430-FOL Operator's Manual
8/06/87
P/N 88-007201-01

NOTE

This manual is intended to be used with the Model 430 Indexer Operator's Manual.

Table Of Contents

Chapter 1. Encoder Following

1.1. Introduction

The Compumotor 430-FOL indexer allows a user to proportionately track the velocity and position of a "master" shaft, or motor, with a "slave" motor. The requirement of the master is that it have a TTL output quadrature encoder attached to it's shaft to feed position and velocity information back to the 430-FOL. This information allows the 430 to output a step rate to the slave at a user selectable ratio to the master shaft, or motor. The correction factor (rate at which tracking errors are corrected), and encoder sample rate (rate at which the 430-FOL samples the position and velocity of the master shafts encoder), are both user programmable as well.

1.2. Features

The 430-FOL allows the user to track the master shaft at a continuous constant ratio, at continuous changing ratios, at a constant ratio of a predefined distance, or, at changing ratios over predefined distances. Many different ratios may be used in a single ratio profile. The slave may be programmed to follow in the same, or in the oppsosite direction of the master shaft. These capabilities make the 430-FOL appropriate for even the most complex follower applications.

1.2.1. Sample Period

The 430 Encoder Follower system begins with a TTL quadrature encoder mounted on a motor shaft or system component. This shaft is refered to as the "master". The encoder provides a step rate to the 430-FOL proportional to the rotational speed of the "master" shaft. The 430-FOL periodically checks the position of the "master" via the encoder feedback. The rate (refered to as sample period) at which the 430-FOL checks the "master's" position is programmable by the user. The 430-FOL interpolates the master's velocity and position for the last sample period at the end of each sample period. With this data, the 430-FOL can determine the amount of error between the slaves desired position, and it's actual position. All or part of this error will be corrected in the next Sample Period (the percentage of error corrected depends upon the user selected value of the Sample Correction Attenuation Factor. (See Below)). Sample Period is set with the 0097 command.

Sample Correction Attenuation Factor (SCAF)

The value for SCAF is inserted into the formula:

$$\text{Corrected Error} = \text{Total Error} / 2^{(SCAF)}$$

The valid range for SCAF is 0 - 7. The lower this number the faster the correction rate, i.e. The rate of correction is inversely proportional to the setting of SCAF. This value will effect the gain in a logarithmic, rather than a linear fashion.

1.3. Constructing a Predefined Follower Move Segment

Encoder follower move segments must be predefined with two seperate commands. These commmands are the "define move segment distance" (cmd 286), and "define move segment" (cmd 201). The 286 command allows the user to define the number of motor steps to be output (this value is placed in R1,R2), over the number of encoder steps to be recieved (this value is placed in R3,R4). These commands are intended to define a move segment, not to execute one. Execution of predefined segments is covered in section XX.

For example:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	1000	0000	1000	0286

This command will define a move segment that will output 1 Motor step for every Encoder step received (1:1 Ratio), up to 1000, at which point the segment will be complete. This command is equivalent to both the "Set Acceleration" (cmd 0262), and "Set Velocity" (cmd 0261), in a normal move definition, in that it must be executed before a predefined move segment command is executed (cmd 0201). Since you are defining both the number of encoder steps and the number of motor steps for the segment, the average ratio of motor steps to encoder steps during the segment is also inplied (in this case the implied Average Ratio (AR) = 1 or 1000 Encoder steps/1000 motor steps). There are three ratios involved in any follower move segment, the average ratio, the beginning ratio, the final ratio. The average ratio is just that; the average ratio of the entire move segment. This ratio will be the same as the beginning and final ratios on constant ratio follower segments. Segments that cause the motor to either ramp to a new ratio will have different beginning and final ratios. The average between the beginning and final ratio will be the avarage ratio. The mathmatical formula for this is as follows:

$$\text{Avg Ratio} = \text{Final Ratio} + \text{Beginning Ratio} \quad / \quad 2$$

The beginning ratio of any follower segment is equal to the final ratio of the segment preceding it. The final ratio of follower move segment is defined with the "define move segment" command (cmd 0201). The integer portion of this ratio is defined

in Register #3, the fractional portion of this ratio is defined in Register #4.

