www.triomotion.com

Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292333
Fax: +44 (0)1684 297929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Ph: +1 412.968.9744
Fx: +1 412.968.9746

**T E C H N I C A L   N O T E**

**Doc No.:** **TN30-17**
**Date:** **Dec 2003**
**Version:** **1.0**
**Subject:** **Look-ahead move sequence**

# Look Ahead Move Buffer and Speed Control

Development Implementation Notes ver 4: 26 July 2002 CDB

**These notes apply to system software version 1.6055**
The “look ahead move buffer” is designed to allow sequences of short high speed moves to be merged together. With the usual Motion Coordinator software the VPU (velocity profile unit) can only look into the NTYPE “next move” buffer in order to decide if deceleration is required. If the deceleration distance is say 20mm and 10mm moves are being executed the VPU will start to decelerate the axis unnecessarily.

The current implementation of the “look ahead move buffer” allows up to 16 moves to be buffered ahead under many circumstances. This number could potentially be altered if required.

Absolute moves are converted to incremental moves as they enter the buffer. This is essential as the vector length may be required to decide on deceleration. However it should be noted that if a move is cancelled by the programmer the absolute position will not be achieved.

The software will wait until the buffer is clear (and therefore make the buffering system similar to earlier versions) if:

(1) The interpolating group is changed for an axis.

(2) A MOVEMODIFY, CAM, CAMBOX, CONNECT, MOVELINK, REVERSE or FORWARD is to be loaded. (Note that these are all single axis moves)

The CAMBOX, MOVELINK and CONNECT don’t use the VPU at all so there is little point in using the look-ahead buffer. If this type of moves were in the buffer the VECTOR_BUFFERED couldn’t be calculated.

The REVERSE and FORWARD commands do not have to check deceleration distance.

There are 3 BASIC axis parameters which allow the programmer to check what is going on in the buffer:

**ENDMOVE_BUFFER** This holds the absolute position at the end of the buffered sequence. It is adjusted by OFFPOS/DEFPOS. The individual moves in the buffer are incremental and do not need to be adjusted by OFFPOS.

**VECTOR_BUFFERED** This holds the total vector length of the buffered moves. It is effectively the amount the VPU can assume is available for deceleration.

**MOVES_BUFFERED** This returns the number of moves being buffered by the axis.

CANCEL(1) clears all the moves in the buffer.  CANCEL clears the current move.

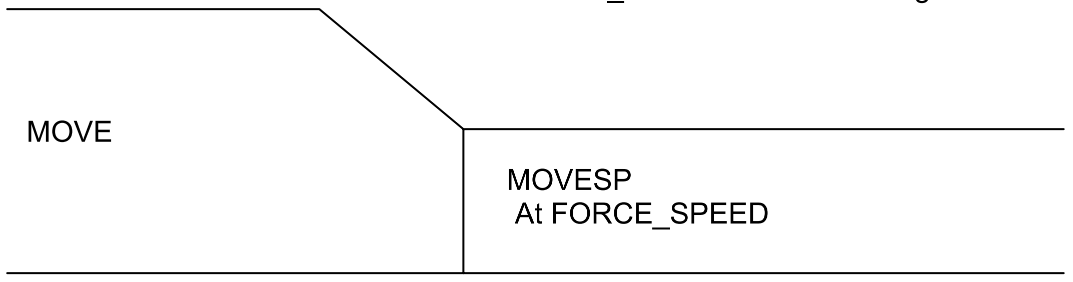
From version 1.6052 in the look ahead version only there are 4 new move types added.  These are equivilents to the existing MOVE, MOVEABS, MOVECIRC and MHELICAL.  The new move types are:

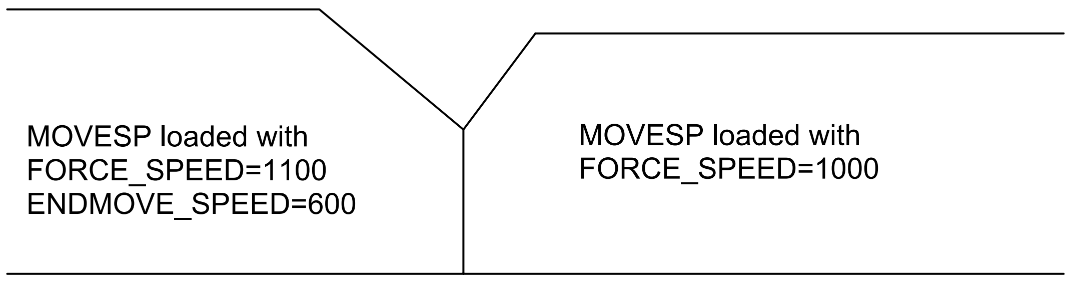
| | |
|---|---|
| **MOVESP** | (MTYPE=6) |
| **MOVEABSSP** | (MTYPE=7) |
| **MOVECIRCSP** | (MTYPE=8) |
| **MHELICALSP** | (MTYPE=9) |

