

# Q\_MAM\_P V2.0

---

## Table of Contents

|                                    |    |
|------------------------------------|----|
| Overview .....                     | 2  |
| VERSION HISTORY .....              | 2  |
| EXECUTION .....                    | 3  |
| Q_MAM_P_V2 interface .....         | 5  |
| INSTRUCTION INPUTS .....           | 5  |
| INSTRUCTION OUTPUTS .....          | 7  |
| INSTRUCTION ERROR CODES .....      | 8  |
| Q_MAM_P: HOW IT WORKS .....        | 9  |
| Q_MAM_P: Sum Motion Profiles ..... | 12 |
| Q_MAM_P: CASCADE LINKS .....       | 13 |
| Q_MAM_P: FEEDFORWARD MODE .....    | 15 |

## Overview

This AOI targets selected Axis (in\_Axis) to its target position while avoiding collision with preceding Axis (in\_Axis\_Preceding). To compensate motion delay in queued movements this AOI allows two different strategies: Axis Preceding Current Position, or, alternatively, Axis Preceding Target Position

The target Position for the Mover is entered as absolute on the Trak.

**Queued:** If the way to the target is not free the Axis (Mover) queues until the way is free or the instruction is disabled

**Positive:** Only one direction possible

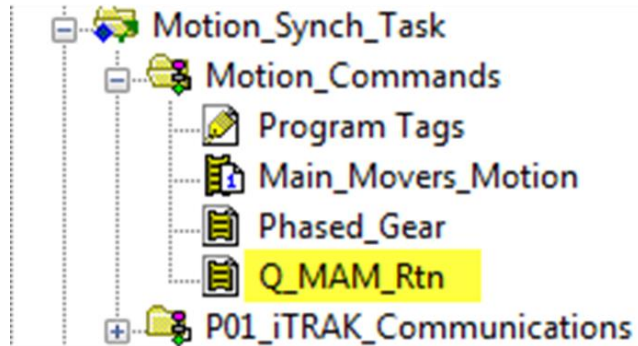
### Note that:

When enabled the Q\_MAM\_P instruction terminates all current movements assigned to this Axis. Subsequently the distance to be covered is turned into an incremental movement inside the instruction. **This allow others Motion Commands to be superimposed.**

## VERSION HISTORY

| Version | Edited By     | Date       | Additional Information / Changes  |
|---------|---------------|------------|---|
| V 2.0   | Tommaso Fochi | April 2021 | Include a second FeedForward Method<br>Use Interpolated Command Position to extrapolate Axis Position   |
| V 1.9   | Tommaso Fochi | 2020       | Added support for Jerk in % of Time.<br>Improved descriptions<br>When an Axis is a follower, it does not apply Jerk as it follows leader Axis |
| V 1.8   | Tommaso Fochi | 2019       | Added absolute positioning when possible, to solve rounding issues due to summing of incremental distances                                    |
| V1.7    |               |            | Changed Merged Mode on the MAM instructions inside the AOI  |
| V1.5    |               |            | Positive Movement only.<br>Motion instructions are all relatives.<br>Movement target can be updated during Motion                             |
|         |               |            |   |





**Very Important:** If the event task where Q\_MAM instructions are placed experiences task overlap the behavior of the Q\_MAM themselves is no longer reliable.

In case of collision between movers verify the task overlap count

The Q\_MAM Instruction manage its internal state also during the Enable\_In\_False state. This instruction only fits for LADDER logic

## Q\_MAM\_P\_V2 interface

Queued MAM Positive with Preceding Axis Position Feedforward

| Q_MAM_P_V2   |                                   |  |           |
|--|-----------------------------------|--|-----------|
| Queued MAM Positive with Preceding Axis Position Feedforward |                                   |  |           |
| Q_MAM_P_V2   | Queued_MAM[0]                     | Sts_Command_Position                     | 2293.1536 |
| xx                     | 0                                 | Out_FFMode_Active                        | 2         |
| in_Axis  | Mover_00                          | xx | 0         |
| in_Axis_Preceding  | Mover_09                          | ERR                                      | 0         |
| xx                     | 0                                 | EXERR                                    | 0         |
| in_rCmdPosition  | Q_Mam_pars[0].rTargetPos          |  |           |
|  | 0.0                               |  |           |
| in_rCmdVel   | Axis_Vel[0]                       |  |           |
|  | 3000.0                            |  |           |
| in_rCmdAcc   | Axis_Acc[0]                       |  |           |
|  | 35000.0                           |  |           |
| in_rCmdDec   | Axis_Dec[0]                       |  |           |
|  | 35000.0                           |  |           |
| in_dCmdAccJerk_PercentageOfTime                              | CmdAccJerk                        |  |           |
|  | 33                                |  |           |
| in_dCmdDecJerk_PercentageOfTime                              | CmdDecJerk                        |  |           |
|  | 33                                |  |           |
| xx                     | 0                                 |  |           |
| Cfg_Unwind   | ActualFormatData.Cfg.rTrackUnwind |  |           |
|  | 4000.0                            |  |           |
| Cfg_Axis_Min_Distance  | Axis_Cmd_Distance                 |  |           |
|  | 100.0                             |  |           |
| xx                     | 0                                 |  |           |
| in_MotionGroup   | Axes                              |  |           |
| xx                     | 0                                 |  |           |
| In_LinkedParametersInput                                     | Q_MAM_LinkedParameters[9]         |  |           |
| Out_LinkedParametersOutput                                   | Q_MAM_LinkedParameters[0]         |  |           |
| xx                     | 0                                 |  |           |
| In_EnablePositionFeedForward                                 | Enable_PositionFeedForward        |  |           |
|  | 1                                 |  |           |
| Cfg_Feedforward_Mode   | FeedForward_Mode                  |  |           |
|  | 0                                 |  |           |

