PTS AND PTS TOOLKIT SOFTWARE

MOTION GENERATOR PACKAGE

USER MANUAL

Issue 1

June 1996

(MAN520)

Copyright Notice

Software Version

This manual reflects the following firmware/software versions:

- PTS Toolkit software version 1.2 (Motion or Professional edition)
- PTS Firmware version 1.7.3 or later

Important Notice

Quin Systems reserves the right to make changes without notice in the products described in this document in order to improve design or performance and for further product development. Examples given are for illustration only, and no responsibility is assumed for their suitability in particular applications.

Although every attempt has been made to ensure the accuracy of the information in this document, Quin Systems assumes no liability for inadvertent errors.

Suggestions for improvements in either the products or the documentation are welcome and should be addressed to:-

Quin Systems Limited
Oaklands Business Centre
Oaklands Park
Wokingham
Bershire
RG41 2FD

Telephone:	0118 977 1077
Facsimilie:	0118 977 6728
E-Mail:	Support@Quin.co.uk
CompuServe:	100572, 1631

Contents

1. Introduction

The Motion Generator is a software tool which allows the machine designer to create, modify and implement custom positional mapping between two machine axes. It can be thought of as a tool to design and implement machine cams: but without the limitations of a mechanical cam.

When using the Motion Generator each custom machine motion is called a MAP (one axis is *mapped* to another). There are a number of terms and phrases associated with the motion generator which have specific meanings: see the Motion Generator Glossary of Terms (section 1.2 on page 6) for a full description of these.

This manual covers both the PTS based motion generator and the PC based PTS Toolkit motion generator:

- The PC based PTS Toolkit motion generator is part of the Windows™ based PTS Toolkit. It has been developed to allow you to graphically design custom machine motions between two axes of a machine controlled by a PTS. This graphical design (called a map *shape*) can then either be processed to form a map *table* (a master/ slave position table which the PTS will use to perform positional mapping between the two axes) which can be downloaded and stored in the PTS, or downloaded to a PTS as a map shape and the PTS based motion generator used to perform the map table generation.
- The PTS based motion generator is a built in function that can be purchased for the PTS. It is a calculation engine that processes map shape information provided in PTS variables to generate a map table.

The format of this manual is arranged as follows:

Remainder of this chapter: overview and glossary for the motion generator.

Chapter 2: The PC based PTS Toolkit motion generator. A description of using the motion generator as part of the PTS Toolkit software, explanation of all menus and dialogue boxes used and an example of a typical motion generation problem.

Chapter 3: The PTS based motion generator. A description of the functions and uses of the PTS based motion generator, including a worked example.

Chapter 4: Overview of segment types and their attributes (applies to both implementations of the motion generator).

Chapter 5: Technical information on segment types and general motion generator parameters (applies to both implementations of the motion generator).

1.1 Motion Generator Overview

The motion generator provides the ability for the user to create complex position mappings between one axis and another. This replaces the need for mechanical cams and linkages, and is far more powerful and less restrictive. Before using the motion generator a number of issues need to be understood:

- You are designing a map shape which relates the position of the slave axis to the position of the master axis at any point along the master axis (which will be of a finite, and determinable, length). This is essentially a cam, although you can do things that a cam could not. During the design of the map shape you will determine the map shape using segments, and setting parameters for these segments.
- Certain restrictions have been applied, by default, to the segment to segment boundaries to ensure that the generated map shape is physically achievable:

Slave Derivative	Segment *nn* to Segment *nn+1* boundary conditions (defaults)
Position	**Match**: the start position of segment *nn+1* will match the end position of segment *nn*. *This cannot be altered.*
Velocity	**Match**: the start velocity of segment *nn+1* will automatically be adjusted to match the end velocity of segment *nn*. This may have the effect of modifying the shape of segment *nn+1* which can lead to unexpected segment shapes. [For certain segment types it is possible, but not advisable, to break this matching].
Acceleration	**Unspecified**: for all segment types excluding polynomial and cubic spline acceleration at the beginning of segment *nn+1* is determined by the segment type. Therefore there may be a step change in acceleration across the *nn* to *nn+1* segment boundary. **Match**: a polynomial segment and the cubic spline segment automatically matches their start acceleration to the end acceleration of segment *nn*. For the polynomial segment alone this matching can be broken.
Jerk	**Unspecified**: the jerk value at the beginning of segment *nn+1* is determined by the segment type. Therefore there may be a step change in jerk across the *nn* to *nn+1* segment boundary. For the polynomial segment alone it is possible to change this to match with previous or specify the required value.

Table 1: Default Segment Boundary Conditions

- The map shape wraps round. In other words the last segment connects to the first segment. However the continuity laws across this segment to segment boundary are, by default, different from those on all other segment to segment boundaries:

Slave Derivative	Last Segment to First Segment boundary conditions (defaults)
Position	**Specified**: the end position of the last segment and the start position of the first segment can both be specified. Therefore there can be a step change in position across this boundary (so that it is possible to create a motion for a 'rotary' slave axis). You can manually align the end position of the last segment to match the start position of the first segment if required (for a 'linear' slave axis).
Velocity	**Match**: normally the start velocity of the first segment will be automatically adjusted to match the end velocity of the last segment. The one exception to this rule is when *all* segments inherit their velocity from the previous segment: the first segment will then start with zero velocity. In this case use a constant velocity segment to create any required velocity offset.
Acceleration	**Unspecified**: for all segment types excluding polynomial and cubic spline the acceleration at the beginning of the first segment is determined by the segment type. **Match**: if the first segment is either a polynomial or cubic spline segment then it automatically matches it's start acceleration with the last segment's end acceleration. For the polynomial segment alone this matching can be broken.
Jerk	**Unspecified**: the jerk value at the beginning of the first segment is determined by the segment type. For the polynomial segment alone this can be changed to matching or start jerk specified.

Table 2: Last to First Default Segment Boundary Conditions

- At NO instance does a map shape or map table contain any time dependent information. The velocity and acceleration of the slave axis at any point will depend upon the velocity of the master axis.
  For the PC based PTS Toolkit motion generator only: to enable you to develop a map shape that is within the physical constraints of your machinery it is possible to enter the number of cycles per minute required (how many times the map shape will be cycled through). This sets the master velocity at a constant value and allows the motion generator to put figures on the slave velocity and acceleration shapes.
- The default segment type is polynomial. This segment type is the most configurable of all the available types and should be used in preference to another type whenever possible. With the exception of the start of segment position parameter (which must always match the end of the previous segment) it is possible to modify the boundary constraints for all the derivatives at both the start and end of the segment. This means that it can be configured to provide a smooth transition between known points.

1.2 Motion Generator Glossary of Terms

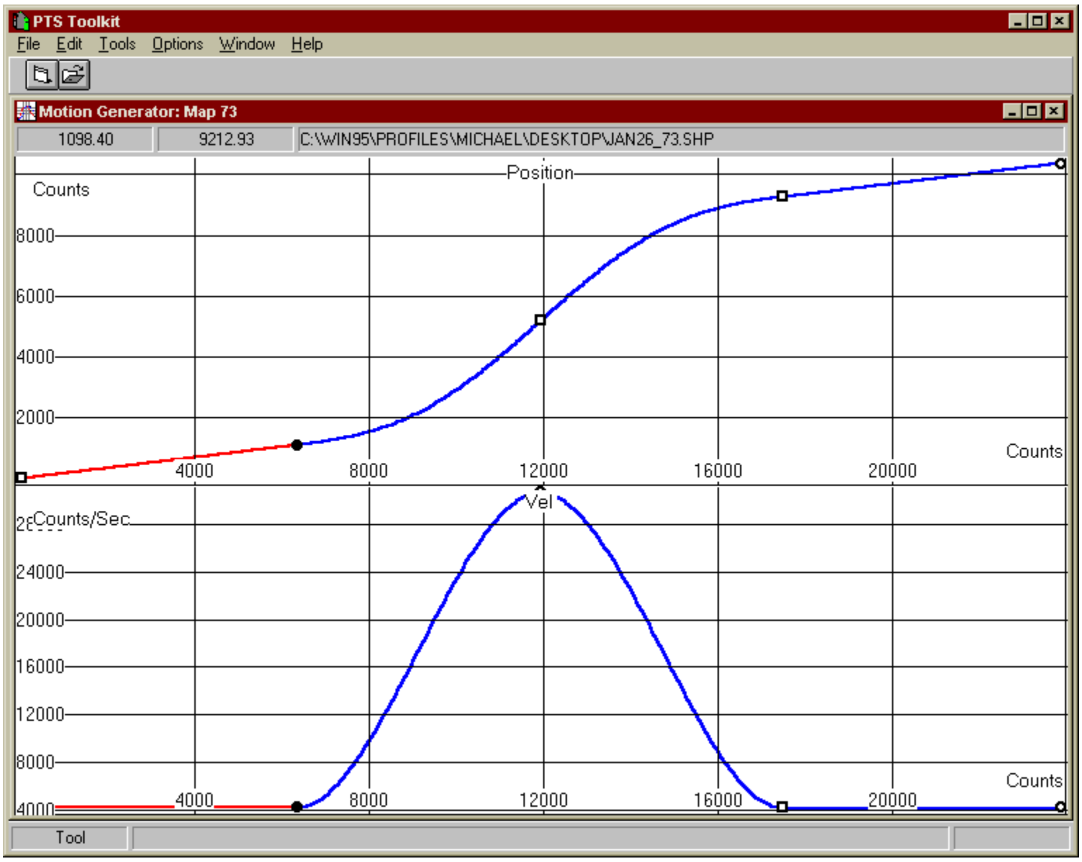
Map Shape: The name given to the information which creates the drawing seen on the PC screen in the motion generator package, to the data loaded and saved from PC disk to create this image, and to the list of variables used in the PTS itself to drive the built-in motion generator. A map shape is a high level description of the master/slave mapping you have created and consists of a list of segments, with the relevant additional parameters for the segment type (PC based only: all of which you can adjust using the mouse or dialogue boxes). The value of the slave position at any given master position can be determined by calculations using the map shape data. Contrast this with a map table.

