Circular Interpolation on Axes Other Than X, Y, and Z

Turbo PMAC firmware supports circular interpolation directly only using the X, Y, and Z axes of a coordinate system. There are occasions when users desire the capability on other axes as well.

In this example, we have a system with X, Y, U, and V axes. The U-axis is parallel to the X-axis; the V-axis is parallel to the Y-axis. We desire the ability to do XY circular interpolation (which is a standard capability), but also XV, UY, and UV. We write the example in G-code (RS-274) style, because of its automatic use of subroutines in Turbo PMAC, but there are other ways to do this as well. We use the standard G17 code to specify XY circles, and invent G17.1 for XV, G17.2 for UY, and G17.3 for UV.

The example involves a double switch. If we want to be able to execute circles using the U and V axes, we must assign their motors to the X and Y axes for this period. But the motion program commands them as U and V axes, so we must then trade the U and V command values with the X and Y command values, which we do in a subroutine.

Substitutions and Definitions

```
#define Mtr1AxisCode  M158          ; #1 code for axis assignment
Mtr1AxisCode->Y:$0000C0,16,4
#define Mtr2AxisCode  M258          ; #2 code for axis assignment
Mtr1AxisCode->Y:$000140,16,4
#define Mtr3AxisCode  M358          ; #3 code for axis assignment
Mtr1AxisCode->Y:$0001C0,16,4
#define Mtr4AxisCode  M458          ; #4 code for axis assignment
Mtr1AxisCode->Y:$000240,16,4
#define UVW  4                      ; Code for assigned to UVW
#define XYZ  7                      ; Code for assigned to XYZ
#define Farg  Q106                  ; Variable F argument read into
#define Iarg  Q109                  ; Variable I argument read into
#define Jarg  Q110                  ; Variable J argument read into
#define Uarg  Q121                  ; Variable U argument read into
#define Varg  Q122                  ; Variable V argument read into
#define Xarg  Q124                  ; Variable X argument read into
#define Yarg  Q125                  ; Variable Y argument read into
#define BitsPassed  Q100
#define Fval  32                    ; Bit value for F passed
#define Ival  256                   ; Bit value for I passed
#define Jval  512                   ; Bit value for J passed
#define Uval  1048576               ; Bit value for U passed
#define Vval  2097152               ; Bit value for V passed
#define Xval  8388608               ; Bit value for X passed
#define Yval  16777216              ; Bit value for Y passed
; N0                                ; G00 (Rapid) subroutine
RAPID
PRELUDE G17.5                       ; Automatic subroutine call
RETURN
N1000                               ; G01 (Linear) subroutine
LINEAR
PRELUDE G17.5                       ; Automatic subroutine call
RETURN
N2000                               ; G02 (CW Circle) subroutine
CIRCLE1                             ; CW circle mode
PRELUDE G17.5                       ; Automatic subroutine call
RETURN
N3000                               ; G03 (CCW Circle) subroutine
CIRCLE1                             ; CCW circle mode
PRELUDE G17.5                       ; Automatic subroutine call
```

```
RETURN
N17000                              ; G17 subroutine
NORMAL K-1                          ; XY plane in Turbo PMAC
DWELL 0                             ; Stop lookahead and blending
AxisCode=0                          ; Use X and Y for circles
Mtr1AxisCode=XYZ                    ; #1 assigned to X
Mtr2AxisCode=XYZ                    ; #2 assigned to Y
Mtr3AxisCode=UVW                    ; #3 assigned to U
Mtr4AxisCode=UVW                    ; #4 assigned to V
CMD"&1PMATCH"                       ; Re-align coordinate system
DWELL 5                             ; Give time for PMATCH cmd
RETURN
N17100                              ; G17.1 subroutine
DWELL 0                             ; Stop lookahead and blending
NORMAL K-1                          ; XY plane in Turbo PMAC
AxisCode=1                          ; Use X and V for circles
Mtr1AxisCode=XYZ                    ; #1 assigned to X
Mtr2AxisCode=UVW                    ; #2 assigned to V
Mtr3AxisCode=UVW                    ; #3 assigned to U
Mtr4AxisCode=XYZ                    ; #4 assigned to Y
CMD"&1PMATCH"                       ; Re-align coordinate system
DWELL 5                             ; Give time for PMATCH cmd
RETURN
N17200                              ; G17.2 subroutine
DWELL 0                             ; Stop lookahead and blending
NORMAL K-1                          ; XY plane in Turbo PMAC
AxisCode=2                          ; Use U and Y for circles
Mtr1AxisCode=UVW                    ; #1 assigned to U
Mtr2AxisCode=XYZ                    ; #2 assigned to Y
Mtr3AxisCode=XYZ                    ; #3 assigned to X
Mtr4AxisCode=UVW                    ; #4 assigned to V
CMD"&1PMATCH"                       ; Re-align coordinate system
DWELL 5                             ; Give time for PMATCH cmd
RETURN
N17300                              ; G17.3 subroutine
DWELL 0                             ; Stop lookahead and blending
NORMAL K-1                          ; XY plane in Turbo PMAC
AxisCode=3                          ; Use U and V for circles
Mtr1AxisCode=UVW                    ; #1 assigned to U
Mtr2AxisCode=UVW                    ; #2 assigned to V
Mtr3AxisCode=XYZ                    ; #3 assigned to X
Mtr4AxisCode=XYZ                    ; #4 assigned to Y
CMD"&1PMATCH"                       ; Re-align coordinate system
DWELL 5                             ; Give time for PMATCH cmd
RETURN
N17500                              ; Move-processing subroutine
READ(X,Y,U,V,I,J,F)
IF (BitsPassed&XVal>0)              ; X passed?
  IF (AxisCode=0 OR AxisCode=1)     ; X not switched
    XCmd=XVal
  ELSE
    UCmd=XVal
  ENDIF
ELSE
  IF (AbsMode=0)                    ; Incremental mode
```