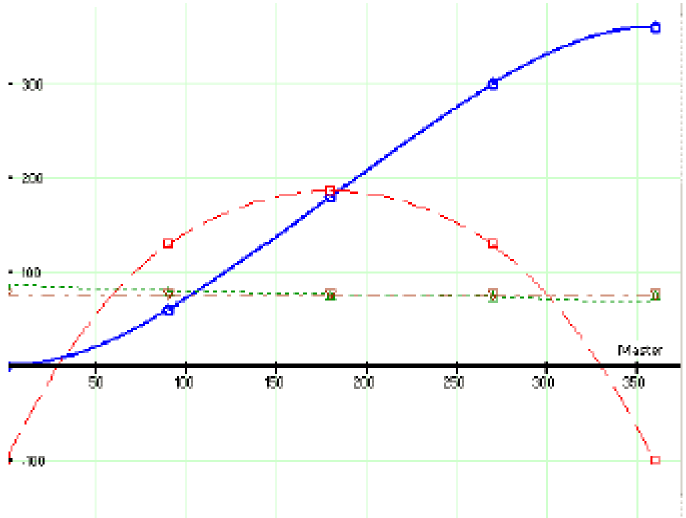
Implementing Cam Recovery on Logix-Based Controls

Purpose of this Document: This document is intended to instruct program developers in the procedures required to recover a CAM from an E-Stop condition without having to re-home the axes.

Application Description: This is a general-purpose application note which can be used across a wide range of applications where machines with multiple synchronized cams must be restarted without returning the machine to the home or start position.



When a multi-cam machine stops: For this illustration, lets' use a packaging machine that involves two axes of control. Let's assume that cam #1 has stopped at 180 degrees, and cam #2 has stopped at 270 degrees. At this stage of machine operation, these are the correct relational positions for each cam. The desired goal is for the machine to restart exactly where it stopped, without disturbing the flow of product through the machine, or damaging product that is still in the machine. This mandates a restart routine that does not require homing either of the cams to their starting positions.

Control System Description: Consists of:

a. HMI Supervisory Workstation
b. Network from controller to field devices
c. ControlLogix Version 11.11

CONTROL CHALLENGES:

Most applications will not allow a system or machine to re-home or go to the starting position without an operator having to clear out product from the machine to start up the system. This can prove costly and time consuming every time an e-stop condition occurs. When cammed to a master axis, The CAM points are not absolute so when stopping in the middle of a CAM the slave axis does not know where it should be located relative to the master.

CONTROL SOLUTIONS:

This document will show two CAM recovery solutions. One uses a Virtual axis for every Real Axis. The next solution creates a simulation using a virtual recovery master and slave. This paper will expound on each scenario and which is best for your situation.

Understanding Cam recovery scenarios

A **cam** consists of multiple **velocities (V1, V2)** assigned to a motor at different positions of a master drive.
A **CAM Table** is made up of the correlating **positions (P1 … PN)** of the slave in regard to the master

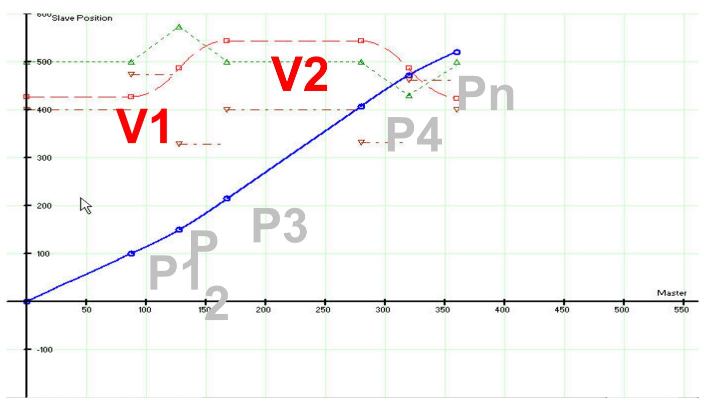


Figure 1

To interconnect between different velocities different methods of interpolation can be employed to "smoothly" transition between different speeds

- Sin
- Cubic spline (3rd order)
- 5th order polynomial
- …
-

ControlLogix uses cubic spline as the native interpolation method

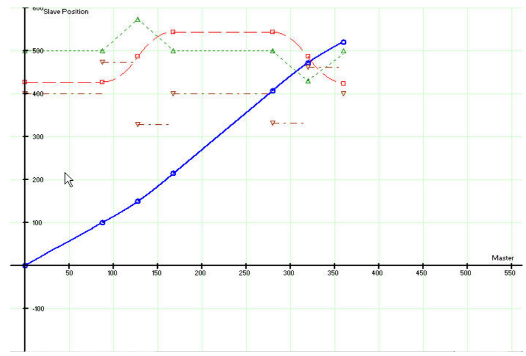


Figure 2

There are multiple scenarios where a slave requires dynamic synchronization to a master

- The axis has physically faulted and the master has moved onwards
- There is no fixed position where master and slave can be started from together (e.g. mechanical alignment of tools)
- There is a third axis interlocked with the slave that prohibits the slave from moving on its own.
- Product has to be removed from the machine.