Map Table: The name given to a list of co-ordinates (both master and slave) which, when plotted, will produce the drawing seen on the PC screen of the PTS Toolkit motion generator. The resolution of this drawing will depend upon the resolution of the map table. A PTS will use the map table (and perform a linear interpolation between adjacent points) to control the relative mapping of the slave axis to the master axis.

Segment: A single section of the map shape. A segment will have a type to define its shape and may well have certain additional parameters which affect its shape.

2. PC based PTS Toolkit Motion Generator

The PC based PTS Toolkit motion generator forms one of the tools available through the PTS Toolkit (others include tuning screens and PTS ’scope).



The motion generator window presents the current map shape as a series of segments, in the above picture both the position and velocity derivatives are being viewed. The currently selected segment is highlighted in red (using the motion generator default colour scheme) and the position of the mouse pointer is given in map co-ordinates in the top left hand corner of the motion generator window.

When the motion generator is first started a splash screen will be displayed while the program loads. You are then presented with a blank map consisting of a single segment. This map can then be edited as required, or a previous session loaded from the PC disk.

The following three sections detail all the menus, dialogue boxes and mouse actions associated with the motion generator. Then section 2.4 on page 19 gives a worked example of using the PC based PTS Toolkit motion generator. The final section in this chapter gives details of how to use the generated map in the PTS.

2.1 Menus

The PTS Toolkit menus change when different tools are selected. This section details the menu options presented for the motion generator tool:

2.1.1 File Menu

New: Gives a new, blank, map for editing. You will be prompted to initially configure the map using the map shape settings dialogue (this same dialogue can be used at a later stage to alter the map configurations) (see section 2.2.1 on page 10 for an explanation of the dialogue box).

Open: Loads a map (that you had previously saved) from the PC disk. The file extension used is '.SHP'.

Save: Saves the current map to the PC disk. The file extension used is '.SHP'.

Save as PTS file: Saves the current map shape to the PC disk as a map shape. This is the format used to download the map shape to the PTS. The file extension used is '.PTS' and the file can be loaded into PTS Note or downloaded to a PTS via the terminal window. Three dialogue boxes are presented during this process: the standard windows file save dialogue box to specify a file name, a dialogue box (see section 2.2.4 on page 18) allowing you to configure how the map shape is stored in the PTS and finally a dialogue allowing you to specify the required number of points on the map (see section 2.2.3 on page 17).

Save as Camlinks Data: Saves the current map shape to the PC disk for reading into the Camlinks motion modelling package (in the '.MOT' file format, detailed in the Camlinks documentation).

Import PTS file: Loads a file which contains PTS variables forming a map shape (extension '.PTS'). The file can contain standard PTS mathematical equations which will be interpreted in the same manner as in a PTS.

NOTES:

1) If you save a map in both the default format and as a map shape you may notice differences when loading both formats back into the motion generator. This is because the two formats have different resolutions.

2) The standard PTS mathematical operators are supported, but PTS logic or multi statement lines are not. Any necessary, but unknown, PTS variables will be prompted for.

Download Map Table: The map table is downloaded into the PTS. This can then be used to map the motion of one axis to another. A dialogue box (see section 2.2.3 on page 17) is presented when this menu option is selected allowing you to configure the generation of the map table, and providing some useful information for using the map table in the PTS.

NOTES:

1) A terminal screen needs to be open for this option to be available.

2) A map table can only be downloaded into a PTS with a valid motion generator licence. The PTS Toolkit software will confirm this before proceeding with the operation: a dialogue box will be used to show the progress of this.

3) The table can also be downloaded as a Profile, for single axis execution. Again, a motion generator licence is needed.

Download Map Shape: A terminal screen needs to be open for this option to be available. The map shape is downloaded into the PTS. The motion generator within the PTS can then be used to generate a map table. Two dialogue boxes (see section 2.2.3 on page 17 and section 2.2.4 on page 18) arepresented when this menu option is selected allowing you to configure how the map shape is stored in the PTS, and the number of points you require in the map table when generated by the PTS.

Most Recently Used Files List: Provides a convenient shortcut to files which were used recently in the PTS Toolkit.

Close Motion Generator: Closes the Motion Generator package (but doesn’t quit the PTS Toolkit).

Exit: Quits PTS Toolkit.

2.1.2 Edit Menu

Undo: The motion generator provides an undo feature allowing you to backtrack over recent changes.

Redo: Allows you to forward track through previous undo’s (only available if you have not done any editing after undoing)

Autoscale: Scales the map shape to fit neatly into the available screen size.

Halve scale: Halves the current scaling of the map shape.

Zoom: Allows you to zoom into any visible part of the map shape by dragging a rectangle with the mouse over the area you require.

Append Segment: Adds a segment to the end of your map shape. The segment will be 200 counts long (scaled as appropriate) and of a polynomial type.

Delete Segment: Deletes the currently selected segment. You cannot delete the segment if it is the only one in a map shape.

Insert Segment: Inserts a segment to the left of the currently selected segment. The new segment will be 200 counts long (scaled as appropriate to the user units) and of a polynomial type. All segments to the right of the inserted segment will be shifted to the right by the appropriate amount.

Split Segment:Splits the current segment at the position indicated by clicking with the mouse. Only certain segments are 'splittable', and this menu option will not be available if the currently selected segment cannot be split.

Edit Segment: Configure the currently selected segment settings using the dialogue box detailed in section 2.2.2 on page 13.

Edit Map Settings: Configure the map shape settings using the dialogue box detailed in section 2.2.1 on page 10.

Display Colours: Allows you to modify the colours of the motion generator screen display.

Line Settings: Allows you to modify the thicknesses map shape line on the screen display.

2.1.3 Options Menu

Position/Velocity/Acceleration/Jerk: You can choose which derivatives are displayed on the motion generator screen at any one time. Each derivative is displayed as a horizontal tile, with the position derivative at the top of the screen. You can work in any derivative window, and see how changes you have made affect the other derivatives as you work. It is not possible to turn off the display of all four derivatives: the position derivative will always be made visible if you do.

Auto Y Scale: When selected this option causes a continuous auto Y scale of the map shape.

Grid: Toggles the grid on the motion generator screen display on or off.

Toolbar: Toggles the toolbar display on or off.

