Suspend/Resume
of Cams

Control Techniques Americas
12005 Technology Drive, Eden Prairie, MN 55344
Phone: 800-893-2321
Fax: 952-995-8020

Introduction

This application note shows how to take advantage of the suspend/resume feature of cams within PowerTools to perform the following common functions:

1. Determine where the Master axis should be based on the Follower's position
2. Determine where the Follower axis should be based on the Master's position
3. Correct use of cam registers and functions used with Resume
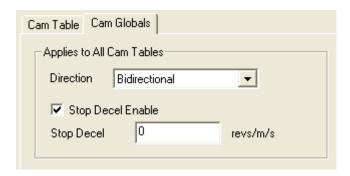4. Starting a cam from the middle of the cam table

Suspending a Cam

A cam in motion can be suspended by using the Cam.Suspend = True command in a program or by assigning the suspend to an input in the Assignments view. If using the Cam.Suspend = True in a program. The program must set Cam.Suspend = False before resuming the cam.

When Cam.Suspend is active, the cam motion will decelerate to a stop. The deceleration rate is set by the Stop Decel value set in the Cam Globals tab of the cam view:

Once suspended, the cam can be resumed at any time as long as the Cam.ResumeAvailable status bit is on.

After suspend is complete, the user can index, jog or perform any other motion. It is advised to move the cam follower to the desired location before resuming the cam.

Resuming a Cam

Mechanical cams and their cam follower never loose synchronization with each other, unless a component breaks or wears out. Electronic cams do not have the mechanical linking components and therefore can be out of sync with each other. Emergency stops and drive faults can lead to the master and follower loosing synchronization. In order to re-sync the master and its follower a resume procedure must be used.

Two cases exist for resuming a cam:

1. Machines where the master axis never stops and the cam must be started with the master in motion. This case can simply resume the follower motion when the master passes a particular position. This case also covers machines that are very tolerant to an out-of sync condition upon starting and stopping.

2. Machines where the master and follower axes cannot be out of synchronization when the cam master and follower are started or resumed. This type of machine must start the motion with both axes at rest located on the path defined by the cam table. There are two methods to re-sync the cam prior to initiating motion:
    a. Move the master to match the follower position according the path defined by the cam table
    b. Move the follower to match the master position according the path defined by the cam table

Case 1

As the master is moving through its cycle, the follower motion can start at some particular point of the master, at that point the follower ramps up to speed to match the master. A Cam.Resume command can be used to implement this procedure. Even though the cam may never have been 'suspended' prior to the Resume.

This procedure can be implemented in the following user program code:

```
Cam.0.Initiate
Wait for Cam.CommandingMotion

Do While (!Cam.CommandComplete or Cam.ResumeAvailable)

    If  DriveInput.2 Then
        Cam.Suspend
        Wait For DriveInput.2   = OFF
    Endif


    If DriveInput.3 and Cam.ResumeAvailable Then
        Wait For DriveInput.3 = OFF

        'Initiate a Capture event
        Capture.0.CaptureEnable = ON
        Capture.0.CaptureClear
        Capture.0.CaptureActivate = ON
        WAIT FOR CONTROL LOOP
        Capture.0.CaptureActivate = OFF

        'Calculate the internal Master Resume value
        var.var3 = 90
        If(Capture.0.CapturedMasterPosHomed > var.var3) then
            Capture.0.CapturedMasterPosition = Capture.0.CapturedMasterPosition + 360 + var.var3 - Capture.0.CapturedMasterPosHomed
        Else
            Capture.0.CapturedMasterPosition = Capture.0.CapturedMasterPosition + var.var3 - Capture.0.CapturedMasterPosHomed
        Endif

        'Wait for Master to pass and Resume the cam motion
        Wait For RealTimeProgram.CapturedMasterPosition > Capture.0.CapturedMasterPosition
        Cam.Resume using capture.0
    Endif

Loop
```

Code description

The cam is first initiated and the code drops into the Do While Loop

DriveInput.2 section
To start a cam in the middle of the cam table, which may be desirable with a moving master, we must issue a Cam.Suspend to allow the cam resume command to function.

