Sample Dancer Arm Application Using JOG Offset Register

While precautions have been taken in the preparation of this note, Parker Corporation and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

**********DISCLAIMER**********
This software program is provided free of charge and without warranty of any kind, either expressed or implied. In no event will PARKER HANNIFIN CORPORATION or PARKER COMPUMOTOR be liable for any damages, including but not restricted to lost profits, lost savings, or component failure arising out of the use or inability to use this software program. The sole purpose of this program is to demonstrate the functional application of the customers desired application. It is the responsibility of the user to insure that this program is not misused.
****

This is the one of the simplest implementations of a dancer arm application. The analog feedback device attached to the dancer arm is fed directly into the command position summation point, a.k.a. Primary Set Point (PSP) by writing the value into the JOG OFFSET register (shown in Figure 1 as the "Jog Profiler" register). The user's program has have to actively monitor the analog input and store the value into the JOG OFFSET register. Program processing time will be important to achieve a high bandwidth for this signal.

There is no need to tune the axis with respect to the dancer arm input, although the ADC GAIN could be used to adjust the amount of arm deflection vs. the amount of correction.

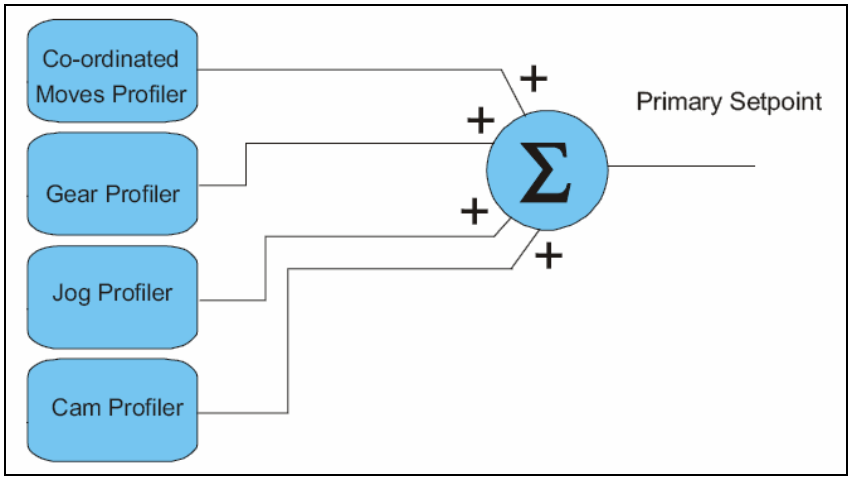

Figure 1 : Primary Set Point Summation

The dancer arm correction is input as a "position correction" command which has an immediate affect on the axis velocity. The ADC GAIN command can be used to invert the polarity of the command signal to allow the dancer arm input to either add or subtract from the axis' commanded position. Since this is a position correction, this would not be useful to correct for product stretch since it has a finite range. See sample code on next page.

Sample Dancer Arm Application Using JOG Offset Register

Sample program:

```
PROGRAM

REM Setup ADC0 so +10V to -10V range is -4096 to +4096 counts
ADC ON
ADC MAX 1
ADC 0 SCALE 10
ADC 0 GAIN 4000
ADC 0 POS 0

REM setup our axis
RES X
PPU X4000

REM This program will jog axis X with ENC2 as the clock/count source for the
motion

REM Set the jogging count source for X to use ADC0

REM  The dancer arm continuously corrects position by injecting a jog offset
REM into the commanded position of the axis.

REM  Loop until input 24 is activated
WHILE (NOT BIT 24)
        REM Set the jog offset for X equal to value on ADC0
        P12297 =  P6408
WEND

ENDP
```

Sample Dancer Arm Application Using JOG Offset Register

Another sample with averaging the signal:

```
PROGRAM
REM set aside memory for arrays and loop counter
DIM LA1
DIM LA0(20)
DIM LV10

REM Setup ADC0 so +10V to -10V range is -4096 to +4096 counts
ADC ON
ADC MAX 1
ADC 0 SCALE 10
ADC 0 GAIN 4000
ADC 0 POS 0

REM setup our axis
RES X
PPU X4000

REM This program will jog axis X with ENC2 as the clock/count source for the
motion

REM PREPOPULATE ARRAY WITH CURRENT ANI READING
FOR LV0 = 0 TO 19 STEP 1
LA0(LV0) = P33288
LV1 = LV1 + LA0(LV0)
NEXT

REM  The dancer arm continuously corrects position by injecting a jog offset
REM into the commanded position of the axis.

REM  Loop until input 24 is activated
WHILE (NOT BIT 24)
FOR LV0 = 0 TO 19 STEP 1
        REM change one of the ADC samples at a time
        LV1 = LV1 - LA0(LV0)       : REM subtract old entry
        LA0(LV0) = P6408           : REM retrieve new entry
        LV1 = LV1 + LA0(LV0)       : REM add new entry
        LV2 = LV1 / 20             : REM average the data points
        PRINT LV2                  : REM must use LRUN to see print's
        P12297 =  LV2 : REM Set the jog offset for X equal to value on ADC0
NEXT
WEND

ENDP
```