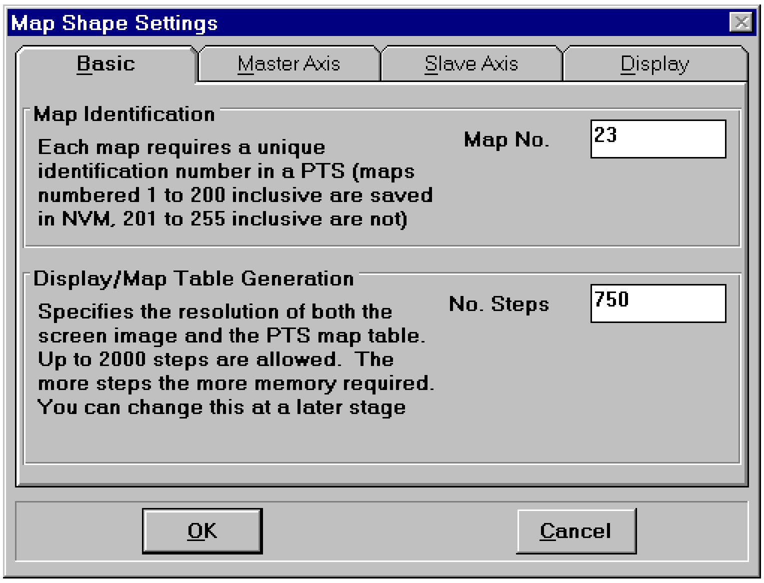
2.2 Dialogue boxes

The following dialogue boxes are presented by menu options or mouse actions when using the PC based PTS Toolkit motion generator, and are unique to the motion generator:

2.2.1 Map Shape Settings

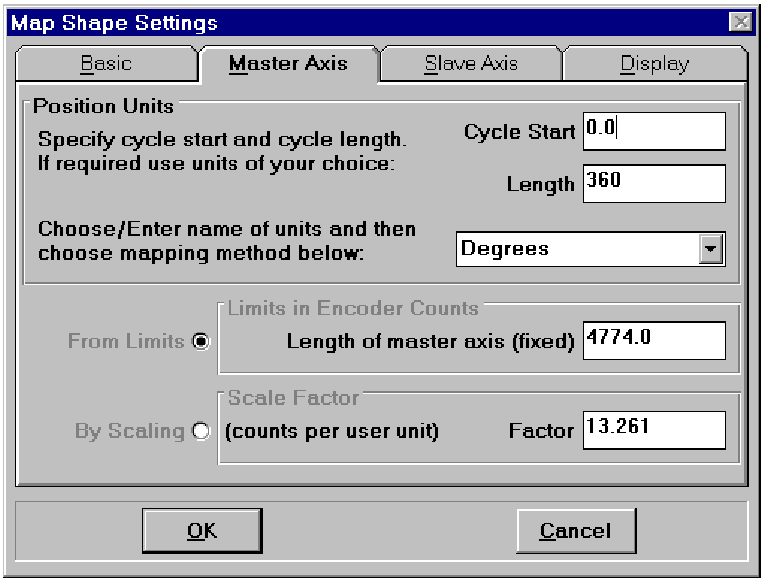
The map shape settings dialogue is presented for a new map, or when the menu option Edit map settings is chosen from the Edit menu. The dialogue consists of four tabs: Basic, Master Axis, Slave Axis and Display. Each tab deals with an aspect of the global settings required to display a map shape appropriately.

When a new map is created you will be required to visit all four tabs. Later you can adjust individual settings by selecting the appropriate tab.



Map Identification: Each map has a unique identifier, the in the range 1 to 255 inclusive. This identifier will also be used for the map table when downloaded into the PTS. Maps numbered 1 to 200 inclusive will be saved in NVM (Nonvolatile memory) in the PTS by the ‘SP’ command, maps numbered 201 to 255 inclusive are not, and can therefore can be used as ‘working’ maps.

Number of Map Steps: When a map shape is turned into a map table the motion generator needs to know what resolution you require. The more map steps the greater the resolution of the map table, but also the larger the table. The maximum number of steps is 2400.



Master cycle start co-ordinate: This is the start of map position co-ordinate for the master axis and can only be altered using this dialogue box.

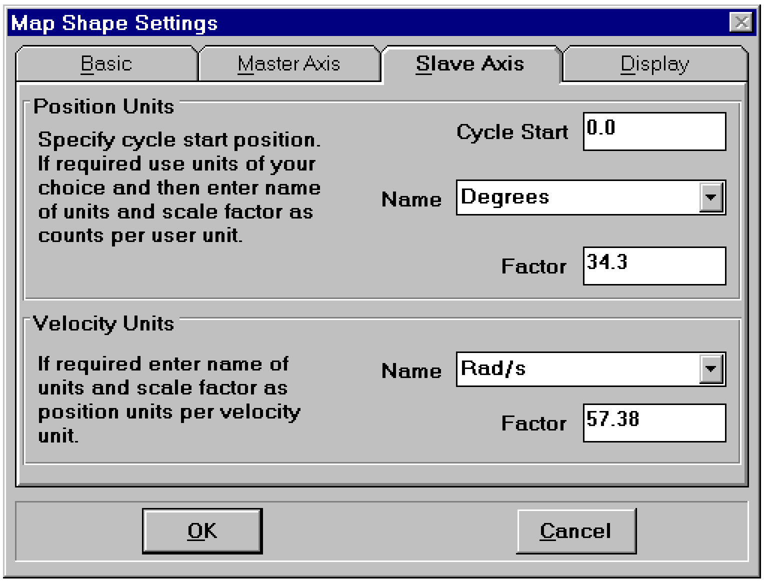
Notes:

1. The master start co-ordinate is a relative value. It shifts the master axis by the specified amount. Its value is in user units.

2. In the PTS the master axis of a map will *always* start at zero counts. Therefore use the PTS commands MB or MF to adjust this to match the PC based master cycle start co-ordinate if required.

Master length: The length of the map shape along the master (horizontal) axis (in position units).

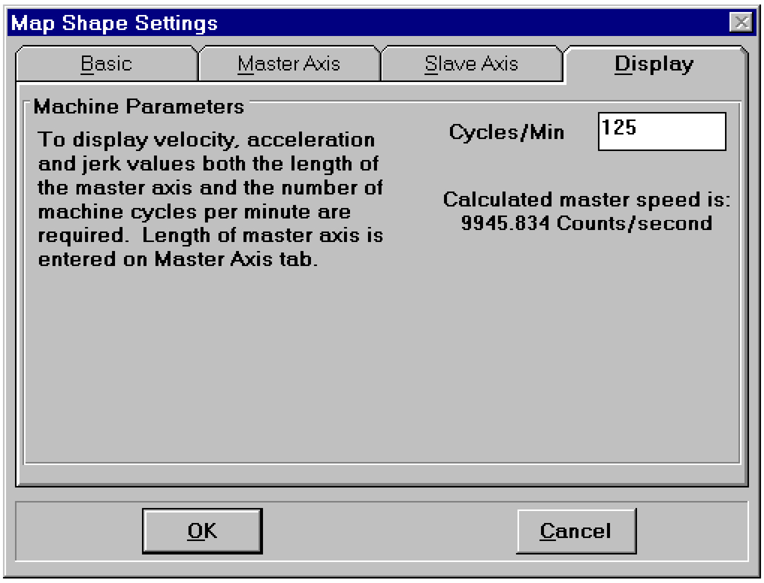
Master axis units & scaling factors: It is possible to work in engineering units rather than encoder counts. To do this the Motion Generator requires to know the conversion factor between your chosen units and counts, entered either as the length of the master axis in both units or as a scale factor. Note that when changing to/from encoder counts this dialogue box will limit what changes can be made. To change both the units (to/from encoder counts) and the axis length requires two visits to the dialogue.



Slave cycle start co-ordinates: This is the start of map position co-ordinate for the master axis and can only be altered using this dialogue box.

Note: The slave start co-ordinate is an absolute value. Its value is in user units.

Slave axis units & scaling factors: It is possible to work in engineering units rather than encoder counts. To do this the Motion Generator requires to know the conversion factors. The slave position conversion factor is in encoder counts per user unit. The slave velocity conversion factor is in user position units per user velocity unit.



Cycles/Min.: A map shape does not contain any time data. However, to allow you to model the shape correctly, and within the mechanical limits of your machine, you can set the machine cycles per minute. This determines the (constant) velocity of the master axis and allows the motion generator to put a scale to the slave velocity, acceleration and jerk axes.

2.2.2 Edit Segment Settings

The edit segment settings dialogue box (chosen from the Edit menu) brings together all the parameters that apply to each segment in one dialogue box. To do this 'tab' buttons are used. The dialogue is also different from standard Windows dialogue boxes in that it can remain on the screen whilst you edit the map via the mouse or use other menu commands. This is a very powerful feature, allowing you to move quickly about your map shape and analyse certain points.

If you make changes to the parameters in this dialogue you will notice that some happen immediately whilst others wait for you to press the 'Apply numeric changes' button (or press the enter key with the relevant parameter selected). This update method has been chosen to reflect the purpose of this dialogue: to analyse the current segment as well as make changes. The features available in this dialogue can be divided up into the three tabs, and general controls.

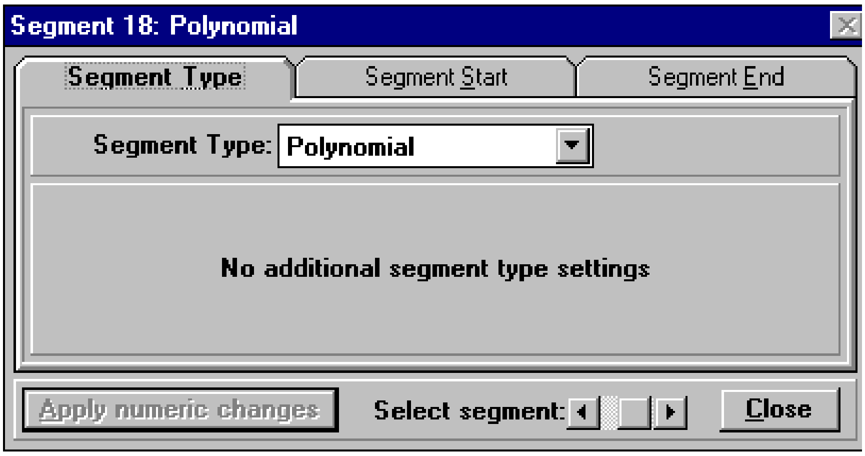
2.2.2.1 General Controls

Apply numeric changes: if you make numeric changes in the current tab these can be applied by pressing this button, or the return key for individual entries.