## INSTRUCTION INPUTS

| INPUT DATA        |              |                               |
|-------------------|--------------|-------------------------------|
| NAME              | DATA         | DESCRIPTION                   |
| in_Axis           | AXIS_VIRTUAL | Axis to Move                  |
| in_Axis_Preceding | AXIS_VIRTUAL | Preceding Axis                |
| in_rCmdPosition   | REAL         | Q_MAM Target Position (units) |

|                                 |                            |   |
|---------------------------------|----------------------------|---|
| in_rCmdVel                      | REAL                       | Command Velocity<br>(units/sec)   |
| in_rCmdAcc                      | REAL                       | Command Acceleration<br>(units/sec^2)   |
| in_rCmdDec                      | REAL                       | Command Deceleration<br>(units/sec^2)   |
| in_dCmdAccJerk_PercentageOfTime | DINT                       | Acceleration Jerk<br>(expressed in % of Time)   |
| in_dCmdDecJerk_PercentageOfTime | DINT                       | Deceleration Jerk<br>(expressed in % of Time)   |
| Cfg_Unwind                      | REAL                       | Axis Unwind (Units)   |
| Cfg_Axis_Min_Distance           | REAL                       | Spacing between Axes<br>(Units)   |
| in_MotionGroup                  | MOTION_GROUP               | Project Motion Group  |
| In_LinkedParametersInput        | UDT_Q_MAM_LinkedParameters | Q_MAM Linked<br>Parameters. This input<br>comes from the preceding<br>axis Q_MAM instruction. It<br>includes state and value of<br>the selected feedforward<br>method |
| In_EnablePositionFeedForward    | BOOL                       | Enable Position<br>Compensation   |
| Cfg_Feedforward_Mode            | DINT                       | Select whether you want<br>to compensate Motion<br>delay using:<br>0 = Leader Axis Current<br>Position (PPI mode)<br>1 = Preceding Axis Target<br>Position (PTI mode) |

## INSTRUCTION OUTPUTS

| OUTPUT DATA               |                            |  |
|---------------------------|----------------------------|--|
| NAME                      | DATA                       | DESCRIPTION  |
| EN                        | BOOL                       | INSTRUCTION ENABLED  |
| IP                        | BOOL                       | INSTRUCTION IN PROCESS   |
| ER                        | BOOL                       | INSTRUCTION FAULTED  |
| PC                        | BOOL                       | INSTRUCTION COMPLETED  |
| Out_LinkedParametersInput | UDT_Q_MAM_LinkedParameters | Q_MAM Linked Parameters. This output goes to the Following axis Q_MAM instruction. It includes state and value of the selected feedforward method  |
| Sts_Command_Position      | REAL                       | Instruction Current Command Position   |
| Out_FFMode_Active         | DINT                       | Specify the current Feedforward Mode active<br>-1 = Instruction not Enabled<br>0 = No feedforward Enabled<br>1 = Current Position Feedforward Mode. Preceding Axis is not a Follower<br>2= Current Position Feedforward Mode. Preceding Axis is a Follower<br>3= Target Position Feedforward Mode. Preceding Axis does not have PTI enabled<br>4= Target Position Feedforward Mode. Preceding Axis has PTI enabled |
| Sts_MoveStatus            | BOOL                       | Axis in Move Condition   |
| ERR                       | DINT                       | Motion Instruction Error Code  |
| EXERR                     | DINT                       | Motion Instruction Extended Error Code   |