What is required to move the slave to a corresponding CAM "position" without moving the master?

<u>Within linear segments</u> the position for the slave to move to is simply the linear interpolation between Cam points. The math is fairly simple

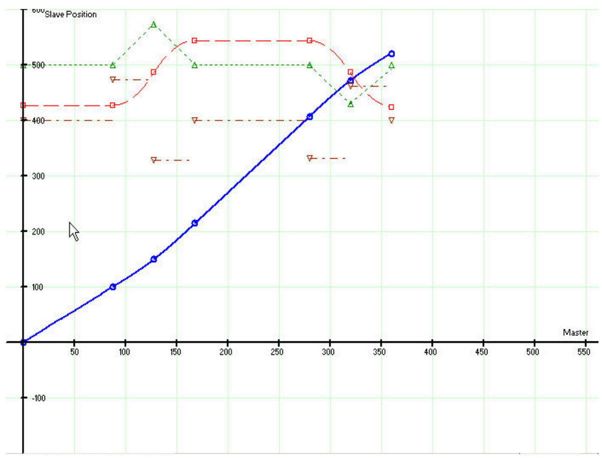


Figure 3

<u>Within cubic segments</u> the math required involves a minimum of 4 points (up to 6 points if adjacent segments are cubic as well) and the algorithm is considered proprietary. The math is **NOT** simple at all.

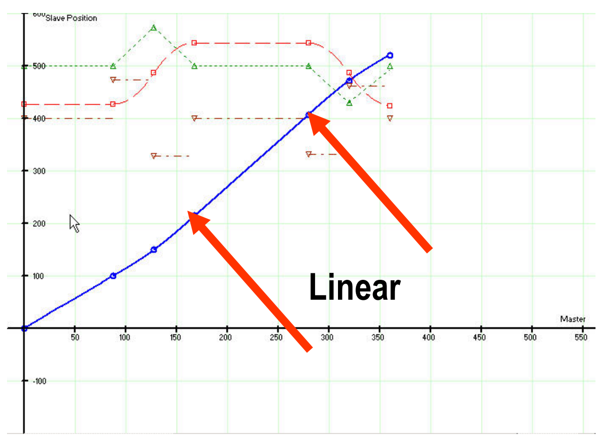


Figure 4

There are different possibilities to recover a faulted slave. In this case we will show two possibilities for cam recovery.

<u>#1 -A Virtual Axis for Each Real Cammed Axis</u>

Use a virtual axis (which is not faulted by an E-Stop) and gear a real axis to it

- Recover by aligning the physical axis to the virtual
- Requires two axis objects (one real and one virtual axis) for each axis that needs synchronization
- Easy to implement!!

Does **<u>not</u>** work for dynamic homing scenario or third axis scenario

Simulated Run - Use two additional VIRTUAL axes for the entire application

a. Use one as a Virtual Recovery Master and as a Virtual Recovery Slave

b. Load in all cammed axes CAM tables into the RecoveryCAM table using recovery master and slave

c. Move the Recovery Master to the Master Position

d. Look at the Recovery slave to see where it should be

d. Due to the fact that the simulated master and slave are actually running, the recovery will take little time (typically less than 100ms)

When to use which recovery?

- Use a Virtual recovery (easiest to implement) when:
  Each Servo obtains two axis objects, the physical being geared to a virtual that contains the application profile, if
  – The number of axes in the application is limited (<8)
  – Processor performance is not an issue
  – There is no dynamic homing (Slave is started initially at a fixed master position)
- Use a Simulated Run recovery (most powerful) when:
  Two virtual Axes are used to calculate Slave values for all CAMs in use, if
  – RAM is an issue
  – Performance is an issue
  – High accuracy is required
  – Axes object count is an issue (>8, < 32 per processor)

CAM Recovery Using ControlLogix

Note: ControlLogix has a cam editor shown in Figure 5. The CAM points are in relative coordinates thus making it difficult to recover when in the middle of a CAM, if the machine is stopped in an irregular manner such as an E-Stop condition.