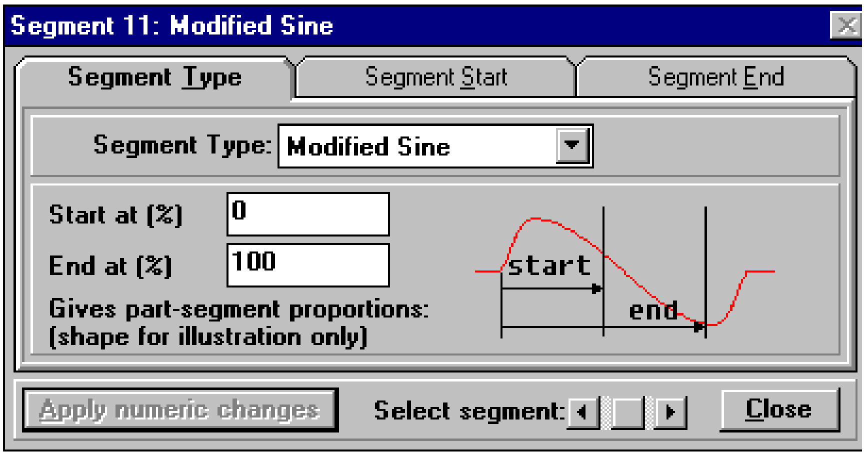
Select segment scroll bar: You can step through the segments in your map shape using this scroll bar, allowing you to analyse the settings of each segment.

Close button: Closes (hides) the dialogue box.

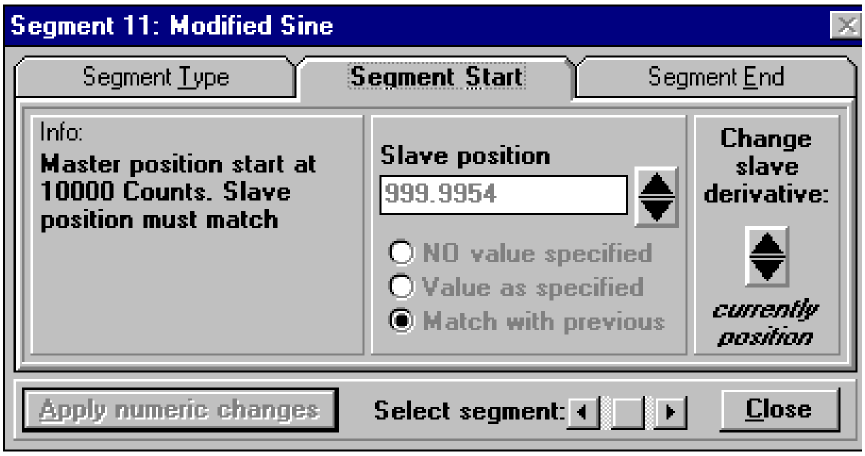
2.2.2.2 *Segment Type Tab*



The parameters available in the segment type tab depend upon the currently selected segment. In the above picture a polynomial segment is selected and there are no other settings available (all polynomial segment settings are performed in the other two tabs). In the example below a modified sine segment is selected and there are two extra settings: a start and end percentage for the segment.



2.2.2.3 Segment Start Tab



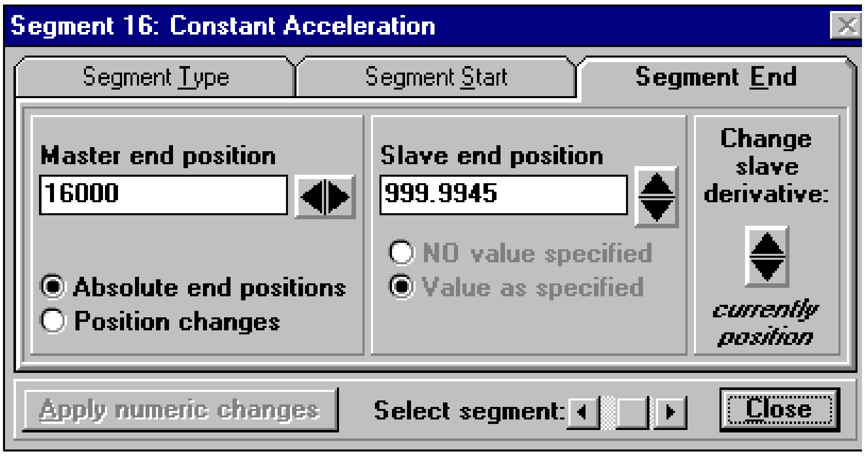
The segment start tab displays the parameters and constraints that apply at the start of the selected segment.

Info.: This text box contains information relevant to the start of the segment and the current derivative.

Slave Position/Velocity/Acceleration/Jerk: Depending upon the current derivative selected the slave display will change. The radio buttons will show what (if any) options are available. The text entry box (or the arrow buttons alongside the text box) can be used to specify a new value when radio button option 'Value as specified' is selected.

Slave derivative: For each segment type there are four constraints given by the four slave derivatives available; position, velocity, acceleration and jerk. Use these arrow buttons to change the current derivative information displayed in the dialogue box.

2.2.2.4 Segment End Tab



The segment end tab displays the parameters and constraints that apply at the end of the selected segment.

Absolute/Relative positions: two radio buttons allow you to choose whether the position co-ordinates are displayed as relative (position changes across the segment) or absolute (end of segment position co-ordinates) co-ordinates. This also affects what happens if you change the values, as will be explained below.

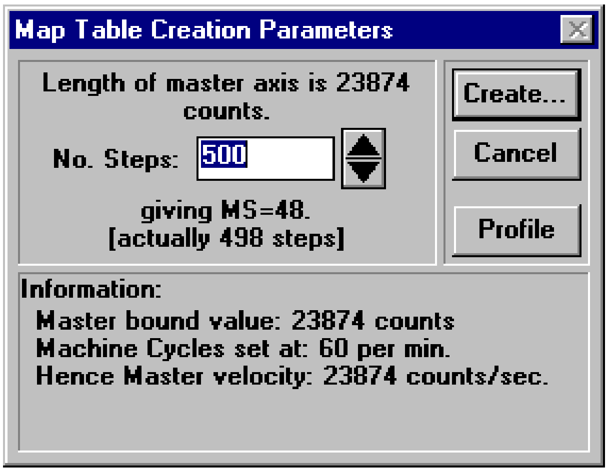
Master position change *OR* Master end position: the length of the segment can be specified in one of two ways, as a relative or an absolute co-ordinate. This is determined by the two radio buttons immediately below the master axis text box. By modifying the absolute master position *both* the segment being edited and the next (higher) segment are altered, this is the same as when using the mouse to perform this operation. [Note: if the next segment cannot accommodate the position change then a ripple through will happen until a segment which can accommodate this change is reached]. By modifying the relative master position (master position change) the segment being edited is affected and *all* higher segments are moved to match the new segment length.

Slave Position/Velocity/Acceleration/Jerk: Depending upon the current derivative selected the slave display will change. The radio buttons will show what (if any) options are available. The text entry box (or the arrow buttons alongside the text box) can be used to specify a new value when radio button option ‘Value as specified’ is selected.

When slave position is selected this can either be a position change across the segment or an absolute end position. Altering the slave position change will have the effect of modifying the current segment *and* moving *all* higher segments in a relative manner. Altering the slave absolute end position will have the effect of modifying the current segment and the next (higher) segment. [Note: if the next segment cannot accommodate the position change then a ripple through will happen until a segment which can accommodate this change is reached.]

Slave derivative: For each segment type there are four constraints given by the four slave derivatives available; position, velocity, acceleration and jerk. Use these arrow buttons to change the current derivative information displayed in the dialogue box.