For example:

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0001	0000	0002	0000	0201

The number in Reg #1 is the move move number of the follower move segment being defined (in this case, move #1 is being defined). Reg #2 determines the direction of the "slave" motor, a zero, as in this example, will cause the move segment to be in the same direction as the encoder. If Reg #2's value is 0001, the move segment will be in the opposite direction from the encoder. Reg #3 defines the integer portion of the final ratio for the move segment (in this case, Final Ratio = 2). Reg #4 defines the fractional portion of the final ratio (in this case, Final Ratio = 2.0). This command can only be executed after a 286 command has been executed, otherwise, the distance for the predefined segment will be undefined and, therefore, will not be executed by the 430. If several move segments are to take place over the same ratio of motor steps to encoder steps, only one 0286 command is necessary before defining all of them. A typical follower move segment definition would look like this:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	1000	0000	1000	0286
0001	0000	0002	0000	0201

The first segment of any follower move profile has a Beginning Ratio equal to zero. When the beginning ratio and Average Ratio are both known, the Final Ratio can be determined in the following manner:

Average Ratio = AR Final Ratio = FR Beginning Ratio = BR

AR = BR + FR / 2

Algebraic Transposition of this equation results in:

1) **2AR = BR + FR** (Multiply both sides by 2)

2) **2AR - BR = FR** (Subtract BR from both sides)

For the above example the equation would look like this:

1 = 0 + FR / 2

1) 2 * 1 = 0 + FR

2) 2 * 1 - 0 = FR

3) FR = 2

Even though the final ratio can be calculated with the data given with the 0286 command, it is necessary to define the FR with the 0201 command to compensate for any round off errors that may exist at the end of the segment. It is important to remember that the average ratio is only implied in 0286 command. The true average ratio is derived from the beginning ratio (final ratio of preceding segment), and the final ratio specified with the 0201 command. These values inserted into the formula (AR = (FR + BR)/2) will yield the true average ratio. In the above example, a value of 0 rather than 2 in Reg #3 of the 0201 command would result in the motor ramping up to a ratio of 2 and instantaneously going to a ratio of 0 at the end of the segment. The profile would look like this:

At the end of any follower move segment the 430 compares it's actual Final Ratio with the desired Final Ratio, specified with the 0201 command, and instantly corrects any error. This will cause a "hiccup" at the end of a segment whose actual average ratio is not equal to the implied average ratio specified with the 0286 command.

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	1000	0000	1000	0286
0001	0000	0002	0000	0201

The profile generated by this segment will look like this:

Any follower segment that does not have a final ratio of zero, and has no segment following it, will continue to run at the final ratio of the last defined segment, until it is commanded to go to a new ratio, or until it is commanded to stop.

For preset follower moves at least two segments must be defined, with the last segment having a final ratio of zero. This will result in a triangular profile. A trapezoidal profile requires at least three segments, like the triangular profile, the last segment must have a final ratio of zero, and the second segment in a trapezoidal profile will usually have the same beginning and final ratio. Preset follower segments are not limited to three segments, any number and combination of segments may be used to construct more complex profiles.

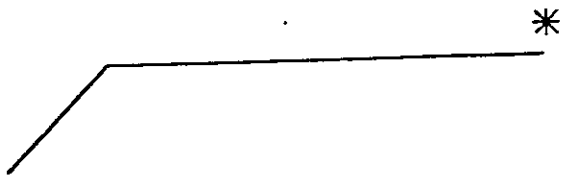
The command lines for defining the second segment of a trapezoidal profile would look like this:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	6000	0000	3000	0286
0002	0000	0002	0000	0201

In this segment an average ratio of 2 is specified with the 286 command. The final ratio specified by the 201 command is also 2. This dictates that the beginning ratio must also be 2 because, (AR = (FR + BR)/ 2). If this segment is added to the example above, the command string for the profile is as follows:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	1000	0000	1000	0286
0001	0000	0002	0000	0201
0000	6000	0000	3000	0286
0002	0000	0002	0000	0201

The profile would look like this:



* Marks the spot where the final ratio is attained. If no new ratio (FOL Move segment) is specified, the slave motor will continue to run at the final ratio of the last processed segment until:

a. It is commanded to stop
b. A limit is encountered
c. A new segment (201) command is executed.