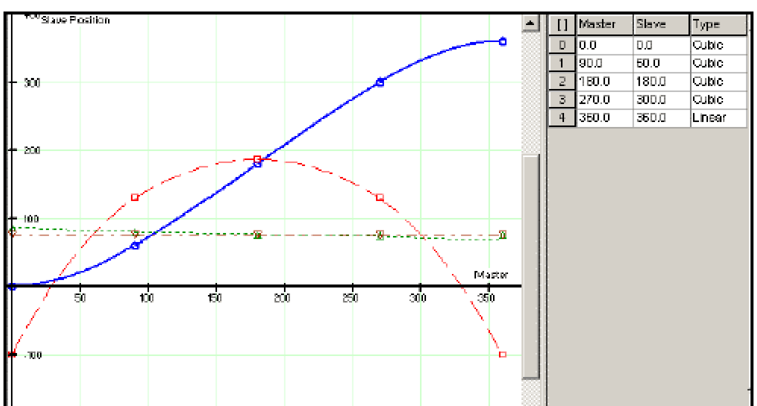


Figure 5

Example: In the figure shown the master CAM points are as follows.

[ ]	Master	Slave	Type
0	0.0	0.0	Cubic
1	90.0	60.0	Cubic
2	180.0	180.0	Cubic
3	270.0	300.0	Cubic
4	360.0	360.0	Linear

Figure 5a

If the machine were to stop at any time between any user input points, the master to slave reference is lost. An example of this would be the master stops at 110 Degrees. Do to the Cubic spline functionality and few points needed, it would be difficult to determine where the slave should be relevant to the master.

ControlLogix CAM Recovery

The next few pages will show an example of two different ways to recover from a CAM. As described previously this solution will use virtual axes for each real axis to recover. The second will use a simulated run of a virtual recovery master and virtual recovery slave. You can determine which is best for you from the criteria shown previously.

CAM Recovery Using Virtual Axes

- The next few pages will explain how to:
- Create All Axes Needed for CAM Recovery
- Link Virtual Axes to the Master Virtual
- Gear all Servo Axes to the Corresponding Virtual
- Move all Servo Axes to their Virtuals
- Starting the System
- Handle an E-Stop condition
- Recover from an E-Stop Condition

Step 1: Create All Axes Needed For Cam Recovery

In this example we have two real axes following a virtual master.
For your machine you will need to create a virtual for every axis which is being cammed to master whether it be real or virtual. This can be seen in Figure 6.

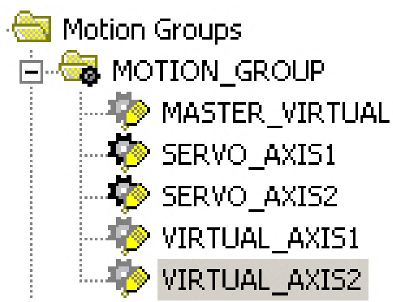


Figure 6

Remember: Every axis that is created will add overhead to the system whether it is real or virtual.

Note: Make sure to create virtuals only for axes that need cam recovery. This does not need to be done for every axis.

Step 2: Link Virtual Axes to the Master

All virtual axes created for CAM Recovery need to be linked to the master. This is done by creating a Motion Calculate Cam Profile (MCCP) and Motion Axis Position Cam (MAPC). This is shown in Figures 7 and 8.