2.2.3 Map Table Creation Parameters



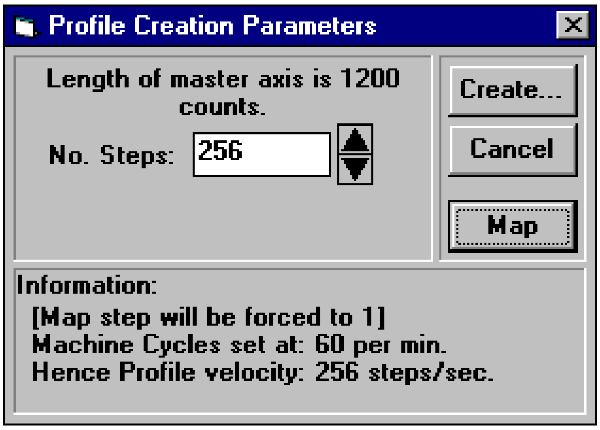
When either creating a map table to download to a PTS, or saving/downloading PTS map shape information this dialogue box is presented. The information contained within the dialogue is very useful when using the map in the PTS and a note should be made of it. The number of steps parameter sets the resolution of the map table (max. 2400). The higher the resolution of the table, the smoother the motion will be, though choosing the number of steps also depends upon the length of the master axis and the complexity of the map shape. The value of the PTS variable ‘MS’ is displayed, determined from your choice of number of steps and the length of the master axis. Due to the use of integers this will sometimes result in fewer actual steps in the map table, as is the case above.

When the number of steps (and hence ‘MS’) does not form an exact integer divisor of the master axis length the motion generator will create a map table with the final entry outside the master bound. This point is caculated by extrapolating a constant velocity line through the previous table entry and the desired master and slave bound values. When the PTS uses the map table it performs linear interpolation between entries and hence the desired slave bound will be met at the master bound (use ‘GM’ on the slave axis to confirm this when in mapping).

The map table creation dialogue box can also be used to generate Profiles. A profile is a single axis map: a series of position co-ordinates relating to *time*.You would use a profile to generate a specific movement for a motor, for example a sine squared acceleration.

This is an advanced use of the PC based PTS Toolkit motion generator, as the X axis is now *time ticks* at a step rate defined by the PTS parameter PV. Therefore you should work in the default units (master encoder counts, slave position encoder counts and slave velocity encoder counts per second). When you create a profile you will choose the number of steps in the profile, each step will take 1/PV seconds.

When the profile button is pressed the dialogue box changes to:



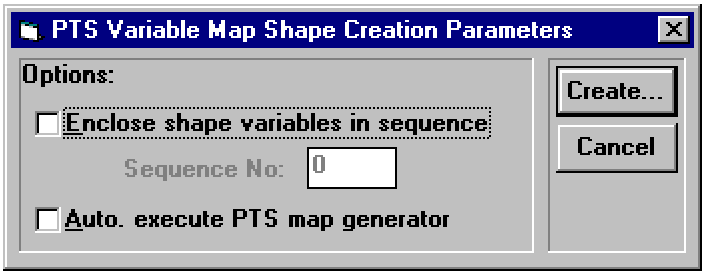
The calculated profile velocity shown in the information box will give the rate of machine cycles selected earlier in the Map Settings dialogue (see section 2.2.1 on page 10). The maximum profile velocity is 256 steps/sec. and the number of steps that you can allot for profile creation is limited so as not to exceed this.

Choose the Create button to continue and build the map table or profile, or the cancel button to abort the process.

NOTE:

- A map table/profile can only be downloaded into a PTS with a valid motion generator licence. The PTS Toolkit software will confirm this before proceeding with the operation: a dialogue box will be used to show the progress of this.

2.2.4 PTS Variable Map Shape Creation Parameters



When creating a PTS variable map shape for use by the PTS based motion generator the above dialogue box is presented. Variables cannot be stored in the nonvolatile memory of the PTS, and therefore if you want the map shape variables to be stored in the PTS they need to be written into a PTS sequence. You can specify the relevant sequence number. It is also possible to automatically execute the PTS based motion generator at the end of the map shape variable list.

Choose the Create button to continue and build the PTS variable map shape, or the cancel button to abort the process.

2.3 Mouse Actions

The motion generator package is designed to be used in an interactive manner and so the mouse can be used to modify and create the map shape:

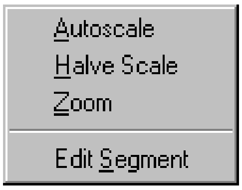
The left mouse button, when clicked, selects the segment nearest to the mouse pointer. This can be used to select which segment to edit using the Edit Segment dialogue box (see section 2.2.2 on page 13).

The left mouse button, when clicked on a node (the join between two segments) and held down, can be used to graphically adjust the segment. Nodes are displayed with either a ■ (can be edited using the mouse) or a ● (fixed). The same node, viewed in different derivatives, may be editable in one derivative and not in another. Note that segment editing with the mouse always uses the node at the end of the segment and affects the end constraints of that segment. It is not possible to graphically edit the start of segment constraints. All mouse editing is absolute: it is changing the end co-ordinates of the selected segment, and will therefore affect the shape of the selected segment and the next (higher) segment. [Note: if the next segment cannot accommodate the changes made using the mouse then a ripple through effect will be seen until a segment which can accommodate this change is reached.]

The left mouse button, when clicked away from the map shape and held down, performs a pan action. The map shape is scrolled along in the direction of the line created by moving the mouse, and to the extent given by the length of the line.

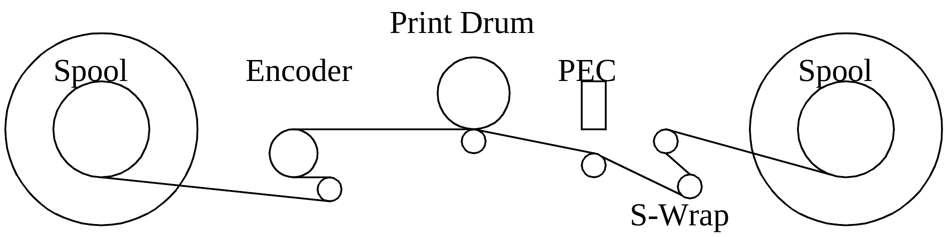
The left mouse button is also used when the zoom action is requested. Press the left mouse button and drag a rectangle over the area of the curve you wish to enlarge.

The right mouse button, when clicked, produces the following menu. This contains some frequently used commands for the motion generator.



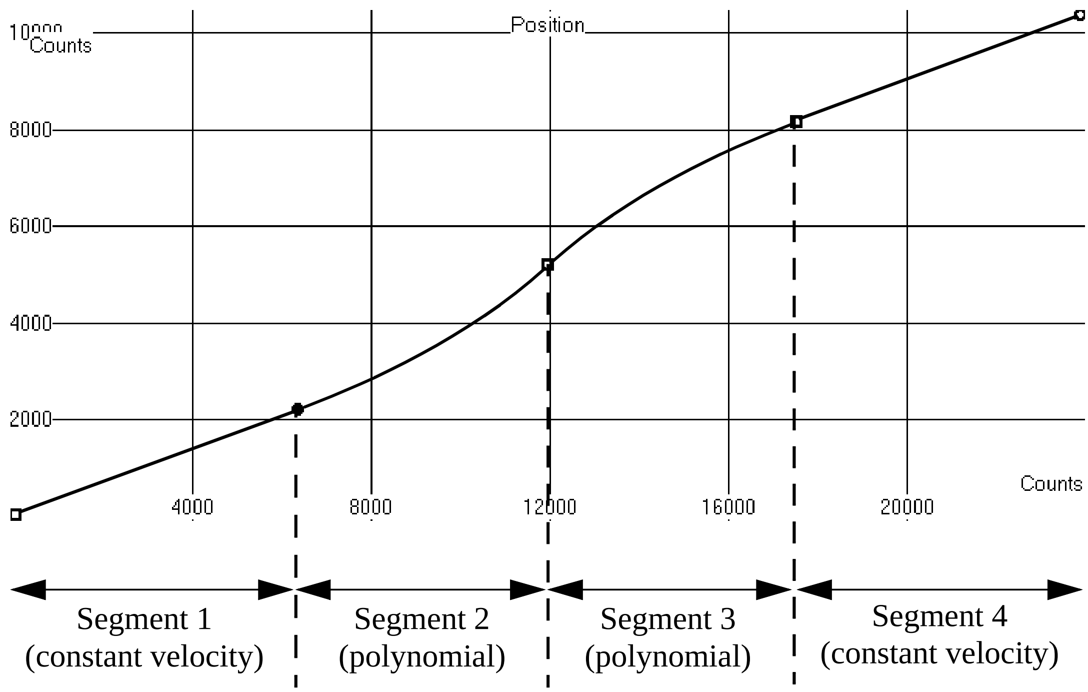
2.4 Example of Using the PC based PTS Toolkit Motion Generator

As an example, we may take a printing system where we have a web being drawn through a machine under tension control. It is required to print on the web in registration to existing marks, using a drum whose circumference is not necessarily the same as the register length. The speed of the print drum needs to match the speed of the web for part of its cycle, and to pause or catch up during the rest of the cycle. The 'contact' portion of the cycle is to be equi-spaced about the registration point (zero) of the web. Web travel is measured by an encoder, and drum rotation by the encoder simulation of the motor drive.



