ends the sequence #1 definition

The value for a in the first **FP** = $Dp_{acc} + Dp_{con}$

The value for b in the second **FP** = Dp_{dec}

Advance and Recede



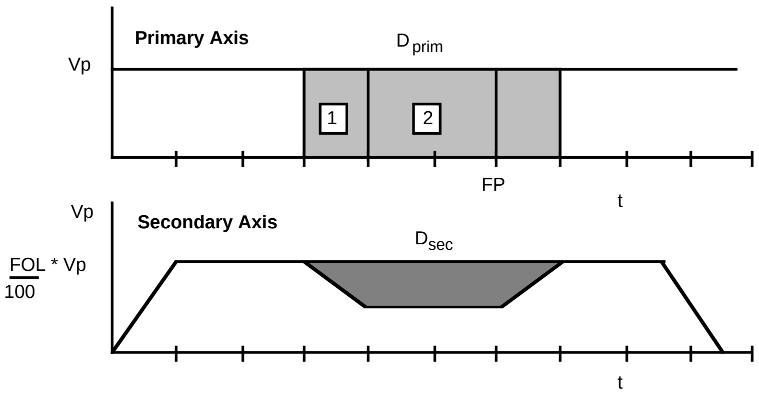
Advance Profile Mode

In the figure shown below, D_{prim} is the distance that the primary axis moves in encoder steps during the advance portion of the profile (lightly shaded region). D_{sec} is the distance in secondary motor steps that the secondary axis advances with respect to the moving primary axis (darkly shaded region).

$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{200*K} - \sqrt{\left(\frac{D_{prim}}{200*K}\right)^2 - \left(\frac{D_{sec}}{\mathbf{FOR}*K}\right)}$$

The .i.breakpoint; to decelerate to FOL1ØØ is entered for the FP value in the advance sequence. FP is a distance equal to the sum of areas 1 and 2 in the primary profile.

$$\mathbf{FP} = D_{prim} - \frac{\mathbf{FEN}}{\mathbf{FAC}} * \left(\mathbf{FOL}_F - \mathbf{FOL}_I\right)$$



Recede Profile

The dark shaded area is the distance that the secondary will recede with respect to the moving primary. The lightly shaded are is the distance the primary will move while the secondary recedes.

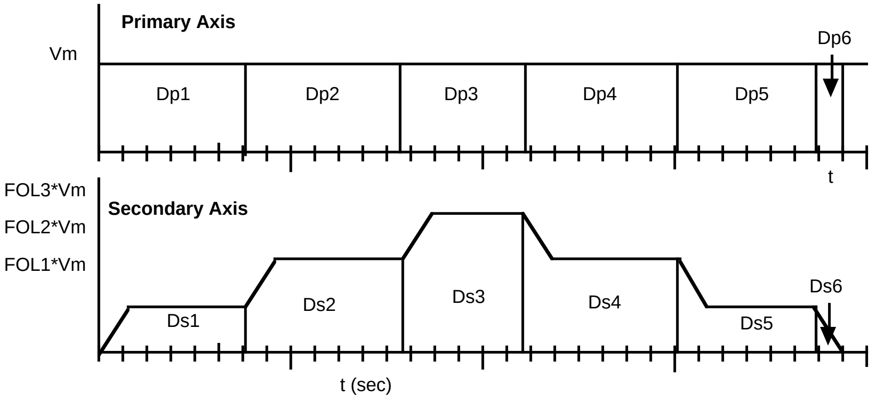
$$\mathtt{FOL} = \mathbf{FOL_I} - \frac{D_{prim}}{200*K} + \sqrt{\left(\frac{D_{prim}}{200*K}\right)^2 - \frac{D_{sec}}{\mathtt{FOR}*K}}$$

K is the following acceleration constant and is equal to: $\frac{\mathbf{FEN}}{100*\mathbf{FAC}}$

FP is equal to: $D_{prim} + \frac{\mathbf{FEN}}{\mathbf{FAC}} * \left(\mathbf{FOL}_F - \mathbf{FOL}_I\right)$

Cam Following

The following figure is a cam profile.



Cam Profile

Each segment of the secondary and primary move profile is marked by a primary distance Dpn and a secondary distance Dsn. The following percentage required for each segment is determined from the equations below. D_{prim} and D_{sec} are equal to Dpn and Dsn for each segment. When accelerating to a higher percentage use.

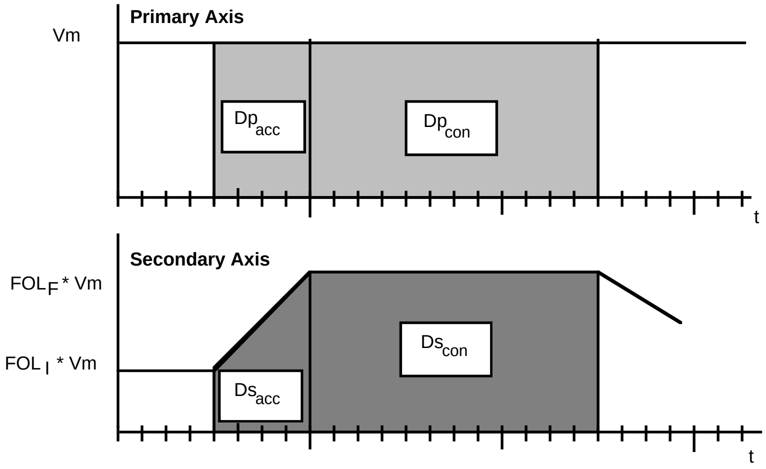
$$\mathbf{FOL} = \mathbf{FOL}_I + \frac{D_{prim}}{100*K} - \sqrt{\left(\frac{D_{prim}}{100*K}\right)^2 + \frac{2*FOL_I*D_{prim}}{100*K} - \frac{2*D_{sec}}{\mathbf{FOR}*K}}$$

When decelerating to a lower percentage, use:

$$\mathbf{FOL} = \mathbf{FOL}_I - \frac{D_{prim}}{100*K} + \sqrt{\left(\frac{D_{prim}}{100*K}\right)^2 - \frac{2*\mathbf{FOL}_I*D_{prim}}{100*K} + \frac{2*D_{sec}}{\mathbf{FOL}*K}}$$

FP or FPA are equal to Dpn or the accumulative Dpn respectively.

Each segment of the cam can be broken down as shown in below and described the equations below.



Cam Profile Segment

$$D_{prim} = Dp_{acc} + Dp_{con}$$

$$D_{sec} = Ds_{acc} + Ds_{con}$$

$$Ds_{acc} = FOR * K * \left(\frac{1}{2} * (FOL_F - FOL_I)^2 + (FOL_F - FOL_I) * FOL_I\right)$$

$$Dp_{acc} = FOL_F - FOL_I * \frac{FEN}{FAC}$$

$$Ds_{con} = Dp_{con} * FOR * \frac{FOL}{100}$$

The distance for FP is $Dp_{acc} + Dp_{con}$, for FPA, is the absolute value of the primary distance since the start of the cam cycle.