12 Cubic Spline Programming

Introduction

Motion control applications requiring fine moves through a set of points require cubic spline interpolation. The Mx4 can run cubic splines either in contouring mode (in which the host continually updates Mx4's DPR with a new set of points), or in table mode (Mx4's table is pre-loaded with a set of points only once). In table mode the user array can be up to 4,096 points long. Each point specifies the position of only one motor.

The DSPL commands useful for cubic spline applications are:

`CUBIC_RATE` Specifies the “time” interval between the two adjacent points in a cubic spline table. This instruction is similar to BTRATE (used in dual port RAM-based contouring applications).

`CUBIC_SCALE` Specifies," position_multiplier” and “position_shift” for all points of a spline table.

`CUBIC_INT` To run on "m" points of cubic spline table, starting from “si” starting index, "n" number_of_times, with the specified axes.

Three Steps to Run Cubic Spline

1) Download the data points using the *Tables* option in Mx4pro v5.

   Also, the DSPL offers floating point arithmetic and trigonometric functions by which new move parameters can be calculated in real time and stored in the table memory.

2) Run the DSPL command `CUBIC_RATE`. This command ***must*** run before issuing `CUBIC_INT`.

3) Use `CUBIC_INT` in your DSPL or host-based program.

We will now discuss six DSPL programs -- starting from simple leading to more advanced applications.

Cubic Spline Trajectory on A Single Axis

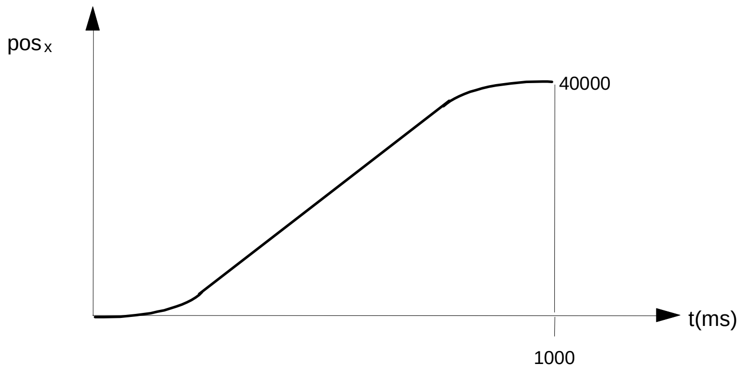
Consider a single axis move as illustrated. This trajectory is characterized by its position at times starting at zero and incrementing every 100 ms. In order to perform cubic spline contouring you must follow the steps as follows:

Step 1: Generate points

Step 2: Create an ASCII file that contains the points and download it to Mx4

Step 3: In your DSPL program use relevant instructions:

```
CUBIC_RATE()
CUBIC_SCALE()
CUBIC_INT()
```



This example helps you understand how a data table is organized.

The Data File for One-Axis Contouring Process

You need to generate an ASCII file similar to the following and save it under any name followed by .DAT, (e.g., CUB1.DAT).

Position (counts)

```
0.00000000000000e+000
1.25000000000000e+003
5.00000000000000e+003
1.00000000000000e+004
1.50000000000000e+004
2.00000000000000e+004
2.50000000000000e+004
3.00000000000000e+004
3.50000000000000e+004
3.87500000000000e+004
4.00000000000000e+004
3.87500000000000e+004
3.50000000000000e+004
3.00000000000000e+004
2.50000000000000e+004
2.00000000000000e+004
1.50000000000000e+004
1.00000000000000e+004
5.00000000000000e+003
1.25000000000000e+003
0.00000000000000e+000
```

You may now download all (21) points to the Mx4 memory.

Memory Capacity

The Mx4 memory size dedicated to cubic spline is 8,192 words. Each point on cubic spline contour is characterized by its (32-bit) position thus requiring two words. Therefore, the total number of points that may be saved in an Mx4 cubic spline table is 4096.

Downloading a Table

Use the Table download facility in Mx4pro v4 on Windows 95/NT. Click on *Tables|Cubic Spline|File|Open* to find and select the file containing the table. Use the *Edit Table* function to make changes to the file. Double click the Offset box and type in the starting index of the table in the Mx4 table memory (this allows multiple tables to be defined and used simultaneously). Click the Download button to transmit the file to the Mx4 board. The number appearing in the adjacent box is the number of points in the file.

DSPL Program

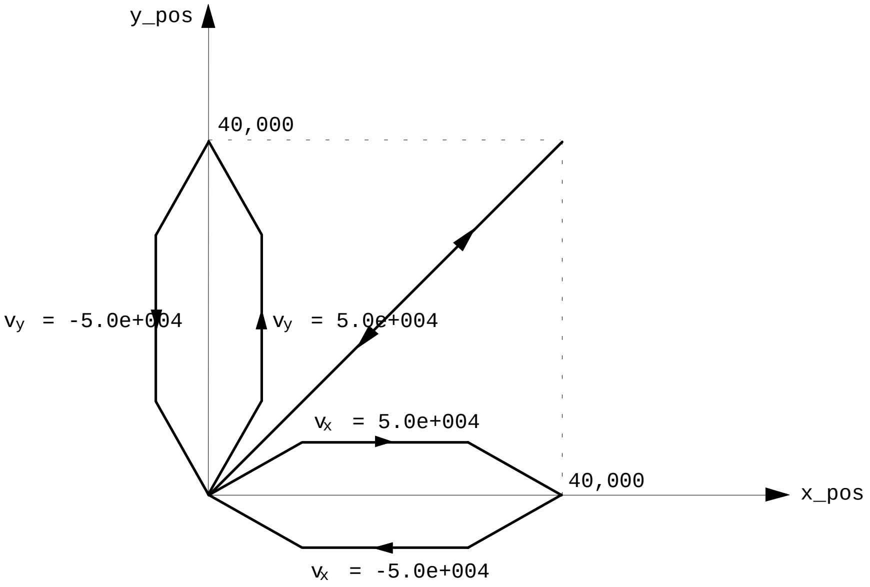
The steps following the transmission of the data table includes setting block transfer rate (`CUBIC_INT`), scaling (`CUBIC_SCALE`) and, running through the points (`CUBIC_INT`).