A typical mapping for the print drum motion (slave, Y axis) against the web encoder (master, X axis) is:

Motion Generator displaying Position only



The graph is constructed from constant-velocity segments 1 and 4, to give contact printing either side of the master zero, and polynomial segments 2 and 3 to link these for catch-up. The master length is 23874 counts (300 mm at 79.58 counts per mm); the contact length of this, 80mm either side of zero, is from 17508, through zero, to 6366. The print drum circumference is 374mm, and the encoder pulses are at 27.76 per mm: thus its cycle is 10382 counts, and the printing is from 8155 through zero to 2221.

Most of the definition of this graph is relatively straight forward:

- Open a new map, with master bounds 23874. Number it, e.g. 73, and choose the number of steps, typically 400. Set a typical running speed, say 60/minute, and click 'OK'.

- Split it into segments, approximately, using the mouse.
- Use Segment Edit to choose the required segment types and master axis breakpoints on a first pass, and to enter key slave axis coordinates on a second pass.
- Calculate the speed for the constant velocity segments (here 0.349 of master axis, or 8328 counts/sec at 1 cycle/sec) and set it as the end speed of segment 3 (which controls 4 and 1). Set speed match from 3 start to 2, and leave 2 end speed unspecified, acceleration 0.
- Check that the picture looks correct, and that the coordinates read correctly in Segment Edit. Close Segment Edit, and save the shape as a .shp file.
- For testing, it is probably best first to download to a PTS as a Map Table. The map can then be executed using two axes in virtual mode, and examined using the Scope facility. For example:

  CH1/VM1/SB23874/SV23874/SA30000/DC30000/PC/VC+
  CH2/VM1/SB10376/ML1/MW01000001/XM73

  The Scope might well be used to show channel 1 demand position, channel 2 demand position, and channel 2 demand speed. On-line mode would give a first check; historical mode would also allow display of channel 2 traces against the X-axis from channel 1.
- For more flexibility of use, the shape definitions can be downloaded to a PTS as a sequence to be called for on-line map generation. Alternatively the shape can be saved as a file; edited to add the ability to take in user revision of values (such as registration or printing lengths); then downloaded and saved in the PTS for production use. See section 3.3 on page 25 which covers these aspects.

2.5 Downloading and Using the map information in the PTS

After designing the map shape the information needs transferring to the PTS and using to map a slave axis to a master axis. There are two methods of transferring the information, either as a map table or a map shape:

2.5.1 As a Map Table

Download the map table to the PTS using the Download Table menu option from the File menu.

Once a map has been downloaded into a PTS the following procedure will let you use the master/slave position mapping (all commands should be typed into the terminal window of PTS Toolkit):

- Is the map a ‘rotary’ or a ‘linear’? If the map is a ‘rotary’ i.e. the slave end position co-ordinate is different to the slave start position co-ordinate then you need to set the **SB** command on the slave axis:
  CH<slave axis channel number>
  SB<value>
  The value that the SB command should be set to is the end value of the slave axis in encoder counts (which must also be the highest value). The SB command tells the PTS to wrap the slave axis round at ±SB (wraps to zero).
- On the master axis set **SB** to the appropriate value (as was given in the table creation dialogue box) in encoder counts:
  CH<master axis channel number>
  SB<value>
- Position both the slave and master axes at their start positions. This might involve referencing or simply a move to an absolute encoder position.
- Link the slave axis to the master axis using the **ML** command to tell the PTS which channel to map to which:
  CH<slave axis channel number>
  ML<master axis channel number>
- Execute mapping on the slave axis using the **XM** command:
  CH<slave axis channel number>
  PC
  XM<map table number>
- Start the master motor moving at the appropriate speed, for example using the **PC** and **VC** commands.

The slave motor will then map to the master motor, following the map shape as designed. One way of confirming this is to use PTS ’Scope to plot the slave axis position (or velocity) information and compare the result with the designed map shape.

All these commands can be written into sequences, and combined with other machine start commands that are required, to automate the process.

2.5.2 As a Map Shape

The map shape can be downloaded to the PTS in the form of PTS variables defining the map shape. This requires the PTS based motion generator to process these variables to create the appropriate map table. As will be explained in chapter 3. on page 23 this method is appropriate when the map shape is further configured to interact with the PTS user panel (for example) to provide operator definable parameters. For more information on this process refer to chapter 3. on page 23.

3. PTS based Motion Generator

The PTS based motion generator is a calculation engine that can process a map shape to produce a map table. The map shape is defined using specific PTS variables which have set meanings and scalings (see chapter 4. on page 27 for a full list of segments, variables and scalings, and chapter 5. on page 38 for a list of further global variables used by the motion generator, plus a summary of the segment specific variables). The PTS based motion generator reads these variables and then creates the appropriate map table.

The PC based PTS Toolkit motion generator should be used to design the map shape initially. A PTS variable map shape data file can then be created and downloaded into the PTS. If the map shape is to remain in nonvolatile memory on the PTS it needs to be enclosed in a sequence, and the PC based PTS Toolkit motion generator will prompt for a sequence number.

Once the sequence/PTS variables are present in the PTS the motion generator needs to be instructed to create the appropriate map table. This is done using the PTS variable $MTR=1 definition (often the last line of the sequence containing the map shape variables). A message will be given indicating that the motion generator has started and is processing the map shape information. Success or failure is indicated by further messages. Once the motion generator has finished the new map table will exist.

The most powerful feature of the PTS based motion generator is the ability to modify the map shape using operator specified values or dimensions, either input via the Quin Operators Panel or from a SCADA package. Because the map shape is defined as variables it is possible to write sequences which modify the values of these variables; therefore modifying the map shape. This allows the operator to, for example, specify the product length, and then the motion generator can recalculate the map table to suit. The example given in this chapter explains this process and the programming involved.

3.1 Commands required to operate the Motion Generator

The PTS based motion generator receives one command only, and this is done via the PTS variable $MTR. This is a trigger variable and tells the motion generator to read the map shape from the relevant PTS variables and generate the appropriate table:

- $MTR=1 will tell the motion generator to generate a map table.
- $MTR=2 will tell the motion generator to generate a profile table.

The variable $MST is used to return success/failure of the motion generator. $MST=0 indicates a success, error codes are listed in the next section and in Appendix A. on page 44.

For testing purposes $MTR=0 will provide a list of the PTS variables that the motion generator finds in the PTS without actually compiling and creating the map/profile table. This list can then be copied into a file on the PC and loaded into the PTS Toolkit motion generator via the import PTS file menu option.

3.2 Information Messages and Error Codes

The motion generator embedded in the PTS reports its progress and any errors/ warnings by using message strings that will appear on a terminal connected to the serial port and error numbers which will be placed in the $MST variable. The following two messages will be generated on a terminal connected to the serial port of the PTS during a successful execution of the motion generator:

Map *(Profile)* *x*: Generation Started

Generation of *x* segment map (*profile*) complete

Where *x* is the map number.

A number of error messages can occur due to incorrect map shape data, and these are split into two groups:

- Interpreter Errors: A problem exists with the variables being used for the map shape and the interpreter is unable to continue. These errors, along with explanations, are listed in this section.
- Calculation Engine Errors: The calculation of the map cannot continue because of a problem with the map shape. These errors are listed in Appendix A. on page 44.

Whatever the source of the error, the error number will be returned in the $MST variable and the error message will be sent to a terminal connected to the serial port.

Error Number	Error Message	Explanation
11	Error: No master start position	The variable $M00 must be defined for a map
11	Error: $M01 required	The variable $M01 must be defined for a map
12	Error: No slave start position	The variable $S00 must be defined for a map
13	Error: can't read map number	The variable $NUM must be defined, range 1 to 255 inclusive
13	Error: map number out of range	The value of $NUM must be in the range 1 to 255 inclusive
14	Error: can't read number of steps	The variable $NPT must be defined. [range 3 to 2400 inclusive: any other value will be saturated at these values]

Table 3: Embedded Motion Generator Error Codes

Error Number	Error Message	Explanation
15	Insufficient free memory for Motion Generator	The quantity of memory required to generate a map depends upon the number of segments and their types. The quantity of memory required to store a map table depends upon the length of the table
16	Error during map (*profile*) transfer	The motion generator was unable to transfer the map (or profile). This would occur when the previous version of the map is being used by the PTS
17	LT*nn*: Position list is undefined (*time*)	A position list is required for segment *nn* (PTS commands ET and LT enter and list position lists)
18	Error: Master axis is too long *or* Error: Map table entry too large	Position greater than ±4194303, either for master axis or for individual slave table entry
19	Error: Map step too large	The map step is determined by: $\text{Map Step} = \frac{\text{length of master axis}}{\text{\$NPT}}$ Map step (MS parameter) can never exceed 65535