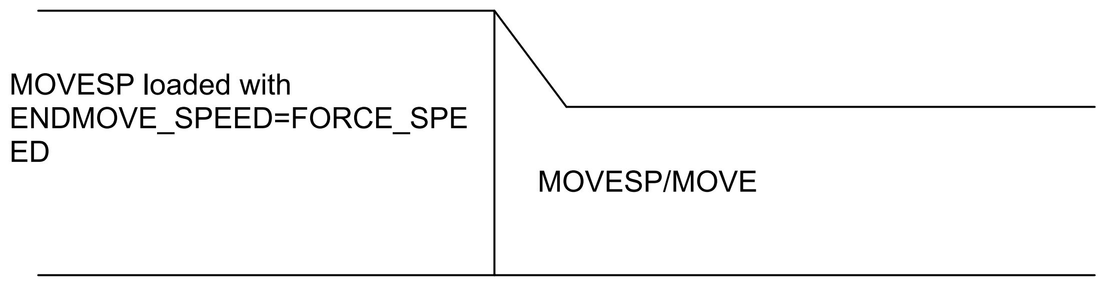
These move types have exactly the same parameters as the none forced speed versions.  However as they are loaded into the move buffer the **FORCE_SPEED** and **ENDMOVE_SPEED** are copied into the buffer with the move.

The 2 new axis parameters ENDMOVE_SPEED and FORCE_SPEED are used to directly control the speed and ramp to exit speed when the MTYPE is 6 to 9.    The ENDMOVE_SPEED and FORCE_SPEED parameter should be set to the required values *at the time of loading the move into the buffer*.

If 2 forced speed move types are merged the speed at the merge point is controlled by the ENDMOVE_SPEED of the outgoing move.  If  a forced speed move follows one or more non-forced speed moves the deceleration will occur to the FORCE_SPEED of the incoming move:







Example 1:

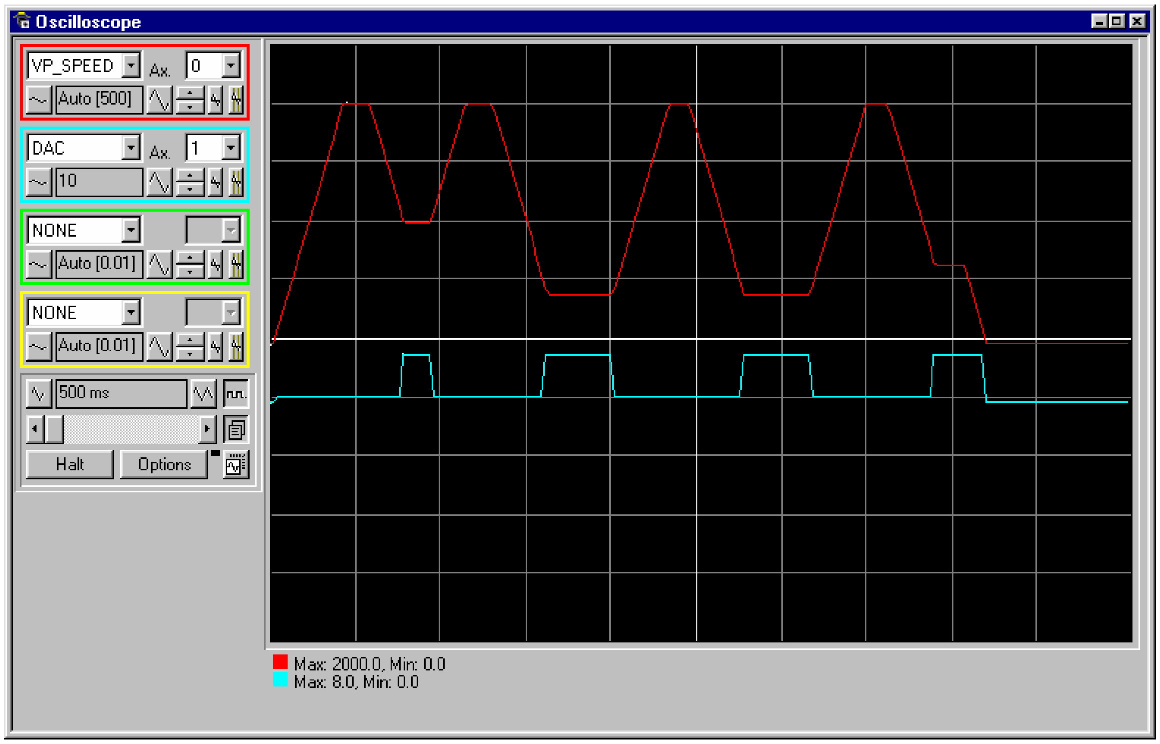
```
SPEED=2000

MOVE(0,1000)
FORCE_SPEED=1000
ENDMOVE_SPEED=1000
MOVECIRCSP(100,100,100,0,1)

MOVE(1000,0)
FORCE_SPEED=400
ENDMOVE_SPEED=400
MOVECIRCSP(100,-100,0,-100,1)

MOVE(0,-1000)
MOVECIRCSP(-100,-100,-100,0,1)

MOVE(-1000,0)
FORCE_SPEED=650
ENDMOVE_SPEED=650
MOVECIRCSP(-100,100,0,100,1)
```