## INSTRUCTION ERROR CODES

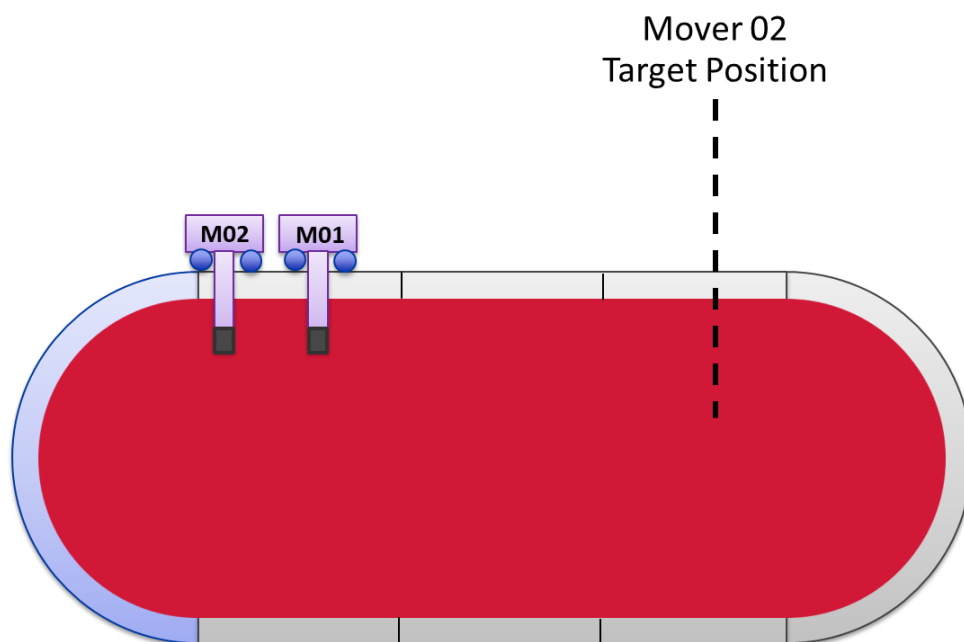
Instruction reports standard Motion Instructions Error Code and Extended Error Code

| ERR  | Description                         | Extended Information |
|------|-------------------------------------|----------------------|
| 1001 | Motion Group Not Synchronized       |                      |
| 1002 | Invalid Instruction Move Parameters |                      |
|      |                                     |                      |



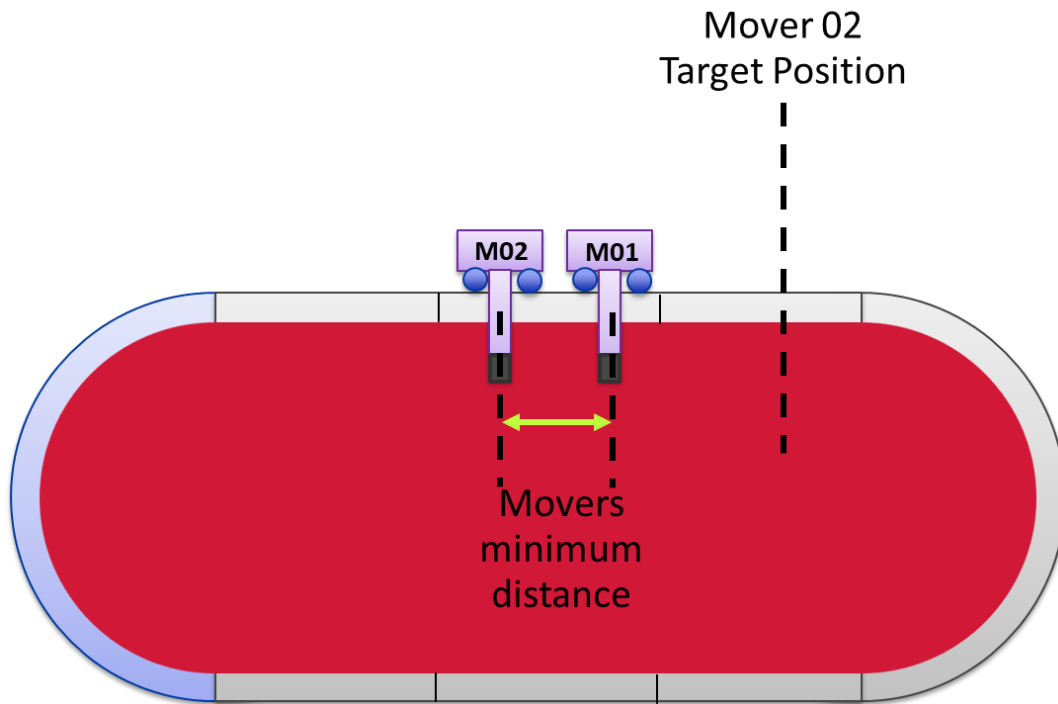
## Q\_MAM\_P: HOW IT WORKS

1 - Assign a Q\_MAM\_P command to selected Axis (in the example Mover 02): if Mover 01 does not move forward the Mover 02 will stand still