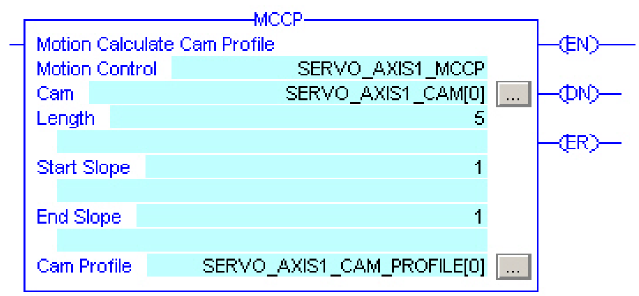


Figure 7

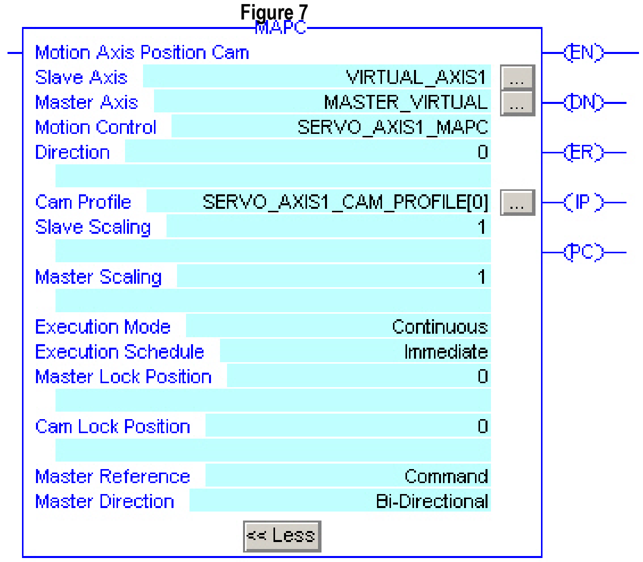
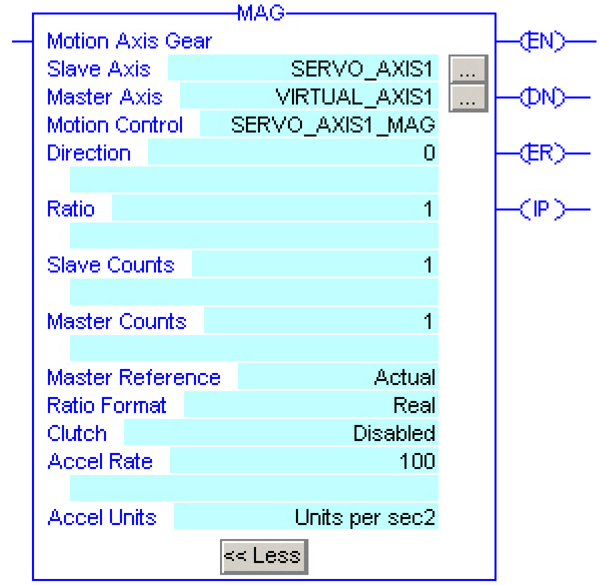


Figure 8

Important:
On power up or after a homing procedure, the virtual axes must be cammed to the virtual or real master. This will never have to be done again until the machine is powered down or the operator issues another homing procedure. This is will shown to you in later steps.

Note:
Notice we are linking the virtual axes to the master, not the Real Axes

Step 3: Gear all Servo Axes to their Corresponding Virtual
Once you have Cammed the Virtuals to the Master Gear all Servo axes to each Virtual. Once this is complete all axes are linked together.



Step 4: Move all Servo Axes to their Virtuals
Once the entire drive train has been created, make sure all servo axes are moved to their irtual's actual position. This is shown in figure 10. Be careful when selecting the move type for the real axis. I.E. the Move Type is two, this means it will move shortest.

Move Type:
0 = Absolute
2 = Rotary Shortest (It will travel the Shortest Path to the Position in Question)
3 = Positive Rotation
4 = Negative Rotation

Note:
The move type is application specific

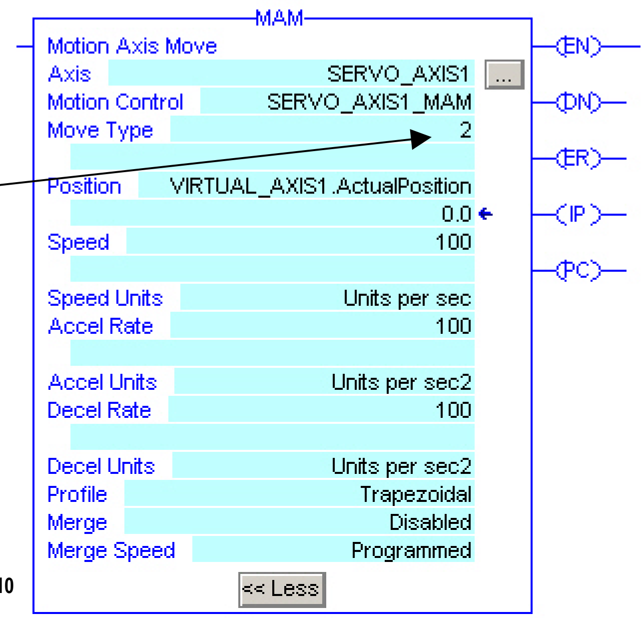


Figure 10

Step 5: Start the System
Now that all axes are linked and ready to go the next step is to start the system jogging or moving the virtual master.