Adding a third segment with a final ratio of zero will complete a trapezoidal FOL profile. The command lines for the final segment are as follows:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	1000	0000	1000	0286
0003	0000	0000	0000	0201

An Average Ratio of 1 is implied with the 286 command. The Final Ratio of 0 is specified with the 201 command. Ramping to a Ratio of 0 is equivalent to issuing a stop command to the "slave" motor. All three move segments defined together would look like this:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0000	1000	0000	1000	0286
0001	0000	0002	0000	0201
0003	0000	0000	0000	0201
0000	6000	0000	3000	0286
0002	0000	0002	0000	0201

Move segments #1 and #3 Have identical Average Ratios, only one 286 command need be issued to define both segments.

1.4. Executing Predefined Segments

Execution of a Predefined FOL segment is accomplished by issuing the 0104 command (Execute Predefined FOL segment). The Sequence definition for executing the predefined segments in the above example is as follows:

Reg #1	Reg #2	Reg #3	Reg #4	Reg Cmd
0001	0000	0000	0000	0217
0011	0000	0010	0000	0097
0000	0001	0000	0000	0096
0001	0000	0000	0000	0104
0002	0000	0000	0000	0104
0003	0000	0000	0000	0104
0000	0000	0000	0000	0096
0000	0000	0000	0000	0216

Line 1) Begin the definition of sequence #1.
Line 2) Set FOL parameters (sample rate = 10 ms, dir = dir of master shaft, Sample Correction factor = 1, Sample Averaging = off). See command description.
Line 3) Enter FOL mode (see command description).
Line 4) Execute segment #1.
Line 5) Execute segment #2.
Line 6) Execute segment #3.
Line 7) Exit FOL mode.
Line 8) End definition of sequence #15.

The resulting profile looks like this:

1.5. Continuous Follower Moves

Segments of indeterminate length are implemented much the same way as preset discrete FOL segments. The main difference being the lack of necessity to define the number of motor pulses in which any given ratio change is to take place in a segment that is not of a predetermined length.

The commands that will be discussed in this section are the 0106 command (Change Ratio), and the 0098 command (Specify

Interval). Both are neccessary to execute a segment of indeterminate length.

The 0098 command determines the number of encoder steps in which the ratio change(s) specified with the 0106 command will take place. Once a number of encoder steps is determined with the 0098 command, each 0106 command entered after it will use the value of encoder steps specified to make the ratio change.

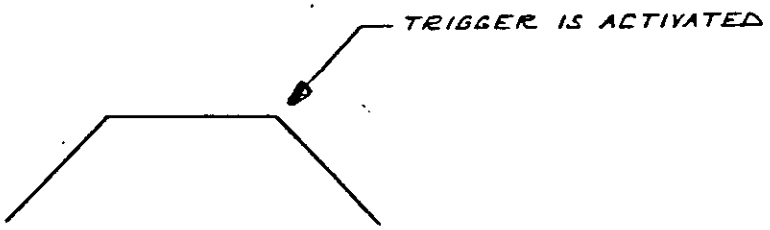
The 0106 command specifies the ratio to which the "slave" motor will ramp to when the 0106 command is executed. The motor will change ratios over the number of encoder steps last specified with the 0098 command. Once the ratio specified with the 0106 command is attained, the "slave" motor will maintain that ratio until a new 0106 command is executed (usually after a Trigger (0044), or Time Delay (0041, 0042, 0043) command).

The command lines below define a simple trapezoidal ratio profile. For the sake of the example, it is assumed that the the "master" encoder is maintaning a constant velocity.

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0011	0000	0001	0000	0097
0000	0001	0000	0000	0096
0000	0000	0000	4000	0098
0000	0000	0002	0000	0106
0002	0000	0000	0000	0044
0000	0000	0000	0000	0106

Lines 1 - 2) Set up FOL parameters and enter FOL mode.
Line 3) Define ratio change interval of 4000 encoder steps.
Line 4) Ramp to a ratio of 2:1.
Line 5) Wait for Trigger 2 to go low.
Line 6) Ramp to ratio of 0.

The resulting profile would look like this.



1.6. Combining Indeterminate and Predetermined Length Profiles

Sometimes it is neccessary to combine both indeterminate and predetermined length profiles within the same sequence of commands. To make this possible the 430 needs to determine both the encoders' and the motors' absolute position. In order for the 430 to do this the 0105 command (Initialize reference counts) must be issued.