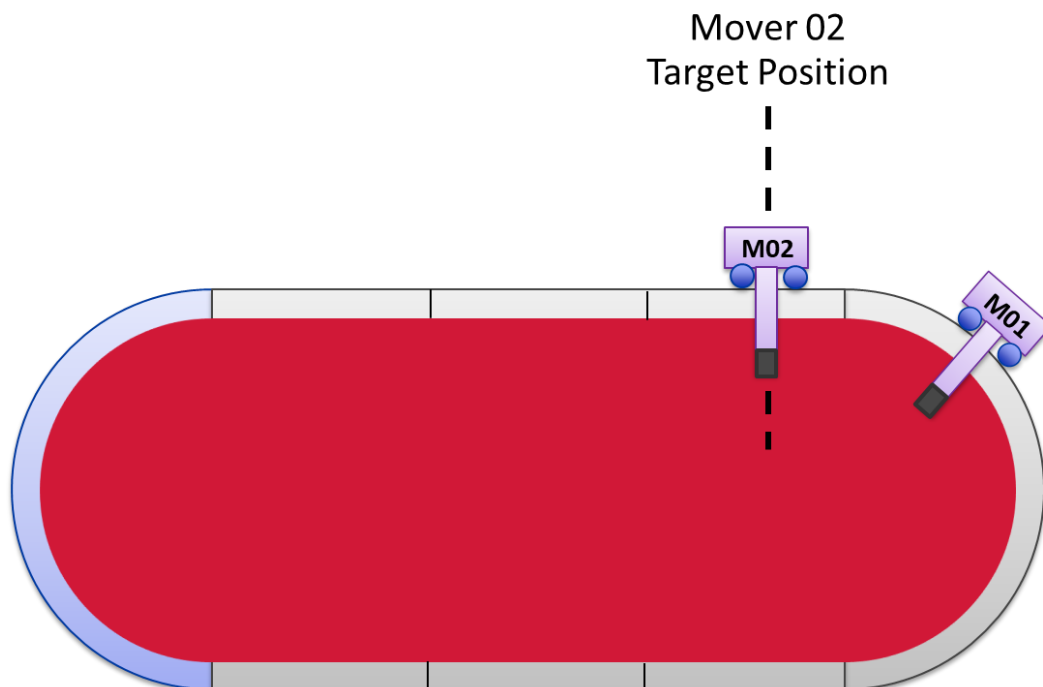


2 – If Mover 01 frees up some space, Mover 02 will proceed towards the Target Position.

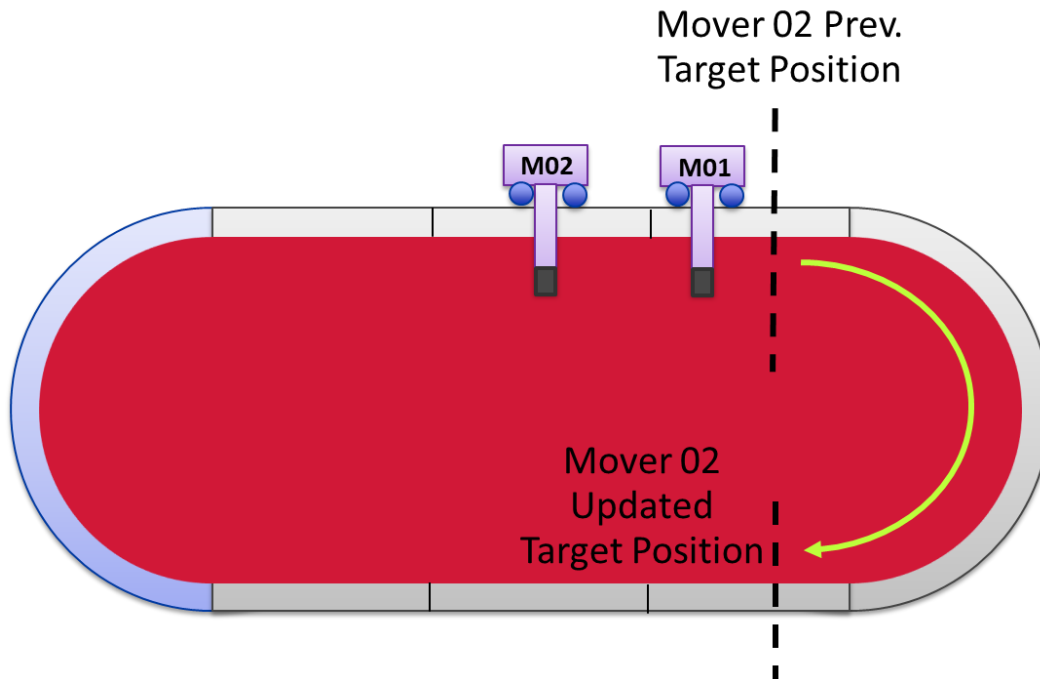
Mover 02 will respect the commanded Distance from preceding Mover



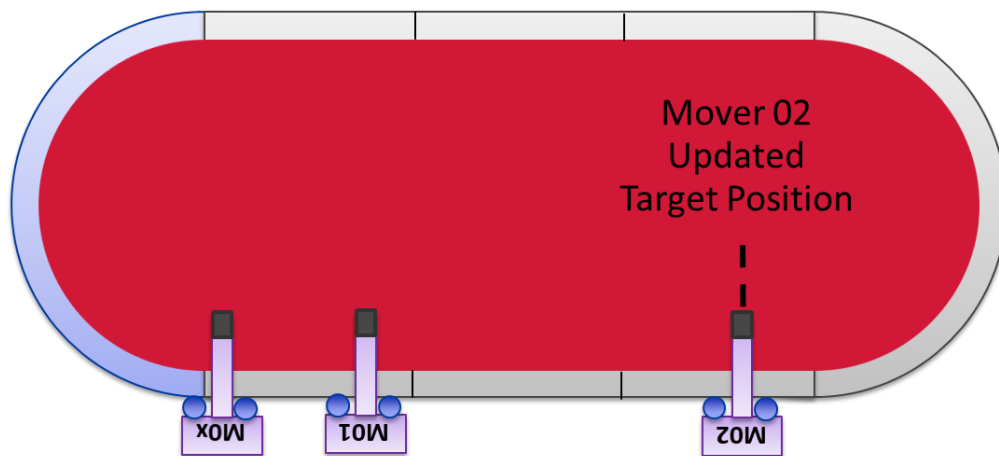
3 – If Mover 01 moves beyond Mover 02 target position Mover 02 will stop at its target