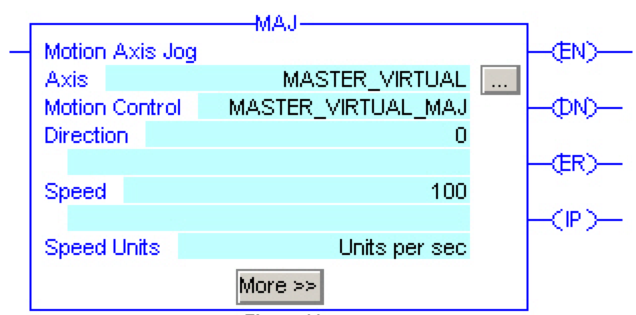


Figure 11

Step 6: Handle an E-Stop Condition
Once a guard door is open, E-stop is pushed, Etc, the system must be stopped at once. The Virtual master is ramped down but under control using an MAS instruction shown in Figure 12.

Note:
The Stop Type is **NOT** all. If the Master is doing a move then the Stop type is (MOVE). If the master is jogging then the Stop type is (JOG). This will keep the other virtuals cammed to the master. This is how the system drive train will stay intact.

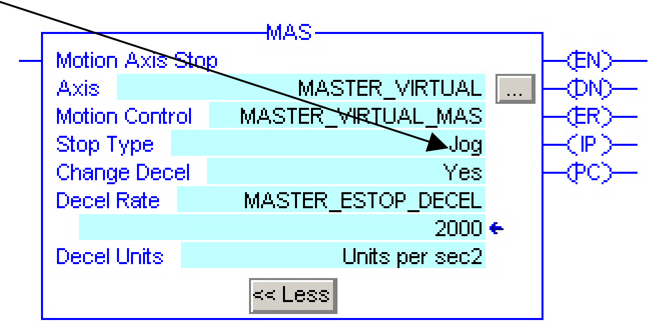


Figure 12

Step 7: Recover from an E-Stop Condition
When Recovering from an E-stop condition Repeat steps 3-5. Remember, that once the virtuals are cammed to the master this does not have to be done again unless the system is powered down or the cams are manually taken out of process.

CAM Recovery Using Simulation of Master and Slave Virtual Recovery Axes
The next few pages will explain how to:

- Create All Axes Needed for CAM Recovery
- Create User Defined Data Types (UDT) for recovery array
- Add this UDT to the Tag Database
- Create a Recovery Routine using the UDT Array
- Create a Simulation Recovery Routine
- Move the Servo Axes to their CAM Lock positions
- Handle an E-Stop condition
- Restart System

Step 1: Create All Axes Needed For Cam Recovery
In this example we have two real axes following a virtual master.
For your machine you will need to create a virtual for every axis which is being cammed to master whether it be real or virtual. This can be seen on in Figure 13.

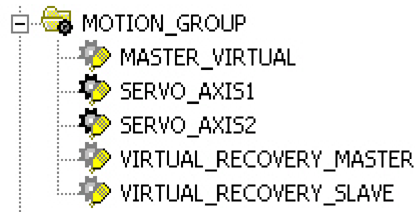


Figure 13

Step 2: Create User Defined Data Types (UDT) for recovery array
Create the Following Tag in the project window under the "User-Defined" folder. The tag members should consist of the following:

AXIS_POSITION CAM
CAM_PROFILE LENGTH
START_SLOPE END_SLOPE

For the CAM and CAM_PROFILE, these need to be created as an array. These arrays need to be created to fit your largest number of cam points.

For Example: If Axis one has 6 cam points and Axis two has 10 Then the array in the UDT needs to be of size 10.

Members:

Name	Data Type
AXIS_POSITION	REAL
CAM	CAM[10]
CAM_PROFILE	CAM_PROFILE[10]
LENGTH	DINT
START_SLOPE	DINT
END_SLOPE	DINT
*	

Figure 14

- Create a tag in the tag database and call it RECOV.
- Make the Type RECOVER.
- Create an array of one plus the total number of real axes.

In this case everything is set to ten.

RECOV	{...}	RECOVER[10]
RECOV[0]	{...}	RECOVER
RECOV[1]	{...}	RECOVER
RECOV[2]	{...}	RECOVER
RECOV[3]	{...}	RECOVER
RECOV[3].AXIS_POSITION	0.0	REAL
RECOV[3].CAM	{...}	CAM[10]
RECOV[3].CAM_PROFILE	{...}	CAM_PROFILE[10]
RECOV[3].LENGTH	0	DINT
RECOV[3].START_SLOPE	0	DINT
RECOV[3].END_SLOPE	0	DINT

Figure 15

Step 4: Create A Recovery Routine Using the UDT Array

When creating a “Recovery Routine” faults are cleared, feedback is turned on, and they are moved to their start position. This section will just show how to set up the MCCP and MAPC instructions using the User Defined Data Types created.