Whe the 0105 command is entered the 430 reads the absolute position of the motor and the encoder. This information allows the 430 to perform a segment of a predfined number of encoder and motor steps. For example,

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0000	0000	0000	4000	0098
0005	0000	0000	8000	0286
0001	0000	0000	0000	0201
0000	0000	0012	5000	0106
0002	0000	0000	0000	0044
0000	0000	0000	0000	0105
0001	0000	0000	0000	0104

Line 1) Sets ratio change interval to 4000 encoder steps.
Line 2) Sets predefined segment distance to 50,000 motor steps over 8,000 encoder steps. (see line 3)
Line 3) Defines segment #1 with a final ratio of 0.
Line 4) Ramp the motor to a ratio of 12.5:1 over 4000 encoder steps.
Line 5) Wait for Trigger #2 to go low.
Line 6) Initialize reference counts. (The 430 will read the absolute position of the encoder and motor when it executes this command).
Line 7) Ramp to a ratio of 0 over 50,000 motor steps and 8,000 encoder steps.

1.7. Encoder Step Delays

It is possible to implement a time delay over a predetermined number of encoder steps. This can be done by defining a constant ratio segment with the Average. Beginning, and Final ratios all equal to zero. For example,

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0000	0000	0005	0000	0286
0001	0000	0000	0000	0201

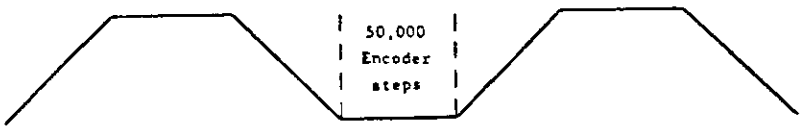
This will define segment #1 as a segment encompassing 50,000 encoder steps, 0 motor steps. When executed with a 0104 command, this segment will cause the "slave" motor to remain motionless during 50,000 encoder steps. A typical use would be in a loop as part of larger repeating ratio profile. For example,

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0000	0000	0005	0000	0286
0001	0000	0000	0000	0201
0002	5000	0005	0000	0286
0002	0000	0001	0000	0201
0004	0000	0000	0000	0201
0005	0000	0005	0000	0286
0003	0000	0001	0000	0201
0011	0000	0010	0000	0097
0000	0001	0000	0000	0096
0005	0000	0000	0000	0032
0001	0000	0000	0000	0044
0001	0000	0000	0000	0104
0002	0000	0000	0000	0104
0003	0000	0000	0000	0104
0004	0000	0000	0000	0104

Line 1) Specify segment distance - Motor stps = 0, Encoder stps = 50,000.
Line 2) Define segment #1 with a Final ratio of 0.
Line 3) Specify segment distance - Motor stps = 25,000, Encoder stps = 50,000.
Line 4) Define segment #2 with a final ratio of 1.
Line 5) Define segment #4 with a final ratio of 0.
Line 6) Specify segment distance - Motor stps = 50,000, Encoder stps = 50,000.
Line 7) Define segment #3 with a final ratio of 1.
Line 8) Define FOL parameters.
Line 9) Enter FOL mode.

Line 10) Loop on the next 5 commands forever.
Line 11) Wait for Trigger #1 to go low.
Line 12) Execute segment #1.
Line 13) Execute segment #2.
Line 14) Execute segment #3.
Line 15) Execute segment #4.

The ratio profile for the above example would look like this:



1.8. Thumbwheel Moves

It is sometimes necessary to select and/or change ratios dynamically from an outside interface. This can be done via BCD inputs DB0-DB16 using the 0193 command (Change to thumbwheel ratio). Whenever the 430-FOL executes a 0193 command it scans the BCD inputs (DB0-DB16) and ramps the motor to the ratio whose value appears on these inputs. The rate at which the 430-FOL ramps to the ratio is determined by the last 0098 command (Specify Interval).