DriveInput.3 section
After the Suspend is executed, a Capture event is initialized and activated.
Var.var3 is a user variable that holds the position of the master that the cam will be resumed at.
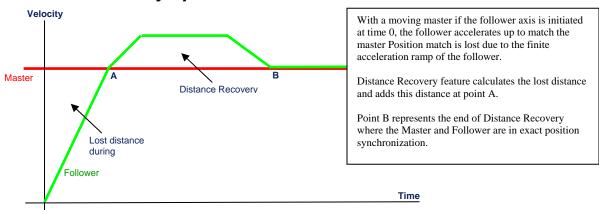
The If Then Else statement performs a calculation with several internal registers that have been loaded by the capture event and are used to calculate the required master position.

A Wait for statement is used to wait for the calculated master position to pass. Even though a Real Time program is not used, the register RealTimeProgram.CapturedMasterPosition must be used.
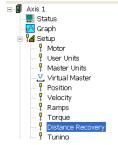
The cam is resumed with a using Capture.0 motion modifier; this forces the drive to exactly sync its motion to the moving master.

Distance Recovery option

With a moving master if the follower axis is initiated at time 0, the follower accelerates up to match the master Position match is lost due to the finite acceleration ramp of the follower.

Distance Recovery feature calculates the lost distance and adds this distance at point A.

Point B represents the end of Distance Recovery where the Master and Follower are in exact position synchronization.

Because the master is in motion and follower must follow a finite acceleration ramp, the master and follower will not be position synchronized on the exact cam path. Using the Distance Recovery feature will cause the drive to perform a small distance (or position) recovery and essentially cause the follower to speed up momentarily until it catches the master. After the Distance Recovery is performed (point B), the master and follower will be exactly on the cam path.

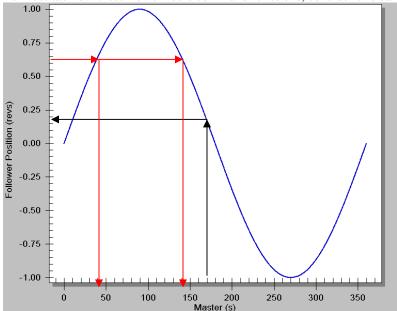Distance Recovery setup can be found in the hierarchy tree under Setup:

Case 2

Machines that cannot tolerate an out-of-sync condition need to have the master or follower moved onto the cam path. PowerTools has two commands that assist in determining where to move the master or the follower. This case implies that the master and slave are stationary before starting motion.

To illustrate how these PowerTools command functions, consider this cam profile:

For every Master position there is one distinct follower position. For example, to determine where the follower should be when the master is at 170 (black arrows), call the function SetCamMasterOffset(Table#, 170) and the result will be the follower's position as it would be interpolated through the cam table (~0.190).

Determining the master position poses a new issue, the master can have multiple positions for any given follower position. To determine where the Master should be when the Follower is at 0.625 (red arrows), call the function SetCamFollowerOffset(Table#, 0.625) and the result will be the <u>first</u> master position as it would be interpolated through the cam table (~40). Call the function again and it will return the next master position (~140). This function will also calculate through all cam tables that have a forward chain value set from the original Table# in the function.

Several other control parameters exist within PowerTools to help a user program properly handle any suspend/resume condition. See the manual Help sections below:

7

SetCamMasterOffset(Table#,MasterPosn)

This program command calculates the follower cam position that corresponds to the MasterPosn argument interpolated through the given cam table chain and stores the calculated value in Cam.ResumeFollowerPosn register.

This value can be very useful when performing a Cam.Resume when the master and follower must both lie on the path defined by the cam table. This function causes no physical movement of the follower axis. The follower can be moved (by index, jog, etc) to this position prior to performing a Cam.Resume.

For Realtime cams, it sets the Cam.ResumeMasterTime with the MasterPosn argument, for synchronized cams it sets the Cam.ResumeMasterPosn with the MasterPosn arguments.

Cam.ResumeAvailable will be active if this command successfully executes.

Repeated calls will return the next master position found within the given table. If no value is found or repeated calls produce no more valid follower positions, the Cam.ActiveTable will be set to -1.

Repeated calls of this function will cause the function to follow the cam table forward chaining path searching for a corresponding MasterPosn. The Cam.ActivePoint and Cam.ActiveTable will be set with the cam table where the MasterPosn was found. Cam.ActivePoint is a relative value within the Cam.ActiveTable.

If no MasterPosn value is found, the Cam.ActiveTable will be set to -1.

