Title: MultiFlex Spiral Motion Path Generation
Products(s): All MultiFlex ETH and PCI motion controllers
Keywords: Spiral Motion, Contour Motion, Pulse Axis, MCCL
ID#: TN1085
Date: August 16, 2013

Summary

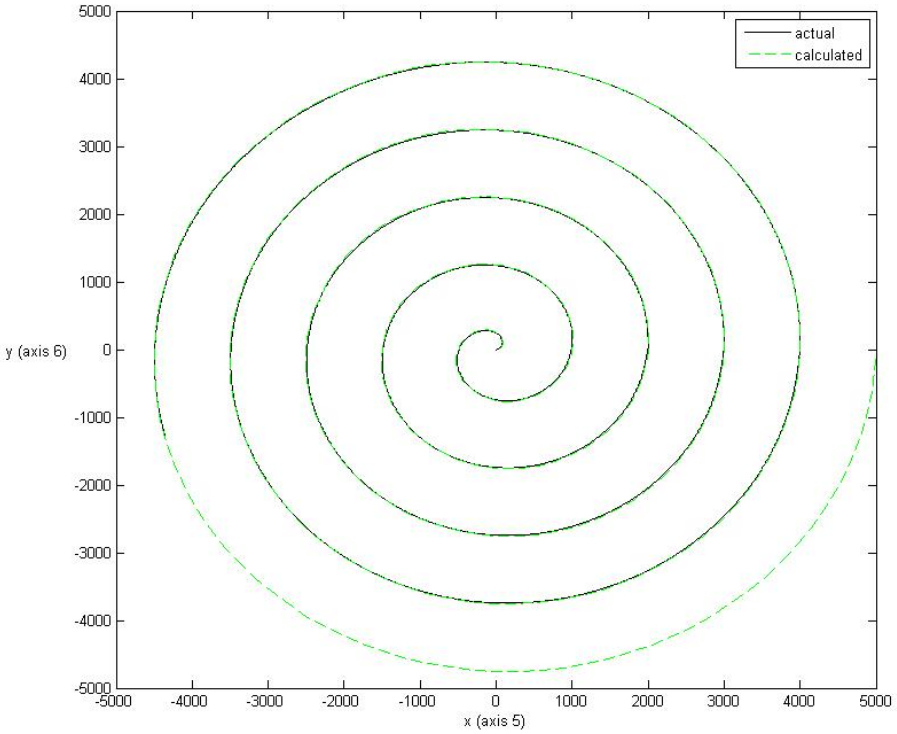
Following is a description of a method for generating spiral motion on PMC Multiflex controllers. This technique utilizes the controller's contouring feature to produce a spiral described by

$$r = a + b\Theta$$

where b is the radial increase constant.

Overview

The figure below shows a spiral with b = 1000 produced by the Multiflex controller. The actual x-y position data has been superimposed over the calculated points representing the ideal curve.



The example MCCL macro code was used to produce the spiral motion where x axis is pulse axis 5 and the y axis is pulse axis 6. The axes are configured as a contour pair to allow smooth motion from point to point.

The angle theta is incremented from 0 to 2Π in 200 steps in each circular cycle. At each step, the radius, sine(Θ), cos(Θ) and the x and y coordinates are calculated and inserted into the contour buffer. In this example, the radial increase is set to 1000/cycle and five cycles are executed.

The macro code in the following section can be transferred into the Multiflex controller and the example motion can be executed by a call to macro 200. The radius can be adjusted by changing the value placed in register 99 shown below by

al1000,ar99

Spiral Motion MCCL Macro Implementation

```
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; axis initialization
;
; axis reset, motion parameter initialization
; set axis 6 contoured to axis 5
;
md2,5mf,6mf,5rt,6rt,5dh,6dh
md3,5mn,6mn,5pm,6pm
md4,5sa500000,5ds500000,5sv200000
md5,6sa500000,6ds500000,6sv200000
md6,5mn,6mn,5cm5,6cm5
;
; call initialization macros set vector velocity
;
md20,mc2,mc3,mc4,mc5,mc6,5vv20000
;
;
;
```

```
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; spiral motion calculation
;
; r = [radial increment / (2 * pi)] * theta
; x = r * cos(theta)
; y = r * sin(theta)
;
; register usage
;
;        r90 - theta               (init = 0)
;        r91 - theta increment     (init = 2 * pi / 200)
;        r92 - const               (init = 2 * pi)
;        r93 - radius              (init = 0)
;
;        r94 - sin(theta)          (init = 0)
;        r95 - cos(theta)          (init = 1)
;
;        r96 - x position          (init = 0)
;        r97 - y position          (init = 0)
;
;        r98 - radial increment    (init = (r99) / (2 * pi))
;        r99 - spiral radius       (init = 1000)
;
;
; set r99 to spiral radius value
; and perform register initialization
;
al1000,ar99
md40,al0,ar90,ar93,av23,ar92,ad200,ar91,al@99,ad@92,ar98
;
;
; calculations
;
;        m42 - increment theta
;        m43 - sin(theta) cos(theta) radius
;        m44 - x and y position
;
md42,ra90,aa@91,ar90
md43,ra90,av11,ar95,ra90,av10,ar94,ra90,am@98,ar93
md44,ra95,am@93,ar96,ra94,am@93,ar97
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;
;
```

```
;
;
; single-cycle spiral position move
;
md50,mc42,mc43,mc44,wa0.005
md51,5cp3,5ca0,6ca0,5ma@96,6ma@97
md52,mc50,mc51,rp199
;
;
; 5-cycle spiral motion
;
md200,mc20,mc40,mc202
md202,mc52,5ws,mc510,rp4
;
; dump registers
;
md500,tr90,tr91,tr93,tr94,tr95,tr96,tr97,tr98
md501,ra90,od"theta:  %f\r"
md502,ra93,od"r:      %f\r"
md503,ra94,od"sin(t): %f\r"
md504,ra95,od"cos(t): %f\r"
md505,ra96,od"x:      %f\r"
md506,ra97,od"y:      %f\r"
md510,mc501,mc502,mc503,mc504,mc505,mc506
;
```