In order for the 430-FOL to correctly interpolate the number it sees on the BCD inputs, it must know the motor to encoder steps ratio. This ratio must be input with the 0192 command before any thumbwheel ratios can be specified. This ratio will be multiplied by the number on the BCD inputs so that the desired specified ratio can be obtained. For example, a 25000 step per revolution motor, being used as the slave, and a 4000 post quadrature pulse per revolution encoder, being used as the master, would yield a thumbwheel scale factor of 6.25.

$$\text{Motor Resolution} = \frac{25000}{\text{Encoder Resolution} = 4000} = 6.25$$

Most commonly a 0193 command would be used after a "Wait on Trigger" command, to give the operator time to change the thumbwheel specified ratio before the 0193 command is executed. For example:

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0000	0000	0006	2500	0192
0002	0000	0000	0000	0033
0002	0000	0000	0000	0044
0000	0000	0000	0000	0193

Line 1) Set Thumbwheel scale factor to 6.25.
Line 2) Loop on the next two commands contiuously.
Line 3) Wait for Trigger #2 to go low.
Line 4) Ramp[1] to ratio specified on Thumbwheels.

In this example, the operator can set the desired ratio on the Thumbwheels and then manual activate the trigger, whereupon the 0193 command will be proccessed and the slave motor will ramp to the specified ratio.

1.9. Using Standard 430 Commands in FOL Mode

All IMMEDIATE 430 command may be executed in FOL mode. DOMINANT commands can only be executed in standard 430 mode, they may NOT be executed in FOL mode. Refer to Model 430 Indexer Operator's Manual, for descriptions of all standard 430 commands.

1.10. Power Up Execution

It is possible for the 430-FOL to operate as a stand alone system. Sequence #15 will be executed automatically on power up if the following conditions are true.

1) Sequence 15 has been defined and saved.
2) Autorecall is enabled (see Model 430 Indexer Operator's Manual).

---

[1] Several necessary 430-FOL commands are not shown in this example. This was done to maintain clarity in the example. The commands that were omitted were, Cmd 0098, Cmd 0097, Cmd 0096.

Chapter 2. Programming Examples

2.1. Programming Example #1

2.1.1. The Problem

A small liquid fertilizer company wants to construct a machine which will allow them to package 10 different products with 5 different concentrations for each product. They create a mix for each product by adding the product concentrate to a water based medium in the desired ratio. As it turns out, most of the 50 possible permutations are different ratios. Also, due to imperfections in the manufacture of the concentrates, the ratios may need to be adjusted slightly during packaging.

2.1.2. The Solution

They construct a machine with a large pump for the water based medium, and a smaller pump for the concentrates. The large pump produces ten times the volume per shaft rotation that the small pump does. The concentrations are entered on the 430 thumbwheels in parts per thousand, with values ranging from 10 to 1000. A value of 1000 should give a 1:1 ratio in volume, which means a 10:1 ratio of small pump speed to large pump speed. With this arrangement, the thumbwuheels simply give the ratio of shaft speeds as XX.XX. Remembering that the 193 command interprets the thumbwheels as XX.XX it remains only to determine the proper scale factor to convert shaft ratio to motor:encoder ratio. A 1000 line encoder on the large pump produces 4000 counts/rev. A 25000 step/rev motor on the small pump leads to a scale factor of 25000/4000 = 6.25. An input on trigger 1 allows pushbutton activation of the new thumbwheel ratio. Sequence 15 is used in conjuction with autorecall so the ratio operation begins at power-up. To prevent sudden pressure changes, the ratios will change over one revolution of the main pump shaft.

2.1.3. The Program

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0015	0000	0000	0000	0217
0000	0000	0000	4000	0098
0000	0000	0006	2500	0192
0011	0000	0015	0000	0097
0000	0001	0000	0000	0096
0002	0000	0000	0000	0033
0002	0000	0000	0000	0044
0000	0000	0000	0000	0193
0000	0000	0000	0000	0216
0000	0001	0000	0000	0293
0000	0000	0000	0000	0291

Line 1) Begin Sequence 15 definition.
Line 2) Ramp to new ratio over 4000 encoder steps.
Line 3) Set Thumbwheel scale factor to 6.25.
Line 4) Set udate rate to 15 milliseconds.
Line 5) Enter FOL mode.
Line 6) Loop on the next two commands contiuously.
Line 7) Wait for Trigger #2 to go low.
Line 8) Ramp to thumbwheel ratio.
Line 9) End Sequence 15 definition.
Line 10) Enable AutoRecall.
Line 11) Save the Program.