First step is to create the MCCP instruction for the first axis in question. In this case our axis is called SERVO_AXIS1. Use array one for the first axis.
Note: Array zero is not used.

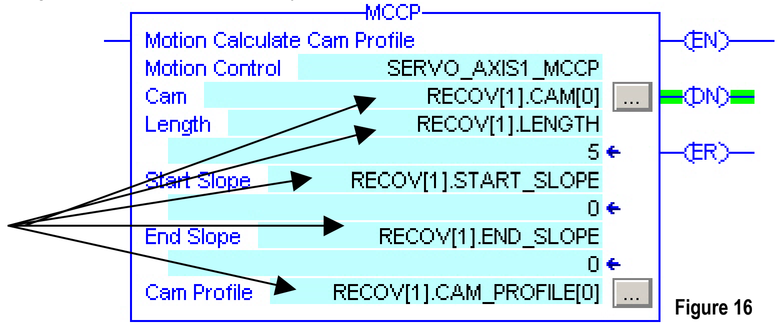


Figure 16

Second step is to create the MAPC. The Cam Profile is the same as the MCCP instruction. If the axis is locked on at all times make sure to have the Execution Mode is set to Continuous. If the axis is not locked on at all times make sure to set it to Once. In this case the CAM is always locked to the master and is Continuous. The execution schedule is immediate. Make sure the Master and Cam Lock Positions are the master actual position. This will tell the MAPC to start the CAM at the master position and not at the beginning.

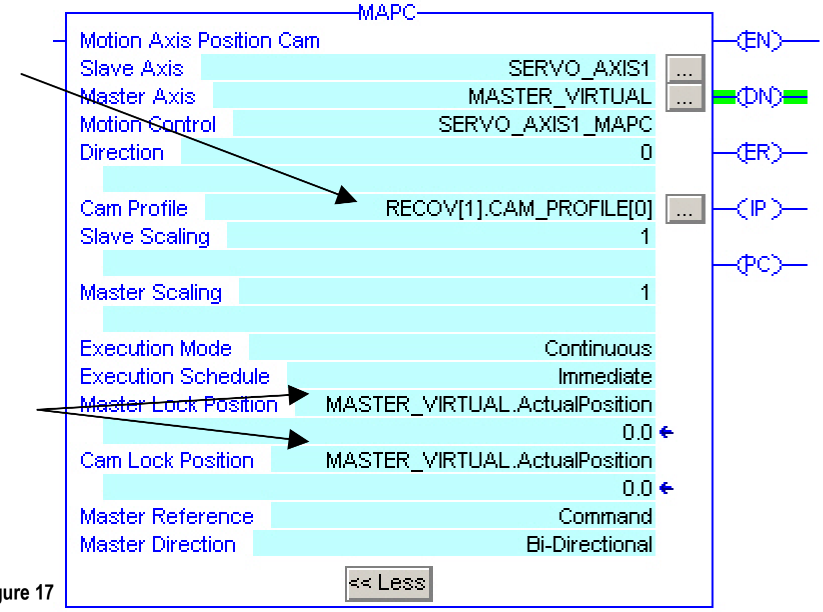


Figure 17

Note: Start the second axis with the next RECOV array element as shown. Increment this array manually for all MCCP and MAPC instructions until all axes have been calculated.

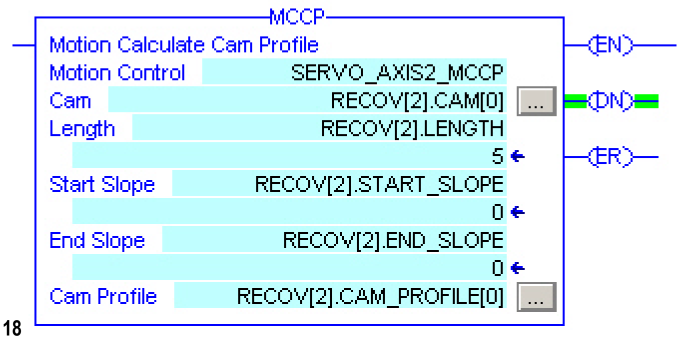


Figure 18

Step 5: Create a Simulation Recovery Routine

The next few steps will demonstrate how to simulate each axis by getting its position according to the master position.