Table 3: Embedded Motion Generator Error Codes

3.3 Example of Using the PTS based Motion Generator

This is the same example as described in section 2.4 on page 19. The on-line map generation using the Motion Generator within the PTS is required to produce maps based on operator-entered values; in this example, using new print and registration lengths. The example assumes that the cutoff lengths are entered via the Operator Panel, into variables $LEN for print length (in mm.) and $REG for registration distance (again, in mm.). The entered values need to be scaled for use, using further variables $ECM for web encoder counts per mm x 100, and $PCM for print drum encoder; these values would generally have been preset and saved using the panel. The example uses one further variable, $MSG, to select the displayed message on the panel. The code to start the map generation would normally be encapsulated in a sequence, saved and retrieved from PTS nonvolatile memory. This sequence could be based on that saved from the PC motion generator: a typical example follows, and would be called after recall or entry of the following variables:

```
$reg=300                      # Registration distance, mm
$len=150                      # Printing length, mm
$ecm=7958                     # Web encoder counts/mm x 100
$pcm=2776                     # Print drum counts/mm x 100
# Sequence to start map calculation
ES10
$MSG=3                        # "Calculating print drum profile"
$M00=0                        # Master start position
$M01=((($LEN/2+5)*$ECM)/100)  # Half print distance (+5mm window)
$M02=($REG*$ECM/200)          # Half registration distance
$M04=($REG*$ECM/100)          # Registration distance
$M03=($M04-$M01)              # Start of print (+5mm window)
$M05=0                        # End of map definition
$S00=0                        # Slave start position
$S01=((($LEN/2+5)*$PCM)/100)  # Half print distance (+5mm window)
$S04=(374*$PCM/100)           # Bound value
$S02=($S04/2)                 # Half bound value
$S03=($S04-$S01)              # Start of print
$W02=32201010                 # Segment 2 end pos. & accel. given
$W03=32201110                 # Segment 3 end pos., vel., & accel.
$Y02=0                        # End accel. 2
$X03=($S01*100000/$M01*10)    # End 3 vel.: ratio *10^6 (sums within
                              numeral range)
$Y03=0                        # End accel. 3
$F01=1                        # Constant speed for printing
$F02=9                        # Polynomial for limp
$F03=9                        # Polynomial for limp
$F04=1                        # Constant speed for printing
$NUM=73                       # Map number
$NPT=400                      # Number of points
CH2/ST                        # Make sure map isn't being used
$MST>                         # Undefine to allow preset value to
                              be given
$MST=-1                       # Preset to -1 prior to running
                              motion generator
$MST>XS201                    # Continuation after motion
                              generation
$MTR=1                        # Trigger motion generation
```

Note that variable $MST has been set up as a trigger variable to run another sequence when the map generator indicates completion status. This sequence would select a new message for panel display, flagging any error, or confirming completion and proceeding with further machine start-up tasks. The remainder of this application is given as an example and fully described in the Quin PTS User's Guide (Revision 2 or later).

The above PTS code can also be imported into the PC based PTS Toolkit motion generator using the File, Import PTS File menu option. All the equations will be interpreted as required, unnecessary lines discarded and any missing variables prompted for. The map shape seen on the motion generator would then be the same as was designed in section 2.4 on page 19.

4. Detail of map segment types

The following segment types are available in the motion generator (the index number refers to the value that would be assigned to the PTS variable $F*nn*):

0. Constant Position
1. Constant Velocity
2. Constant Acceleration
3. Sine-squared Velocity
4. Cycloidal
5. Modified Trapezoidal
6. Modified Sine
7. Triple Harmonic
8. Sinusoidal
9. Polynomial
10. Ramp
11. Throw
12. Position List
13. Quadratic Spline
14. Cubic Spline
15. Sine-constant-cosine
16. Simple Harmonic

In the following sections each segment type is listed, along with a description of the shape of the segment and guidelines for its use. Also included are details about what parameters can be altered, both in the PTS toolkit motion generator, and in the PTS map shape language. Where units are given for a parameter these apply to the PTS based motion generator only, the units used in the PC based PTS Toolkit motion generator will be displayed in the relevant dialogue box and will be derived from the user specified scalings where appropriate.

This key is used:

Value of slave:

P: Position

V: Velocity

A: Acceleration

J: Jerk

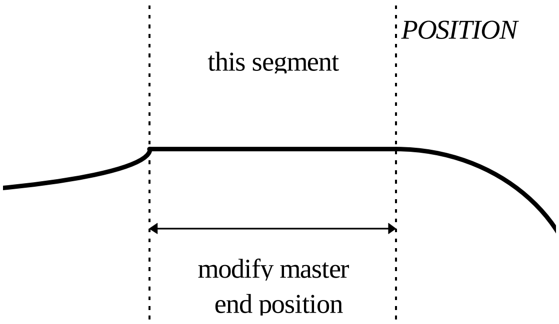
[…]: fixed parameter

$: PTS variable

nn: Segment number: 01 (1st) to 99 (last) inclusive

Note: $M*nn* and $S*nn* have units of encoder counts for the PTS based motion generator

4.1 Constant Position Segment

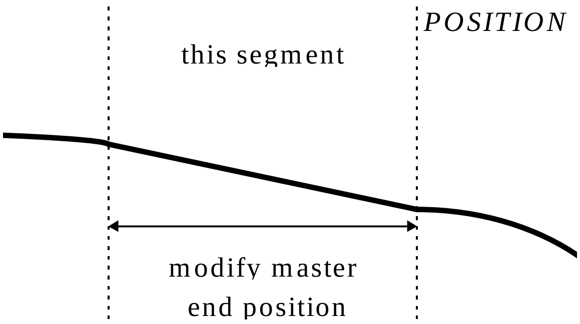


A polynomial segment with both the velocity and the acceleration fixed at zero. The slave position is maintained constant throughout at the end value of the previous segment. This means that only the position is matched from the previous segment, which can lead to an instantaneous change in velocity.

Parameters:

- Segment type $F*nn*=0
- [Segment start constraints: P matches, V, A & J unspecified]
- [Segment end constraints: P, V, A & J all unspecified]
- $W*nn*=20000000
- Master end position $M*nn*=<value> sets end segment master position co-ordinate
- Slave end position $S*nn*=<value> will be present in the PTS variable listing, but is ignored by the motion generator (and will be identical in value to $S*nn-1*)

4.2 Constant Velocity Segment

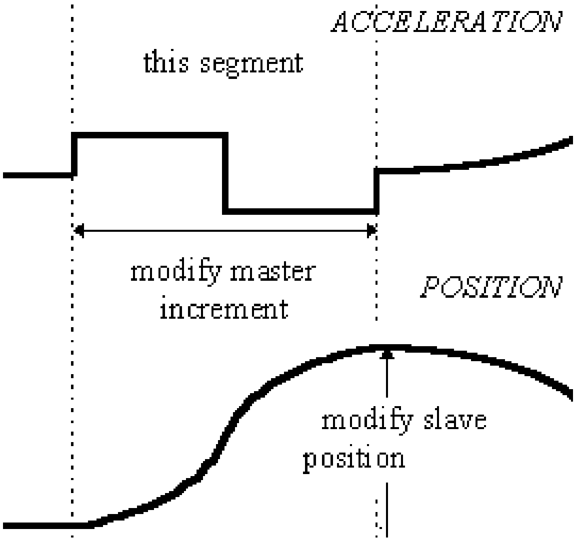


A polynomial segment with the acceleration fixed at zero. The velocity is constant throughout and is determined either by the end velocity of the previous segment or by a specified value. The slave position change is dependent upon both the velocity value for the segment and the value of the master increment.

Parameters:

- Segment type $F*nn*=1
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or specified ($W*nn*=21000000), [A & J unspecified]
- [Segment end constraints: P, V, A & J all unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> will be present in the PTS variable listing, but is ignored by the motion generator (value will be correct for map as saved)
- Start velocity $A*nn*=<value> sets start velocity if required by $W*nn*. Scaling of $A*nn* is as per a polynomial segment (see section 4.9 on page 32).

4.3 Constant Acceleration (& Constant Retardation) Segment



A segment with the acceleration fixed at a constant value. For the first half of the segment the acceleration is positive; for the second half it is negative. The difference between start and end positions determines the value of the acceleration and deceleration.

A more detailed name for this segment would be 'constant acceleration followed by constant retardation'.