SetCamFollowerOffset(Table#,FollowerPosn)

This program command calculates the master position that corresponds to the FollowerPosn argument interpolated through the given cam Table# and stores the calculated value in Cam.ResumeMasterPosn or Cam.ResumeMasterTime register. This value can be very useful when performing a Cam.Resume when the master and follower must both lie on the path defined by the cam table. This function causes no physical movement of the master or follower axis. Once the master resume position is found from this function, the master axis can be moved (by index, jog, etc) to this position prior to performing a Cam.Resume.

For Realtime cams, it sets the Cam.ResumeMasterTime with the interpolated Master Position. For Synchronized cams, it sets the Cam.ResumeMasterPosn with the interpolated Master Position.

Cam.ResumeAvailable will be active if this command successfully executes.

Repeated calls will return the next master position found within the given table. If no value is found or repeated call produce no more valid master positions, the Cam.ActiveTable will be set to -1.

The following user program code shows a simple method to re-sync the follower axis to the master axis

```
Cam.0.Initiate

Do While  True

    If  DriveEnableStatus = off Then     'if Estop occurs
        Wait For Fault.Faulted =off      'wait here until drive fault is reset

        Wait For DriveOK and DriveEnableStatus 'wait here until drive is enables and ready

        ' Now perform a capture, contains a value that has equivelent values for master
        ' feedback and master homed feedback positions, to be used with the Resume later on
        Capture.0.CaptureClear
        Capture.0.CaptureActivate = On
        wait for time 0.1
        Capture.0.CaptureActivate = Off


        SetCamMasterOffset(0, Capture.0.CapturedMasterPosHomed) 'Calculate Follower Position
        Index.1.Dist = Cam.ResumeFollowerPosn  'move follower to the correct position
        Index.CamRecovery.Initiate  'CamRecovery,Rotary Plus,Dist=0.0000 revs,Vel=300 revs/m
        Wait For Index.AnyCommandComplete

        'resume the cam at master  (Actual equivelent in absolute master position)
        Cam.Resume using capture.0

        'master can now be started

    Endif
loop
```

Code description

Cam.0.Initiate
Cam is started, at the master is running. At some point in the future the machine is estopped or a drive fault has occurred (possible a product jam that has caused the follower axis to have a following error fault.

The Drive.Enable.Status is detected by the IF statement, after the power is re-applied to the axis and all faults cleared (not show in the code is any input from an operator panel or push button)

A Capture is performed, this performs several internal calculations, but mainly is used for the Cam.Resume using capture.0 command

9

The SetCamMasterOffset is called with the cam table number and the master position, this will calculate the exact position the follower should be at.

The follower position is loaded into an index position and the index is initiated. The follower is now positioned exactly on the cam path at the corresponding position to the master.

The cam is resumed using Capture.0, this forces the follower's cam motion to exactly match the cam table's location.

Starting a cam in the middle of a cam table

Starting a cam in the middle of the cam table, requires the Cam.Resume command. Before a Cam.Resume can occur, the Cam.ResumeAvialable status bit must be on. This bit can be turned on by calling the SetCamMasterOffset or SetCamFollowerOffset function.

As shown in this code example:

```
'to resume a cam or start a cam from a row other than the starting row
'call this function with the arguments of (table number, Master Position)
'the Master position does not need to correspond to an actual row value in the cam table,
'the function will interpolate between row values.
SetCamMasterOffset(1,1.000)

'Once this function runs, the incremental follower value from the starting row's follower value
'will be stored in the Cam.ResumeFollowerPosn register
'If the function call is succesful, the Cam.Resume bit must go high


'If the incremental distance from the cam's starting point is desired:
index.0.dist = Cam.ResumeFollowerPosn
'Or
'If the Absolute distance including the starting location is desired:
index.0.dist = Cam.ResumeFollowerPosn + Cam[1,0].Follower


'Initiate the Absolute index to the Cam.ResumeFollower value
Index.0.Initiate     'Index0,Absolute,Dist=0.0000 revs,Vel=999 revs/m
Wait For Index.AnyCommandComplete


'Once Index.0 is initiated, the follower will move to a position that corresponds to the
'master's position in the cam table.

'The Cam.Resume bit must be on before issuing a Cam.Resume
If Cam.ResumeAvailable Then
    Cam.Resume
Else
    'cam cannot be resumed
    'check program code and/or master Offset value
EndIf
```