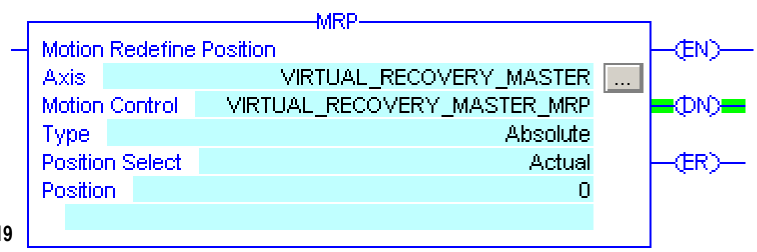


Figure 19

- Redefine the Master and Slave Recovery axis to Zero**.** This is done to start both the master and slave recovery axes to their start position.

- Create a Counter. The preset should be set to the total number of axes needing CAM recovery. In this case it is called RECOVER_CTU. Count up after redefining both axes.

- Create a double integer. This will be your alias tag to step through the simulation routine. In this case it is called NUMBER.

- Move the RECOVER_CTU.ACC into NUMBER

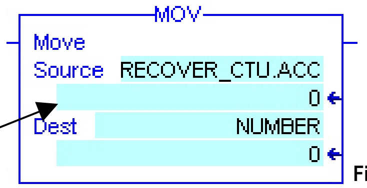


Figure 20

- Create a CAM and CAM PROFILE. The array length should be the length of the longest CAM. In this case they have been called RECOVERY_CAM and RECOVERY_CAM_PROFILE.

- Copy the RECOV CAM element into the RECOVERY_CAM tag as shown. Use the RECOV LENGTH element to tell the copy instruction how many CAM points to copy.

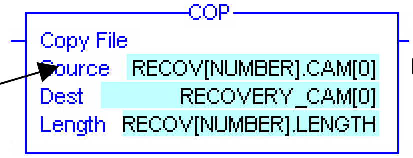


Figure 21

- Create the MCCP and MAPC instruction shown below

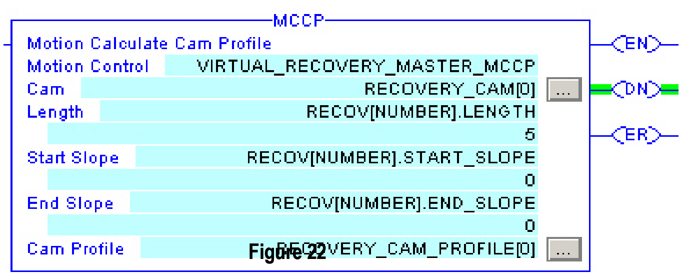


Figure 22

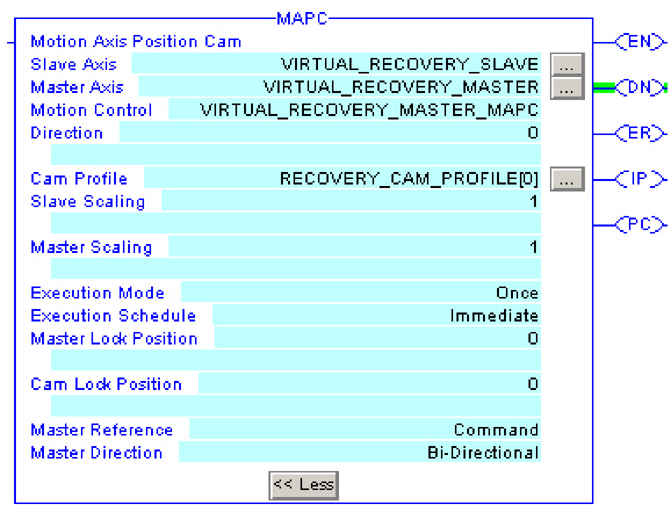
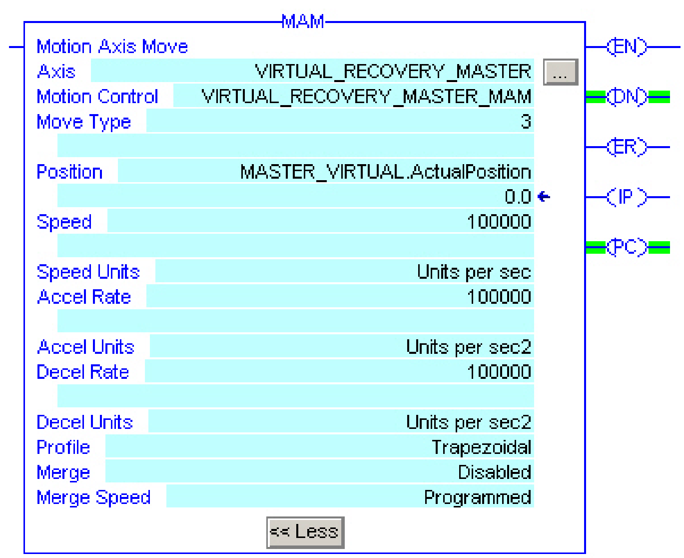