The following illustrates the DSPL program that runs through 21 points of `cub1.dat`.

```
plc_program:
  run_m_program(cubic)
end


cubic:

  ctrl(0x1,1000,15000,5160,3140)      ; set the gains
        pos_preset(0x1,0)             ; preset x and y
        maxacc(0x1,1)                 ; set stop accel

  cubic_rate(500)                     ; set the cubic spline time
                                      ; interval to 100ms
  cubic_scale(0x1,1,0)                ;set the position scale to 1
                                      ; with no shift
  cubic_int(21,0,1,0x1)               ;run 21 points of the table
                                      ; only once
end
```

Cubic Spline Trajectory on Two Axes

This example is similar to the first one and is only modified for two axes. Our objective here is to show how the data points for an additional axis must appear in the data file.



To simplify our presentation, we use similar motions for x and y. In a general case x and y may have any arbitrary shape.

ASCII File for Two-Axis Contouring Process

X Position (counts)	Y Position (counts)
0.00000000000000e+000	0.00000000000000e+000
1.25000000000000e+003	1.25000000000000e+003
5.00000000000000e+003	5.00000000000000e+003
1.00000000000000e+004	1.00000000000000e+004
1.50000000000000e+004	1.50000000000000e+004
2.00000000000000e+004	2.00000000000000e+004
2.50000000000000e+004	2.50000000000000e+004
3.00000000000000e+004	3.00000000000000e+004
3.50000000000000e+004	3.50000000000000e+004
3.87500000000000e+004	3.87500000000000e+004
4.00000000000000e+004	4.00000000000000e+004
3.87500000000000e+004	3.87500000000000e+004
3.50000000000000e+004	3.50000000000000e+004
3.00000000000000e+004	3.00000000000000e+004
2.50000000000000e+004	2.50000000000000e+004
2.00000000000000e+004	2.00000000000000e+004
1.50000000000000e+004	1.50000000000000e+004
1.00000000000000e+004	1.00000000000000e+004
5.00000000000000e+003	5.00000000000000e+003
1.25000000000000e+003	1.25000000000000e+003
0.00000000000000e+000	0.00000000000000e+000

Save this ASCII file as CUB2.DAT and download it to the Mx4 memory.

DSPL Program for Two-Axis Contouring

The following illustrates the DSPL program modified for two motors.

```
plc_program:

        run_m_program (cubic_spline)

end

cubic_spline:

    ; set the gains
    ctrl (0x3,1000,15000,5160,3140,1000,15000,5160,3140)

    ; preset x and y position
    pos_preset(0x3,0,0)


    ; set stop accel rate
```

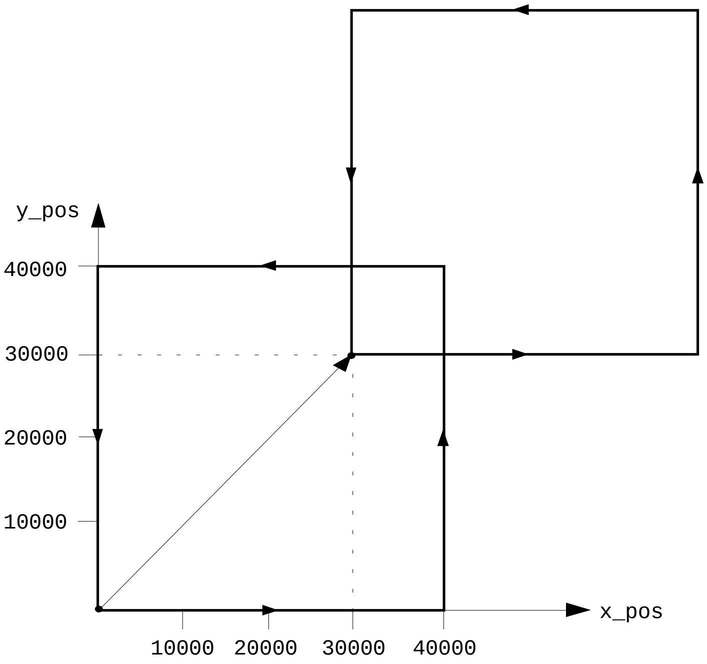
```
   maxacc(0x3,1,1)

   ; set time interval
   cubic_rate(300)

   ; set scale for points
   cubic_scale(0x3,1,0,1,0)

   ; start interpolation
   cubic_int(42,0,0,0x3)
end
```

Dynamic Scaling and Coordinate Transformation

Motion control applications involving cubic spline may be scaled or coordinate transformed. Scaling means the real-time multiplication of "all" positions by a set value. This feature may be used to change the size of the position vector. The velocity vector is modified accordingly.

Coordinate transformation (shift) performs the real-time position shift of Cartesian coordinates. That is, this command, in conjunction with cubic spline, will shift the position of all axes to a new origin. The RTC used for this task is CUBIC_SCALE.

Consider our previous example, in which the system continually repeats the same motion. Now imagine that after cutting a shape, the operator wishes to transform the coordinates to a new origin specified by its positions in x and y directions (e.g.,30000,30000).



The following command shows how this coordinate transfer is accomplished:

```
CUBIC_SCALE(0x3,1,30000,1,30000)
```