AN147

16-131636

Trajectory Sequencing

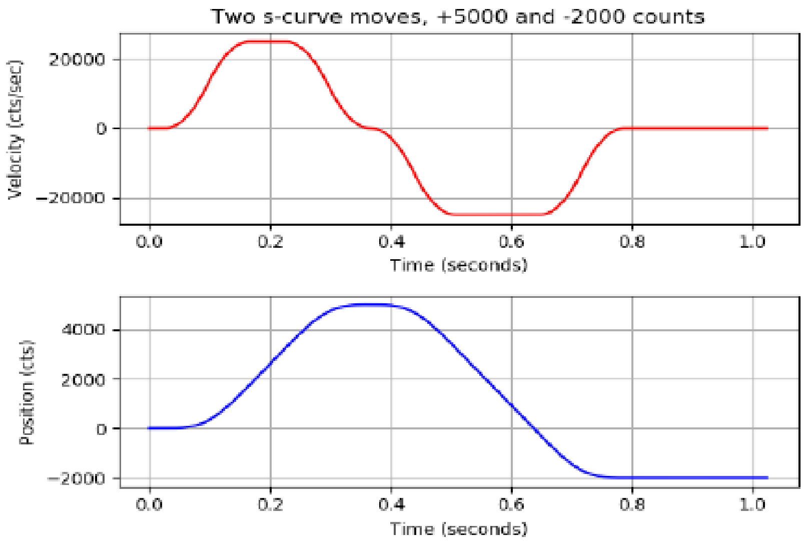
Introduction
Copley's Plus family of servo drives supports the ability to queue up a sequence of moves and execute them one after another automatically. This feature was added starting with version 2.34 firmware for Plus drives.

This type of trajectory sequencing was initially added to support the standard "set of setpoints" feature defined in the DS402 specification for CANopen and EtherCAT drives. When controlling the drive in profile position mode over either CANopen or EtherCAT, the standard method of queuing up moves and starting them defined in the DS402 specification should be used.

This feature can also be used when the drive is being controlled using the serial port with either the binary or ASCII serial commands. This document primarily focuses on trajectory sequencing via the serial port.

Overview
Drives always maintain an 8-level deep buffer of move parameters. Trajectory sequencing works by storing a set of one or more moves to this buffer and then starting motion. As each move finishes, the drive will automatically start the next move with no additional delay. This allows several moves to be performed in sequence.

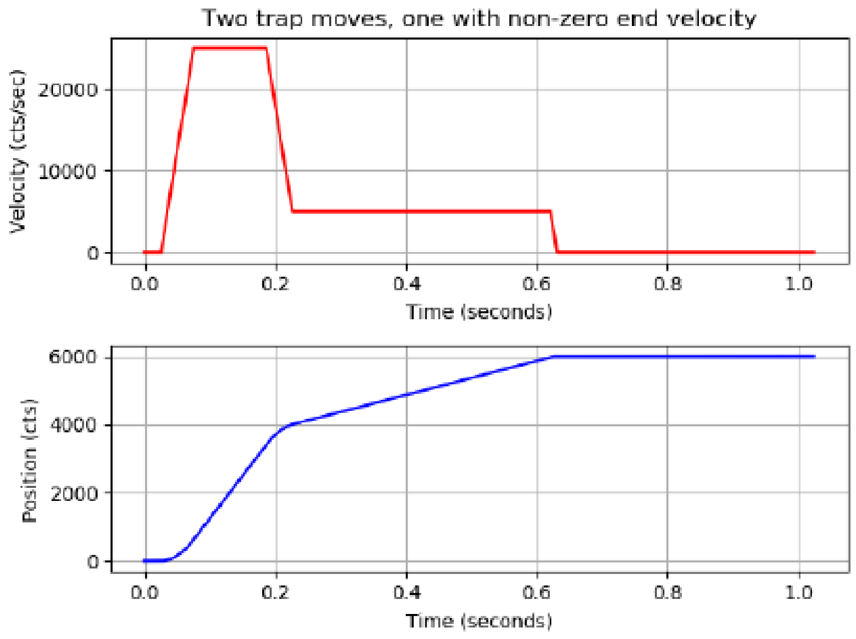
For example, the graphs below show the result of sequencing two s-curve moves. The first move is in the positive direction for 5000 encoder counts, followed by a move in the negative direction for -2000 counts. The second move starts immediately after the completion of the first move.



Additionally, the trapezoidal trajectory generator supports an end velocity parameter which can be used to finish one move at a specified position with a non-zero velocity, and immediately start the next move from that position and velocity.

Note that only the trapezoidal profile generator supports non-zero starting or ending velocities. If the s-curve profile generator is used, then any ending velocity will be ignored. If an attempt is made to start an s-curve profile when the velocity is not zero (such as in a sequence after a trap move with a non-zero ending velocity), the s-curve move will force the velocity to zero at the start of the sequencing.