Figure 23

Move the Master Recovery Virtual to the Master Position when the MAPC is in process. Notice that the Acceleration, Deceleration, and Speed can be set very high. This is because we are using two virtuals to do the CAM recovery. Moving the Recovery Master to the Master Actual Position will let us arrive at the position where the slave should be according to the master. The is because the master and slave are still linked in the MAPC.

- Once the MAM instruction is complete, grab the slave recovery axis position and move it into the RECOV array for that particular axis. This is done by using the NUMBER alias.

Figure 24



- The next step is to stop the Virtual Recovery Slave. This will stop the MAPC.

- If the counter is not done, then start the sequence over again at the first step and increment the counter. Each time the counter is incremented it will look at the next axis until the sequence is entirely complete. Make sure the counter is set to the exact number of axes.

- Once the counter is complete, the simulation routine is done.

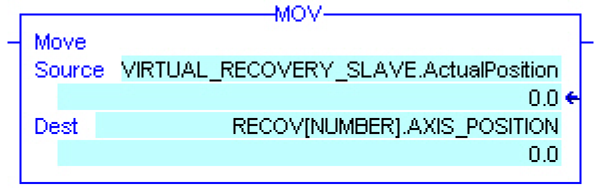


Figure 25

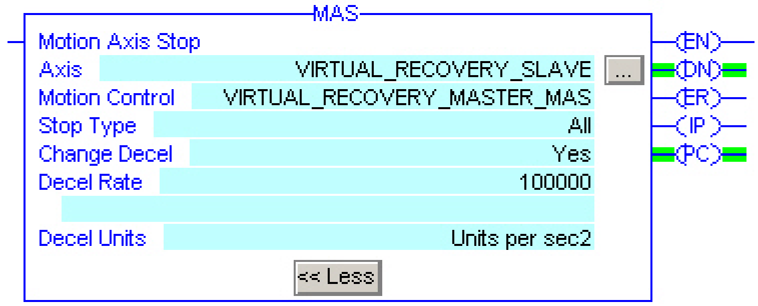


Figure 26

Step 6: Move the Servo Axes to their CAM Lock positions
This step is where the servo axis is moved to its position in accordance to the master.

In the simulation routine, The recovery slave position was moved into each axis position array. After recovery, each servo must be moved to that position. Make sure to use the right move type for your application. In this case we are assuming the axis can move backwards with type two.

Move Type:
0 = Absolute
2 = Rotary Shortest (It will travel the Shortest Path to the Position in Question)
3 = Positive Rotation
4 = Negative Rotation

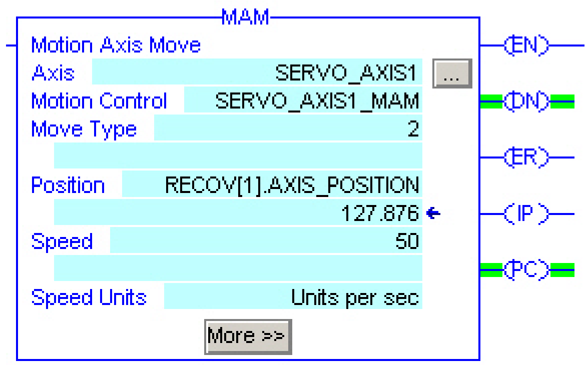


Figure 27

Step 7: Handle an E-Stop condition
When an E-Stop condition occurs, make sure to move a zero into the NUMBER tag and reset the RECOVERY_CTU. This needs to be done to insure the simulation routine will start at the beginning every time an E-Stop condition occurs. Also you want to make sure these things are done on power-up as well. Remember, there might be other data that needs to be reset every time an E-Stop condition occurs.

Step 7: Restart System
After the axes have been moved to their positions, we are now ready to start from the middle of any CAM and recover without moving the master and with minimal movement to each slave.

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (Publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://www.ab.com/manuals/gi) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this document.

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe/Middle East/Africa: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 6356-9077, Fax: (65) 6356-9011

Publication RA-AP004A-EN-P – August, 2003