2.2. Programming Example #2

2.2.1. The Problem

In a certain manufacturing process, a heavy part rides on a conveyor belt with an unpredictable and only roughly constant velocity. Halfway down the line, the part needs to have two spot welds, 4 inches apart, with the first weld being two inches from the leading edge of the part. The belt never stops, because it is loaded with the heavy parts. The spot weld head needs to detect the edge of the moving part, and ramp up to following speed by the first weld location. The weld takes one second, then the welder needs to recess 4 inches with respect to the moving part and start another weld. One second later the second weld is complete and the welding head returns to wait for the next part to move by.

2.2.2. The Solution

The welding head and a photo detector are mounted on a platform which is driven with an L3C linear motor. The linear motor is driven with the 430-FOL and has a resolution of 7500 steps/in. A high resolution encoder is mounted to a roller on the belt and is geared to also have 7500 steps/in after quadrature detect in the 430. Two move segments are required. Segment 1 accelerates to following ratio over 2 inches of motor travel and 4 inches of belt travel to come into synch 2 inches from the edge of the part. Segment 2 decelerates to zero ratio over 2 inches of motor travel and 4 inches of belt travel. Between welds, segment 2 and segment 1 are executed back to back for a total motor travel of 4 inches and a total belt travel of 8 inches. This gives the required 4 inches of recess between welds. After the second weld, segment 2 is executed to bring the motor to a stop before the retrace move is made. Outputs and delays are used between the segments to control the weld. A standard 430 move is used to return to the start position for the next weld.

2.2.3. The Program

Reg #1	Reg #2	Reg #3	Reg #4	Cmd Reg
0001	5000	0003	0000	0286
0001	0000	0001	0000	0201
0002	0000	0000	0000	0201
0015	0000	0000	0000	0217
0011	0000	0010	0000	0097
0016	0000	0000	0000	0033
0001	0000	0000	0000	0044
0000	0001	0000	0000	0096
0001	0000	0000	0000	0104
0001	0000	0000	0000	0011
0000	0000	0001	0000	0042
0001	0000	0000	0000	0012
0000	0000	0000	0000	0105
0002	0000	0000	0000	0104
0001	0000	0000	0000	0104
0001	0000	0000	0000	0011
0000	0000	0001	0000	0042
0001	0000	0000	0000	0012
0000	0000	0000	0000	0105
0002	0000	0000	0000	0104
0000	0000	0000	0000	0096
0000	0000	0000	0000	0052
0000	0000	0000	0000	0216
0000	0001	0000	0000	0293
0000	0000	0000	0000	0291

Line 1) Ramp to ratio over 15000 motor steps and 30000 encoder steps.
Line 2) Define segment #1 with a final ratio of 1:1.
Line 3) Define segment #2 with a final ratio of 0:1.
Line 4) Begin definition of sequence #15.
Line 5) Set follower parameters (update rate to 10 ms).
Line 6) Repeat the next 16 commands continuously.
Line 7) Wait for trigger 1 to go low.
Line 8) Enter FOL mode.
Line 9) Perform segment #1.
Line 10) Turn on output #1.
Line 11) Wait for 1 second.
Line 12) Turn off output #1.
Line 13) Initialize reference counts.
(cont. next page)

```
Line 14)  Perform segment #2.
Line 15)  Perform segment #1.
Line 16)  Turn on output #1.
Line 17)  Wait for 1 second.
Line 18)  Turn off output #1.
Line 19)  Initialize reference counts.
Line 20)  Perform segment #2.
Line 21)  Exit FOL mode.
Line 22)  Return to absolute positon 0.
Line 23)  End sequence 15 definition.
Line 24)  Enable AutoRecall.
Line 25)  Save Program.
```

0096 (Enter/Exit FOL)

Data Words: R2
Type: Sequential, Dominant
Description: Enter/Exit FOL mode. A value of 0 (0000) in Reg #2 will cause the 430 to Exit FOL mode. A value of 1 (0001) in Reg #2 will cause the 430 to enter FOL mode. This command must be issued with a 1 in Reg #2 before any FOL segments can be executed. It is possible to enter and exit FOL mode to perform both FOL segments and standard moves within a sequence.