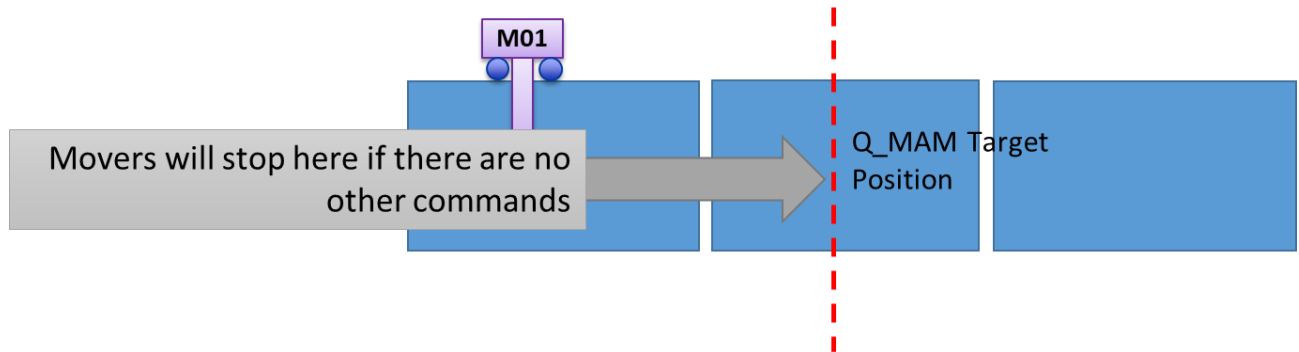
4 – If I change Mover 02 target position during execution the instruction updates the target position



5 – When Mover 02 reaches the target position it stops. Mover 01 is free to continue moving unless it is limited by another mover

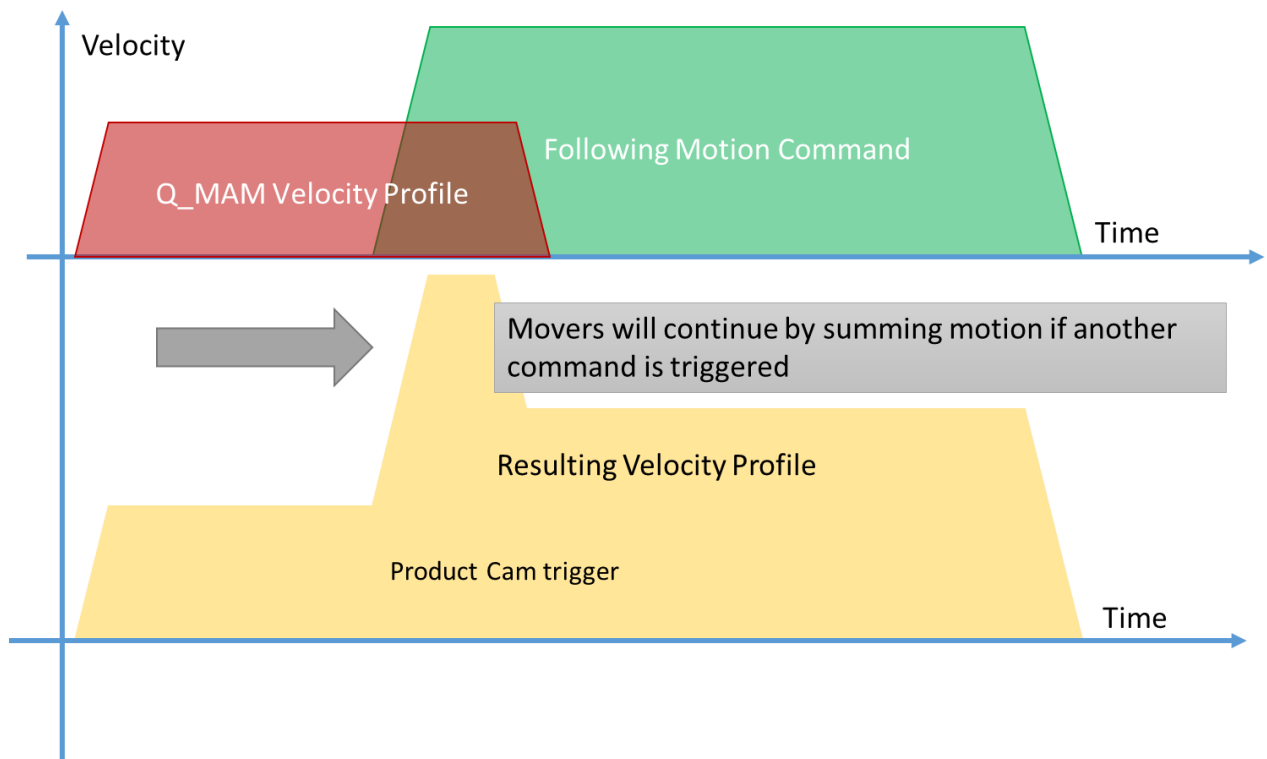


## Q\_MAM\_P: Sum Motion Profiles



When enabled the Q\_MAM\_P instruction terminates all current movements assigned to this Axis.