The image below shows an example of two trapezoidal moves made in a sequence in which the first move has a non-zero ending velocity. The first move is 4000 counts with an ending velocity of 5,000 cts/sec. The second move is 2000 counts with a max velocity of 5,000 cts/sec and a zero ending velocity. The first move is completed at position 4000 and the non-zero velocity, and the second move starts immediately from that point and continues on to position 6000 where it ends with zero velocity.



Programming Sequences

Trajectory sequences can be programmed using either the binary or ASCII serial port protocol. While it is possible to use profile position mode when controlling the drive over CANOpen or EtherCAT, that is outside of the scope of this document.

Through either serial port protocol, the basic series of commands are the same. First, the various move parameters are set, then the trajectory is saved in the drive's trajectory buffer. Once all trajectories have been saved to the buffer, the sequence is started.

The following parameters are used to configure a move. When a trajectory is saved to the internal trajectory buffer, each of these parameters is copied to the buffer for later use. Please refer to the Parameter Dictionary for additional information about these parameters.

Parameter	Description
0xC8	Trajectory mode. This defines the type of trajectory (trapezoidal, s-curve, etc) and whether the move is to an absolute position, or relative to the starting point of the move.
0xCA	Gives the destination position of the move for absolute moves, or the relative distance for relative moves. Units of encoder counts.
0xCB	Gives the maximum velocity for the move in tenths of encoder counts / sec (i.e. 123 would be 12.3 cts/sec).
0xCC	Gives the maximum acceleration for the move in tens of counts / sec / sec (i.e. 123 would be 1230 cts/sec^2).
0xCD	Gives the maximum deceleration for trapezoidal and velocity moves. This parameter is not used for s-curve trajectories. For s-curves the acceleration and deceleration are the same. Same units as 0xCC.
0xCE	Gives the maximum jerk (rate of change of acceleration) for s-curve moves in units of 100's of counts/sec^3.
0x11B	Ending velocity for trapezoidal moves. Same units as 0xCB.

A few additional parameters are useful for working with the trajectory buffer.

Parameter	Description
0x11C	Status of the trajectory buffer. This is a 32-bit read-only value which is bit-mapped as follows:

Bits	Description
0-7	Gives the number of free positions in the buffer
8-15	Gives the number of full positions in the buffer
16-31	Reserved

Once the trajectory limits have been set using the parameters above, the trajectory can be started or added to the buffer using the trajectory update command. When using the binary serial protocol, the trajectory update command uses op-code 0x11. When using the ASCII protocol, the trajectory update command starts with the single letter T.

No matter which protocol is used to send the trajectory update command, this command takes a single 16-bit integer parameter which identifies the type of update to perform.

The following values are supported:

Value	Name	Description
0	Abort	Causes any currently executing trajectory to be aborted. This also causes the trajectory buffer to be flushed.
1	Move	This command first flushes the trajectory buffer if there was anything stored in it. It then starts a new move or updates the currently executing move using the current value of the various trajectory configuration parameters. This is the normal start/update command when the trajectory buffer is not being used.
2	Home	This first flushes the trajectory buffer and then starts a homing sequence.
3	Save	This stores the current values of the trajectory parameters in the next free buffer location.
4	Start Sequence	This starts the trajectory sequence stored in the buffers.

Example

The following set of ASCII commands can be used to create a move like the one shown in the graph on page 2. This move consists of two trapezoidal moves with a non-zero velocity at the end of the first move.

First, set all of the following parameters to define the first move in the sequence. This is a relative trapezoidal move of 4000 counts. Max velocity is 25,000 cts/sec, accel and decel are both 500,000 cts/sec^2, and ending velocity is 5000 cts/sec:

```
s r0xC8 0x0100
s r0xCA 4000
s r0xCB 250000
s r0xCC 50000
s r0xCD 50000
s r0x11B 50000
```

Next, use the trajectory command (simply T in ASCII) with parameter 3 to add this to the trajectory buffer.

```
T 3
```

Next, program the second move. Only the distance (2000 counts), max velocity (5000 cts/sec) and ending velocity (0) are programmed. The trajectory mode and accel/decel are the same as the previous trajectory segment so there is no need to reprogram them.

```
s r0xCA 2000
s r0xCB 50000
s r0x11B 0
```

Finally, add this segment to the sequence buffer and start the sequence using two trajectory commands:

```
t 3
t 4
```

The sequence of commands for the binary serial protocol are the same. For details on how to send binary serial commands, please refer to AN112 – Binary Serial Interface Operation.

Revision History

Date	Version	Revision
4/30/2021	Rev 00	Initial release