See also: 0097 (Set Parameters), 0098 (Specify Change Interval), 0106 (Change to Specified Ratio), 0192 (Specify Thumbwheel Scale factor), 0193 (Change to Thumbwheel Ratio)
Example: 0000 0001 0000 0000 0096 (FOL mode is enabled)

0097 (Set Parameters)

Data Words: R1,R2,R3
Type: Sequential
Description: Set Follower Parameters. R1 = abcd, where R1a is not used, R1b = Set Velocity Range (0 - 2) See table R1b. R1c = Set Sample Correction Attenuation, where c = 0 - 7. A value of 0 for c will result in very fast correction, but will tend to amplify errors arising from a finite sample period, giving jerky motion. Typically a value of 1 or 2 will be appropriate for most applications. R1d = Enable/Disable Sample Averaging, where d = 1 (enable), or d = 0 (disable). Sample Averaging compensates for roughness due to round off errors, innaccuracies in the encoder and/or its' feedback, by averaging the velocities from the last two samples to derive the true output velocity. R2 and R3 determine the rate* (in milliseconds) at which the encoder feedback will be sampled by the 430 (min = 10 ms).

See also: 0096 (Enter/Exit FOL mode), 0098 (Specify change interval)
Example: 0110 0000 0000 0010 0097
(Execution of the 97 command with the registers assigned in this manner will cause the 430 to sample the encoder input every 10 ms, set the velocity range at 0 to 50 Khz, and set the sample attenuation factor to 1. Sample averaging is not enabled in this example.)

Table R1b

b	Velocity Range	Resolution
0	0 - 500 Khz	15.0 Hz
1	0 - 50 Khz	1.50 Hz
2	0 - 5 Khz	0.15 Hz

* The maximum number of encoder pulses that the 430 FOL can count in one sample period is 32,767.

0098 (Specify Interval)

Data Words: R2, R3, R4
Type: Sequential
Description: The number represented by R2, R3 and R4 specifies the number of encoder steps over which the next Ratio change specified by the 0106 command or the 0193 command will take place.

See also: 0193 (Change to thumbwheel ratio), 0106 (Change to specified ratio).

Example: 0000 0000 0001 0000 0098 (The next ratio change (0106 or 0193 cmd) will take place over 10,000 encoder steps.)

0100 (Defn cycle length)

Data Words: R2,R3,R4
Type: Sequential
Description: Defines the number of encoder steps in a cycle. The 430 maintains an internal cycle position between 0 and the defined cycle length - 1.

See also: 0101 (Set cycle position), 0107 (Correct to cycle position).
Example: 0000 0000 0000 6000 0100 (Cycle length is set to 6000 encoder steps.

0101 (Set cycle positn)

Data Words: R2,R3,R4
Type: Sequential
Description: Instantly over-rides the internally maintained cycle position with the position specified by R2, R3, R4.

See also: 0100 (Define cycle Length), 0107 (Correct to cycle position).
Example: 0000 0000 0000 0000 0101 (Cycle position is set to 0.)

0104 (Perform Segment)

Data Words: R1
Type: Sequential, Dominant.
Description: Perform predefined FOL move segment. Where the value in R1, corresponds to the number assigned to the desired predefined FOL segment.

See also: 0201 (Define FOL segment), 0286 (Specify FOL distance).
Example: 0001 0000 0000 0000 0104 (FOL segment #1 will be executed, provided it has been predefined.)

0105 (Initialize Ref Cnt)

Data Words: None
Type: Sequential, Dominant.
Description: When this command is executed, the 430 reads the absolute encoder, and absolute motor position and uses the values as a reference for the next FOL move segment. A typical use of this command would be immediately following a trigger or time delay and immediately before a new "Execute FOL segment" command. This command is neccesary for the 430 to ramp to a new ratio from a segment of indeterminate length (i.e. A segment waiting on a trigger or time delay.).

See also: 0104 (Perform FOL segment), 0106 (Change Ratio).
Example:

```
0001  0000  0000  0000  0044  (When Trigger #1 is in)
0000  0000  0000  0000  0105  (a high state Initialize)
0000  0001  0000  0000  0106  (reference counts, and)
                              (ramp to a Ratio of)
                              (zero).
```

0106 (Change Ratio)

Data Words: R2, R3, R4
Type: Sequential, Dominant
Description: Change to specified Ratio. Execution of this command will cause the 430 to ramp the "slave" motor to the Ratio specified by R3, R4, where R3 is the integer portion of the Ratio (valid range = 0000 - 0255), and R4 is the fractional portion of the Ratio in ten thousandths (valid range = 0000 - 9999). A value of 0 in R2 will cause the slave motor to travel in the same direction as the encoder, A value of 1 in R2 will cause the "slave" motor to travel in the opposite direction of the encoder. Issuing this command will cause an FOL move segment to be executed.

