KOLLMORGEN

www.DanaherMotion.com

SERVO**STAR**® S- and CD-Series Rotary Indexer Mode

Enhancements to the SERVO**STAR**® S- and CD-series products are included in firmware version 4.0.0 and higher that provide flexibility in rotary indexing applications. Rotary indexing systems are characterized by a load driven by the motor where position within one revolution of the load is important, but total revolutions the load has traveled is of no consequence . These application include:

- Rotary Index Tables for CNC and Packaging
- Gear Cutting and Hardening
- C-Axis Material Handling
- Balancer Machines.

In these applications, absolute positioning is commonly preferred to incremental indexing and it is desirable to have the position counter value roll over at each revolution of the load. This feature is sometimes termed “arithmetic modulo” or simply “modulo”.

Consider a machine designed to drill a bolt pattern for wheels used on vehicles. A wheel is placed into the machine where the first hole is drilled. Upon command, the table then rotates the wheel to the next location for drilling. The process continues until the wheel returns to its starting location where the wheel is replaced and the process begins anew. In this application we care only about the bolt pattern spacing and nothing about how many revolutions the machine has accumulated.

Problems can occur when using normal positioning methods for this application. If we use absolute positioning to make the moves for the hole patterns, then the machine must return to 'home' each time the wheel is replaced. This loses through-put efficiency since the path back to “home” would be in the opposite direction as well as requiring that the machine be designed for bi-directional movement.

Using incremental indexing for the application also has its problems. First, there is no method of homing the machine or moving it to a known reference point to align the hole pattern to a wheel registration location. Secondly, incremental indexing only works if the gear ratio (if gearing is used) of the motor to the table is chosen very wisely with consideration given to the number of divisions and 'creeping' (accumulated gain or loss of position) that can occur. In some cases, it is still impossible to accommodate a wide variety of divisions.

DANAHER MOTION

Applications

Rotary Indexer Mode enables two basic applications types: those requiring the position counter (PFB) to roll over (count between 0 and some distance) at a desired distance such as after one revolution and those requiring the division of a revolution into symmetrical stopping points. The SERVO**STAR** now features Modulo and Rotary Index modes of operation to address these applications.

Modulo Mode

The SERVO**STAR** Modulo mode (see MODMODE and PROTARY) is designed to allow the drive to overcome the problems associated with counting the turns that the rotary table moves. The MODMODE operation mode allows the user to define a PFB roll-over point (PROTARY) the firmware uses for 'graceful' roll-over. PFB and PCMD are affected in the roll-over process. When exceeding the roll-over point, they are decremented by the roll-over value, keeping these two variables in the range of 0 to PROTARY and acting as an arithmetic modulo function. When using indexing commands such as MI (Move Incremental) - an indexed move that goes past the PROTARY position value implements the move while the drive keeps PCMD and PFB position values between zero and PROTARY. An absolute position command, MA (Move Absolute), calculates the *shortest path* to reach that position and moves accordingly while again keeping PCMD and PFB position values between zero and PROTARY. To enable this mode, set MODMODE=1 and define the counts the motor moves to move the table 360 degrees by setting PROTARY = counts per revolution. Once these two parameters have been set, the SERVO**STAR**'s position counter (or PFB - the variable that tracks absolute motor position) 'gracefully' rolls over at each revolution.

The advantage of moving the shortest distance to the target position may not be obvious at first, but consider the example above. When the machine drilled its last hole in the bolt pattern, the application wants it to return to the starting location (home or absolute 0). Executing an MA 0 200 command causes the machine to traverse in the same direction it had been operating (through a small angle of 45 degrees) instead of rotating all the way back around (315 degrees) to finish the job. Another advantage becomes obvious when a machine can be typified as a balancer. This machine rotates the load many revolutions at a high speed. After the process performed at speed is complete, the machine needs to position the part at a reference location. Under MODMODE=1 mode of operation, when an absolute position command is executed to get the part where it needs to be, the drive's position loop moves it directly to that position regardless of how many turns had occurred during the actual balancing operation. *(This operation is actually available in firmware versions 3.6.4 and higher.)*

Rotary Index Mode

Consider the above example once again to understand another problem. If the motor is coupled to the load by a 5:1 gear ratio (or any one of a number of choices), the system 'creeps' when using incremental indexing. This is because the motor feedback device has a range of 2^{nth}, where *n* is typically (but not always) 16, resulting in 65,536 parts per revolution. The number 65,536 does not divide by 5 with an integer result. Using incremental indexing for 6 holes in the part, you must index the load 54,613.33333 bits per move (65,536 bits/motor rev x 5 motor revs/table rev x 1 table rev/6 holes). Here, you would choose 54,613 or 54,614. You loose (or gain) position each index and, by the end of the day, the table is far out of position. Additionally, the 6^{th} hole would accumulate the error from each previous index.

The SERVO**STAR** Rotary Index Mode of operation overcomes this problem of 'creep' and is automatically enabled when MODMODE = 1. The number of holes (or stops) per revolution is then entered in the DIVISIONS variable. The SERVO**STAR** performs the appropriate arithmetic functions to keep the error at each stop to within an LSB (least significant bit) of the feedback and prevents accumulated error through multiple revolutions of the machine.

Further support for these modes of operation is available through enhanced digital input (command) functionality as follows:

INxMODE = 22: Assign a digital input to trigger the move defined by 'Divisions'

INxMODE = 23: Assign a digital Input to trigger a move twice that as defined by 'Divisions' (allows the machine to skip a stop location for asymmetrical requirements).

In all cases of Rotary Indexer Mode, the motor's velocity is defined by the variable 'MSPEED0'.

Finally, a new 'homing' (Power up initialize) feature was added to the SERVO**STAR** firmware to help overcome I/O limitations. This feature, called AUTOHOME, when enabled allows the motor to perform the homing routine (HOMETYPE) without having to assign an input for this function.

Variables

MODMODE: Causes the rotary mode to be enabled. This causes the internal position loop counters to roll-over at the PROTARY value.
Range: 0, 1.
0 - Rotary mode not enabled (default).
1 - Rotary mode enabled.
Default: 0.
Saved in EEPROM

PROTARY: Defines the modulo (rollover) value of PFB and PCMD when MODMODE is enabled.
Range: 1000 .. 2^30
Default: 2^21
Saved in EEPROM

DIVISIONS: Sets the number of divisions used for rotary mode indexing causing the index distance to be PROTARY / Divisions with care to round-off and accumulation errors.
Range: 2 to 32767, and -2 to -32767.
Negative value indicates CCW direction.
Default: 4.
Saved in EEPROM

New INxMODE:

22 - triggers incremental move. Distance equals (PROTARY / DIVISIONS).
23 - triggers incremental move. Distance equals (2 * PROTARY / DIVISIONS).
In both cases (22 and 23):
The motion velocity is MISPEED0.
Zero speed causes no movement. A momentary 'b' is displayed.

AUTOHOME: Defines the homing behavior on power up.
Range: 0 to 2.
0 - no homing on power up (default).
1 - try to perform homing on power up. If the homing process can not be executed, then forget it.
2 - try to perform homing on power up. Keep try until the homing process is executed.
Default: 0.
Saved in EEPROM

1.) 'Hold' position causes no movement. A momentary 'b' is displayed.
2.) The direction depends on the sign of DIVISIONS and on the value of DIR.