Q\_MAM\_P internal motion is handled incrementally so that a second motion command (i.e. a Position Cam) can be summed without issues.

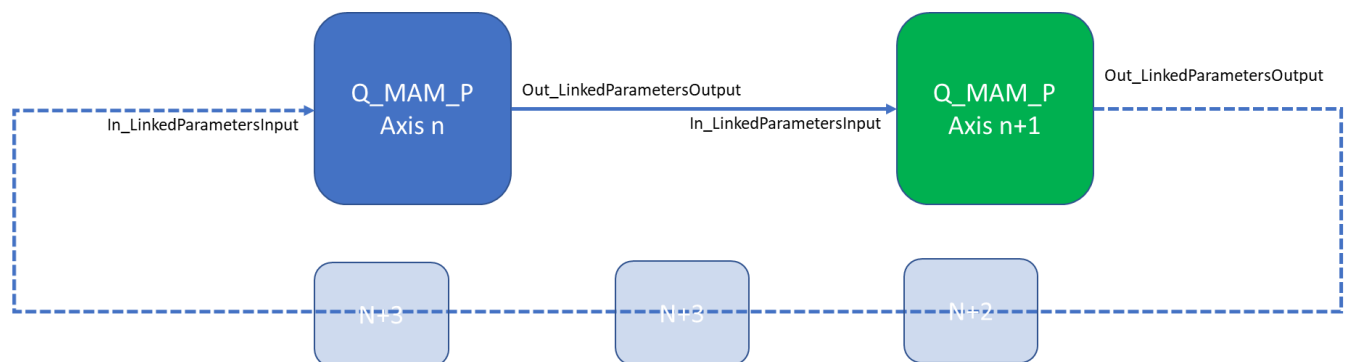


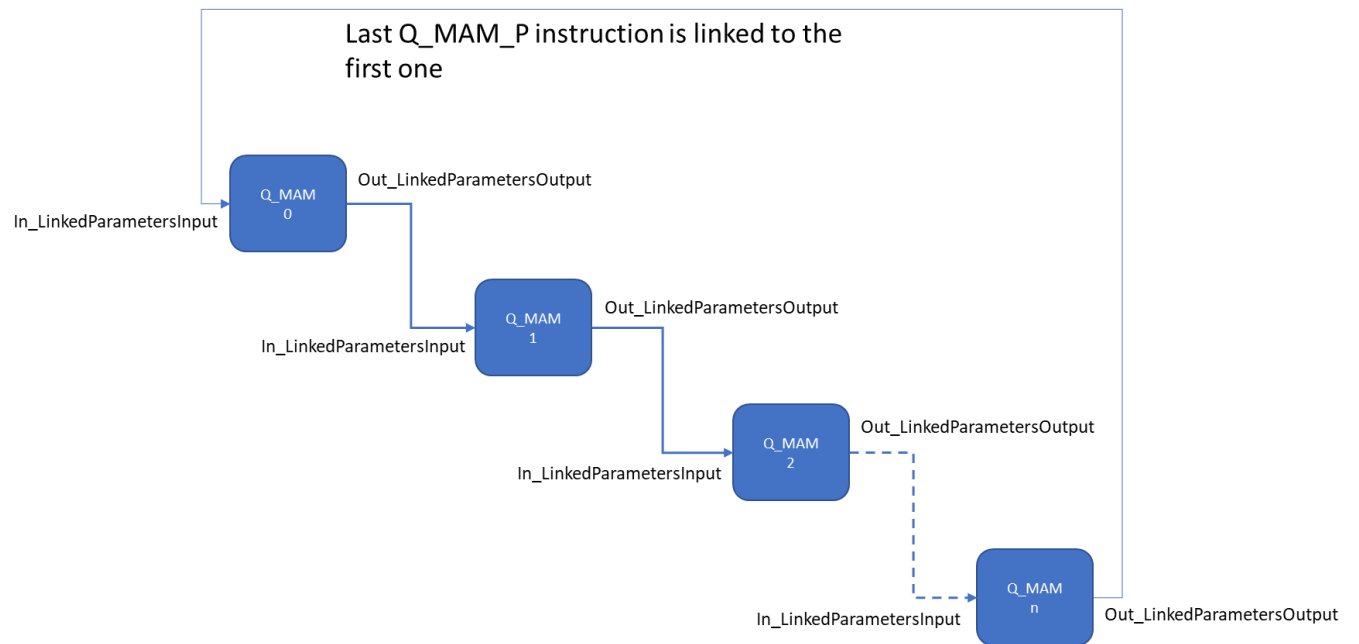
Q\_MAM\_P allows motion command to be summed. I.e. you can enable a Position cam before the mover arrive to its Q\_MAM\_P target, to guarantee a continuous motion behavior.