See also: 0286 (Specify FOL segment distance).
Example: 0000 0001 0002 0000 0106 (Motor will ramp to a Final Ratio of 2).

0107 (Correct to cyc pos)

Data Words: R2,R3,R4
Type: Sequential
Description: Correct to cycle position. When executed, this command will cause the 430 to read the current cycle position and compare it to the position specified by R2, R3, and R4. The difference is multiplied by the current ratio of motor to encoder steps to arrive at the number of motor steps to be corrected. The motor step error is corrected with a "hiccup" in the current ratio over the number of encoder steps specified by the 0098 command (specify interval). The maximum error corrected per "hiccup" is limited to half the encoder change length.

See also: 0100 (Define cycle Length), 0101 (set cycle position).
Example: 0000 0000 0000 3000 0107 (Motor will begin correcting to cycle position 3000).

0192 (Thmbwhl Scale Fctr)

Data Words: R3, R4
Type: Sequential
Description: Set Thumbwheel Scale Factor. The number specified by R3, R4 will be the multiplier used with the thumbwheel ratio specified on the Data inputs when the 0193 command is executed. Where R3 is the integer portion of the multiplier, and R4 is the fractional portion of the multiplier in ten thousandths.

See also: 0193 (Change to thumbwheel ratio).
Example: 0000 0000 0002 5000 0192 (Multiply ratio specified by thumbwheels by 2.5)

0193 (Chng to Thwl Ratio)

Data Words: None
Type: Sequential, Dominant.
Description: Change to Ratio equal to Thumbwheels (AB.CD) (where A is the BCD value of DS16-DS13, B is the BCD value of DS12-DS9, C is the BCD value DS8-DS5, D is the BCD value of DS4-DS1) * Thumbwheel scale factor set with 0192 command.

See also: 0192 (Set Thumbwheel scale factor).
Example: 0000 0001 0002 0000 0106
0001 0000 0000 0000 0044
0000 0000 0000 0000 0105
0000 0000 0000 0000 0193 (motor will ramp to ratio = number specified on DS1-16 * Scale factor set with 0192 command, from a ratio of 2, when Trigger 1 goes low.)

0201 (Define FOL segment)

Data Words: R1, R2, R3, R4
Type: Sequential
Description: Define FOL move segment. This command is used to predefine an FOL move segment for later execution. Where the number in R1 is the designator for the move segment. The value of R2 determines direction (0 = Same direction as encoder, 1 = Opposite direction of encoder). The value of R3 is the integer portion of the Final Ratio (max value =). The value in R4 is the fractional portion of the Final Ratio in ten thousandths. The 0286 command (specify FOL segment distance) must be issued prior to the execution of this command.

See also: 0286 (specify FOL distance), 0104 (perform predefined Fol segment)

Example:

```
0005  0000  0005  0000  0286  (FOL segment #1 is
0001  0000  0002  0000  0201  (defined as a segment of
                              (50,000 motor steps over
                              (50,000 encoder steps,
                              ((Average Ratio - 1),
                              (with a Final Ratio of
                              (2:1.
```

0286 (Define sgmnt dstnc)

Data Words: R1, R2, R3, R4
Type: Immediate
Description: Define segment distance. This command is used in conjunction with the 0201 command (Define FOL move segment) to predefine FOL move segments. This command defines the number of encoder and motor steps for the specific move segment. R1 and R2 specify the number of motor steps that the slave motor will move during the segment. R3 and R4 specify the number of encoder steps the 430 will see during the segment. This command will specify the Average Ratio of the segment by virtue of both the encoder and motor steps being defined. This command must be issued before a 0201 command can be executed.

See also: 0201 (Define FOL segment), 0104 (Execute FOL segmemt)
Example:

```
0005  0000  0005  0000  0286  (FOL segment #1 is
0001  0000  0002  0000  0201  (defined as a segment of
                              (50,000 motor steps over
                              (50,000 encoder steps
                              ((Average Ratio = 1),
                              (with a Final Ratio of
                              (2:1.
```