🟥 Max: 2000.0, Min: 0.0
🟦 Max: 8.0, Min: 0.0

The ‘scope trace shows the speed being ramped down firstly from 2000 to 1000, then to 400 and finally to 650 for the MOVECIRCSP. The blue trace shows the MTYPE. This is being copied into the DAC parameter of axis 1 to allow it to be displayed by the ‘scope.

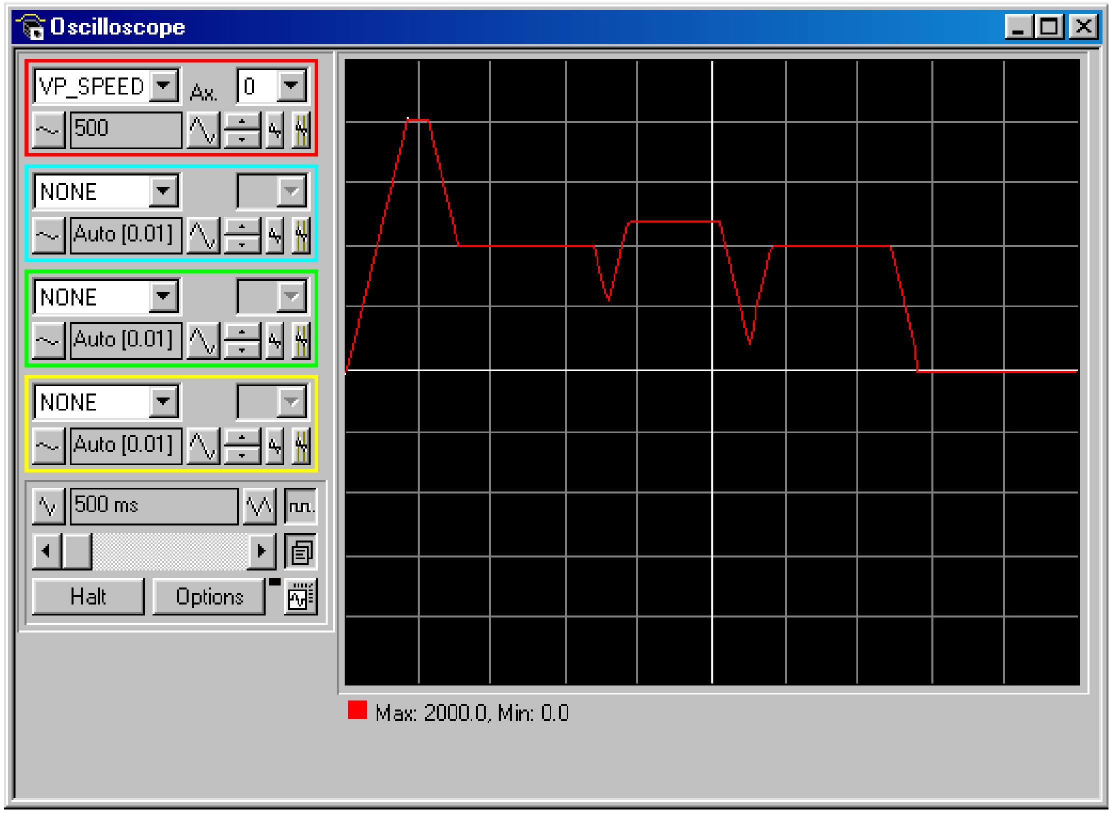
Example 2:

```
MOVE(0,1000)

FORCE_SPEED=1000
ENDMOVE_SPEED=500
MOVESP(1000,0)

FORCE_SPEED=1200
ENDMOVE_SPEED=100
MOVESP(1000,0)

FORCE_SPEED=1000
ENDMOVE_SPEED=500
MOVESP(1000,0)
```



🟥 Max: 2000.0, Min: 0.0