## Q\_MAM\_P: CASCADE LINKS

| Out_LinkedParametersOutput → In_LinkedParametersInput |      |  |
|---|------|--|
| PPI_State   | BOOL | Tells the following Mover whether this mover is a follower or a Leader |
| PPI_Value   | REAL | Propagated Position Increment Value                                    |
| PTI_State   | BOOL | Tells the following mover the actual state of PTI Feedforward Mode     |
| PTI_Value   | REAL | Propagated Target Increment Value                                      |

Each Instance of the Q\_MAM\_P instruction is linked to the previous Axis instruction and to the following Axis Instruction to build a cascade chain of movers status. **Out\_LinkedParametersOutput** and **In\_LinkedParametersInput** propagates data for position Feedforward

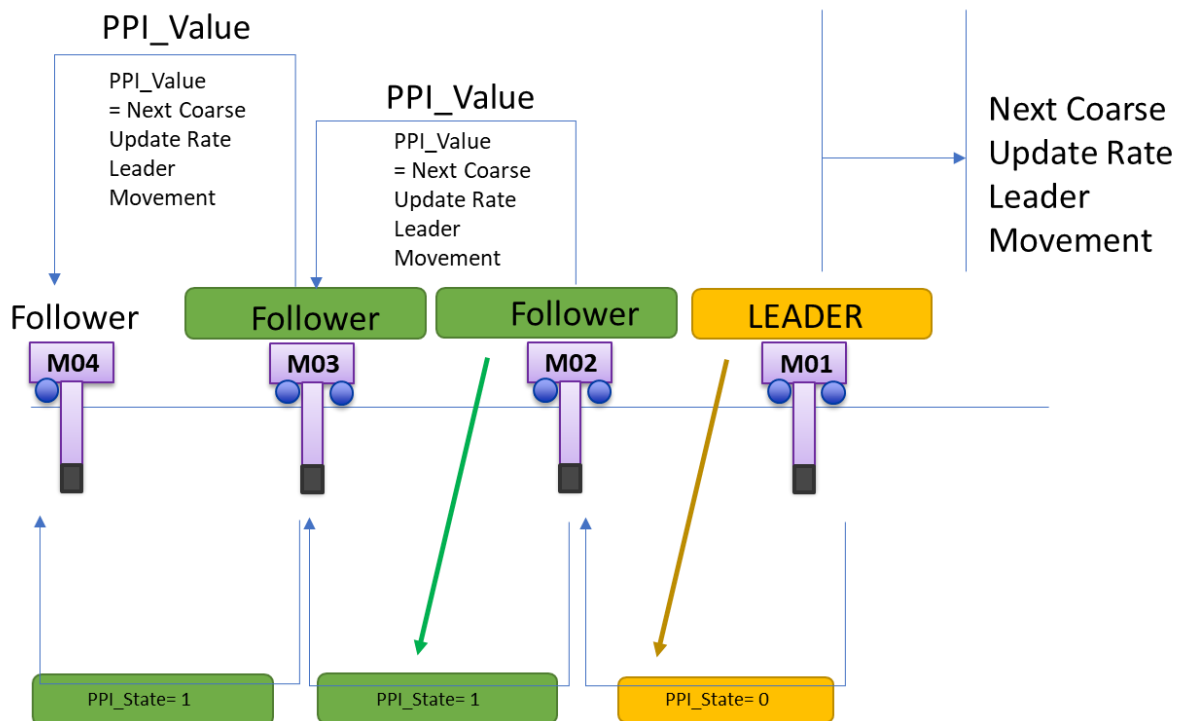




## Q\_MAM\_P: FEEDFORWARD MODE

The Q\_MAM\_P instruction allows two different method to compensate for cascade delay.

### Cfg\_Feedforward\_Mode = 0 -> Leader Axis Current Position



The first Follower Axis (it knows being the first follower as the PPI\_State signal from preceding Axis is zeroed) calculates the Next Coarse Update Rate Leader Position Increment.

Every follower then propagates this signal to the rest of the followers' axes. As they know reading PPI\_State signal.

This way every axis compute as free space to move in the position of its preceding axis or the Leader propagated position increment, mitigating the cascade delay effect

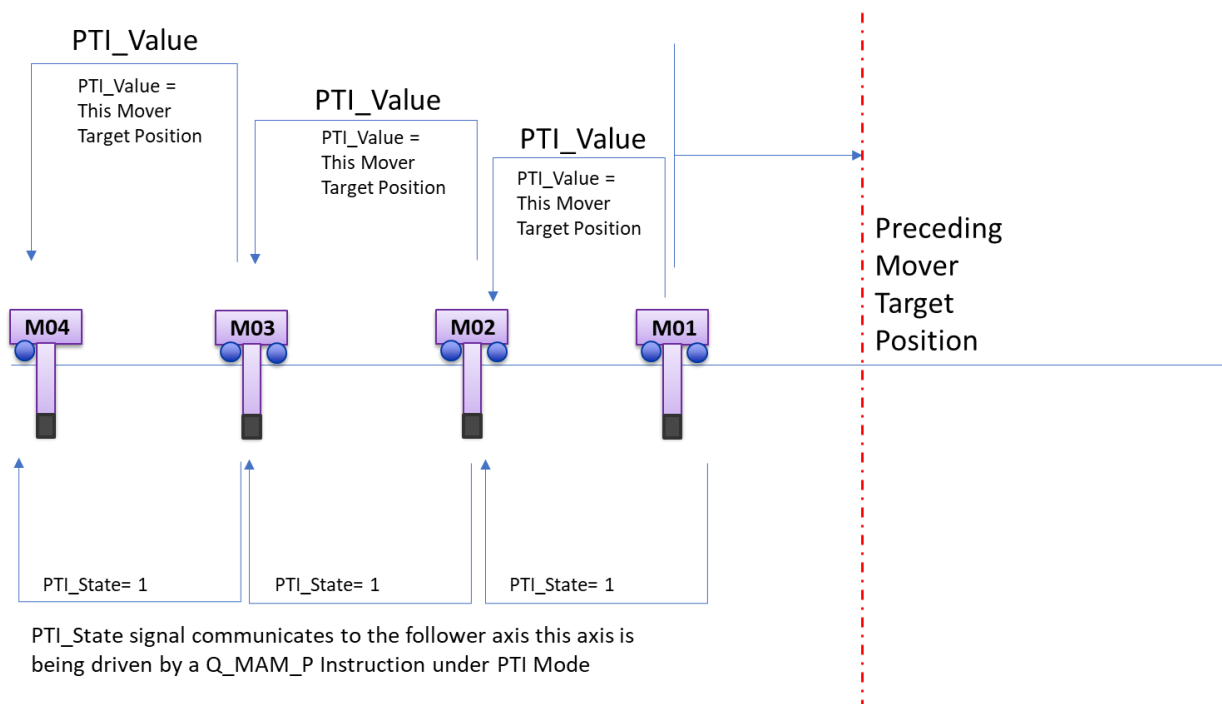
## Cfg\_Feedforward\_Mode = 1 -> Preceding Axis Target Position

When Cfg\_Feedforward\_Mode = 1 (Preceding Axis Target Position) **each Axis transmits its current allowed target position to the preceding mover.**

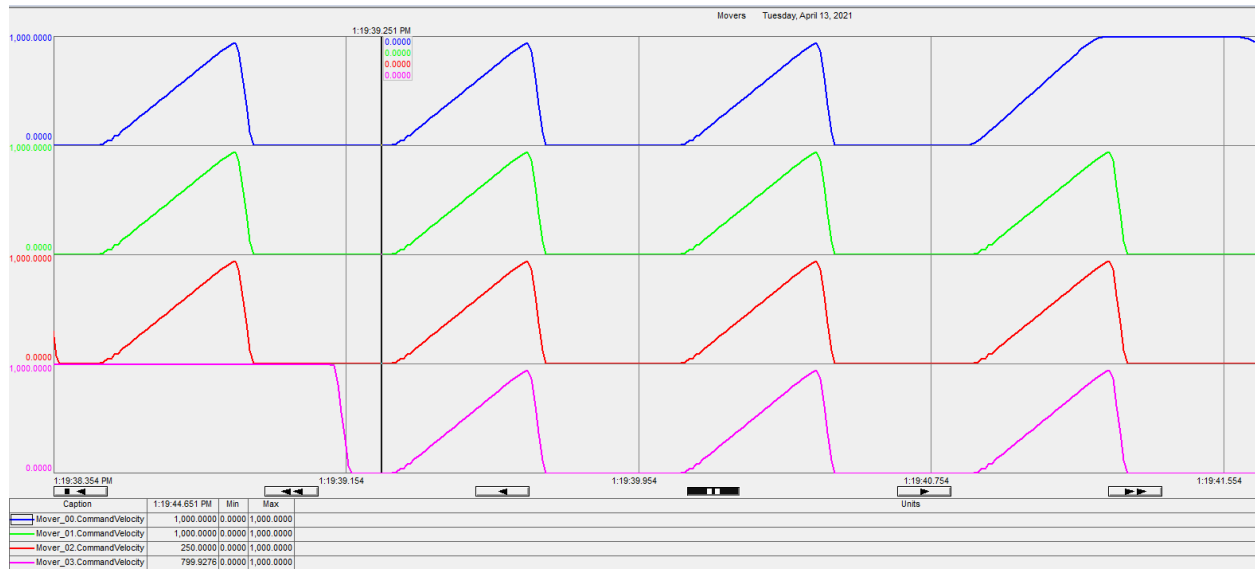
Q\_MAM\_P instruction uses this value to compute follower axis free space to move forward.

This way the instruction considers the preceding mover as already arrived at its target position.

Using the Cfg\_Feedforward\_Mode = 1 nearly cancels all delay effect due to queuing.







As we can see in the above picture representing velocity of queued axes with Cfg\_Feedforward\_Mode=1 the delay effect is practically zeroed, and the axis behave as they were receiving the same command at the same time

**Very Important:** when Using Cfg\_Feedforward\_Mode = 1 ALL Movers MUST be configured with the same Dynamics in the Q\_MAM Instruction

If the follower mover should have any higher dynamics configured movement parameters (Velocity, Acceleration, Jerk) than the preceding Axis this will lead to a probable collision between the axis, as the follower mover considers the space between the preceding axis and its target position as already free for movement.