Parameters:

- Segment type $F*nn*=2
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Percentage of segment used: start at $X*nn*=<value> & end at $Y*nn*=<value> (see section 5.1 on page 38 for scaling information)

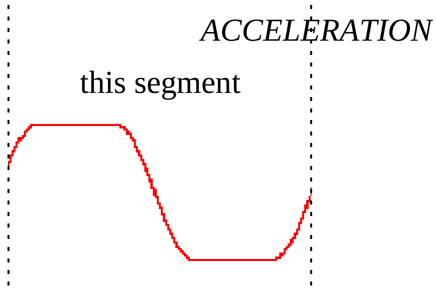
4.4 Sine-Squared Velocity / Cycloidal Segments

Mathematically both the Sine-Squared Velocity segment type and the Cycloidal segment type are identical. The acceleration profile is one cycle of a sine wave; the length is set by the master position increment and the magnitude is set by the slave position change across the segment.

Parameters:

- Segment type $F*nn*=3 for Sine-Squared or $F*nn*=4 for cycloidal
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20222222), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Percentage of segment used: start at $X*nn*=<value> & end at $Y*nn*=<value> (see section 5.1 on page 38 for scale information)

4.5 Modified Trapezoidal Segment

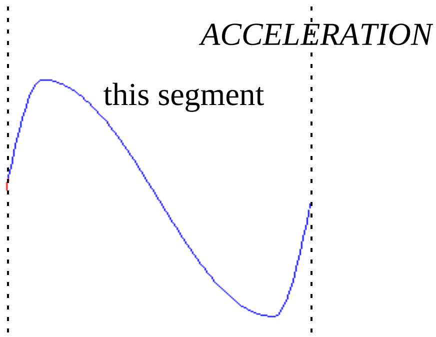


The acceleration profile as fixed to the shape shown in the diagram, however its magnitude is set by the master position increment and the slave position change across the segment. This type of segment is useful for heavy, stiff systems running at low or medium speeds.

Parameters:

- Segment type $F*nn*=5
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Percentage of segment used: start at $X*nn*=<value> & end at $Y*nn*=<value> (see section 5.1 on page 38 for scaling information).

4.6 Modified Sine Segment



The acceleration profile is fixed to the shape shown in the diagram, however its magnitude is set by the master position increment and the slave position change across the segment. This type of segment is useful for systems with some backlash and for medium or high speeds.

Parameters:

- Segment type $F*nn*=6
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Percentage of segment used: start at $X*nn*=<value> & end at $Y*nn*=<value> (see section 5.1 on page 38 for scaling information)

4.7 Triple Harmonic Segment

This segment will always start and finish with zero velocity and acceleration (assuming velocity matching has been switched off for this segment). The acceleration profile throughout the segment is defined by the parameters required: three normalised amplitudes for the harmonics used to construct the curve. These normalised values form the three coefficients of a three term fourier series. One example is: set the first harmonic to 2π, and both the second and third harmonics to zero; the acceleration profile will then be identical to a cycloidal segment. However you only have to specify the first two of these as the third is calculated to satisfy the constraints placed on the segment from the neighbouring segments.

Parameters:

- Segment type \$F*nn*=7
- Segment start constraints: [P matching], V matching (\$W*nn*=22000000) or unspecified (\$W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position \$M*nn*=<value> sets segment end master position co-ordinate
- Slave end position \$S*nn*=<value> sets segment end slave position co-ordinate
- Amplitudes of the three harmonics \$A*nn*=<value>, \$B*nn*=<value> and \$C*nn*=<value> in normalised units of a three term fourier series x10^6

4.8 Sinusoidal Segment

This segment is a sine wave in master/slave position. You can set the start position (in terms of degrees) along the sine shape, the amplitude (in slave axis units) and the number of cycles of the sine wave used for the segment. These three parameters determine the slave end position of this segment and the velocities and accelerations within the segment. Due to the default velocity matching you will often see an ‘S’ curve in position by default, rather than a sine wave. This is because a position sine wave starts with non-zero velocity and the majority of other segments end, by default, with zero velocity.

Parameters:

- Segment type \$F*nn*=8
- Segment start constraints: [P matching], V matching (\$W*nn*=22000000) or unspecified (\$W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position \$M*nn*=<value> sets segment end master position co-ordinate
- Slave end position \$S*nn*=<value> will be present in the PTS listing but will be ignored by the motion generator (value will be correct as saved)
- Cycle start angle \$A*nn*=<value> in degrees x10^6
- Amplitude of sine wave \$B*nn*=<value> in encoder counts
- Number of cycles of sine wave \$C*nn*=<value> (no units x10^6)

4.9 Polynomial Segment

This is a general purpose segment shape and is the most adjustable segment and as such is the default and recommended segment type. A polynomial segment can be thought of as a shape which fits various boundary constraints. You can specify the various settings required for the segment using the mouse and the segment settings dialogue box for typed entry (see section 2.2.2 on page 13).

Parameters:

- Segment type $F*nn*=9
- Segment start and end constraints: the polynomial segment type determines its shape by applying boundary constraints. The constraint word $W*nn* is used for setting these constraints and has the following format:

  A eight digit decimal number which determines the constraints for each of the four derivatives at both ends of the segment. Each digit represents a boundary constraint, and can have one of four values:

  0 unspecified: this boundary constraint will not be applied.

  1 specified: this boundary constraint will be specified by the relevant additional variable (see below).

  2 match: [only applies to start of segment boundary constraints] indicates that this boundary constraint is specified and that the value should match equivalent end boundary value of the previous segment.

  3 don't update this boundary constraint (a don't care option for updating certain boundary constraints only).

$W*nn* Entry	**Segment boundary**	**Boundary constraint**	**Default value**
viii		position	2 (match) *cannot be changed*
vii	start	velocity	2 (match)
vi		acceleration	2 (match)
v		jerk	0 (unspecified)
iv		position	1 (specified) *cannot be changed*
iii	end	velocity	1 (specified)
ii		acceleration	1 (specified)
i		jerk	0 (unspecified)

Table 4: Constraint Word Interpretation

Therefore the default constraint word for a polynomial segment is $W*nn*=22201110

Example: $W*nn*=21101110 indicates that the start velocity and acceleration, along with the end velocity and acceleration have been specified; therefore variables $A*nn*, $B*nn*, $X*nn* and $Y*nn* would need to be present for this segment.

Note: the format of this constraint word applies to all segments, though different segments have different limitations upon which boundary constraints can be altered.

- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Derivative specification parameters: used when the segment constraint specifies that the particular derivative is specified:

Segment boundary	Derivative	Variable Name	Units: Slave encoder counts per
start	Velocity	$A*nn*	$1x10^6$ master encoder counts
	Acceleration	$B*nn*	$1x10^8$ master encoder counts^2
	Jerk	$C*nn*	$1x10^9$ master encoder counts^3
end	Velocity	$X*nn*	$1x10^6$ master encoder counts
	Acceleration	$Y*nn*	$1x10^8$ master encoder counts^2
	Jerk	$Z*nn*	$1x10^9$ master encoder counts^3

Table 5: Polynomial Derivative Parameters

Examples; For every master position count:

$A*nn*=1000000 means that the slave position increases by 1 count.

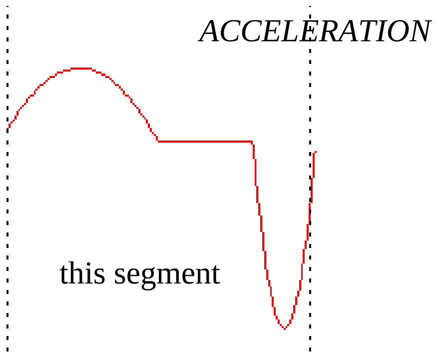
$A*nn*=-24000000 means that the slave position decreases by 24 counts.

$A*nn*=50000 means that the slave position increases by 0.05 counts.

$B*nn*=50000000 means that the slave velocity increases by 0.5 counts per master count (equivalent to a increase in $A*nn* of 500000).

$Z*nn*=-2000000000 means that the slave acceleration decreases by 2 counts per master count (equivalent to a decrease in $Y*nn* of 200000000).

4.10 Ramp Segment



The acceleration profile of a ramp segment is fixed to the shape shown (which consists of two sinusoidal portions joined by a zero acceleration section); however the width of the acceleration and deceleration pulses can be changed (here set to 50% and 20% respectively).

Parameters:

- Segment type $F*nn*=10
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Length of initial acceleration pulse $A*nn*=<value> and of final deceleration pulse $B*nn*=<value> both in percent x10^6 (range 0 to 100% inclusive; and $A*nn*+$B*nn*≤100% always)

4.11 Throw Segment

A throw segment gives a smooth motion starting and finishing with zero velocity and acceleration, with an overall slave position change of zero. The magnitude of the position change at the midpoint of the segment is definable. The segment is symmetrical about the midpoint and should be used in preference to two sequential ramp segments as the acceleration profile is superior.

Parameters:

- Segment type $F*nn*=11
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P, V, A & J all unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> will be present in the PTS variable listing, but is ignored by the motion generator (and will be identical in value to $S*nn-1*)
- Extent of throw $A*nn*=<value> in slave position units (encoder counts for PTS)

4.12 Position List Segment

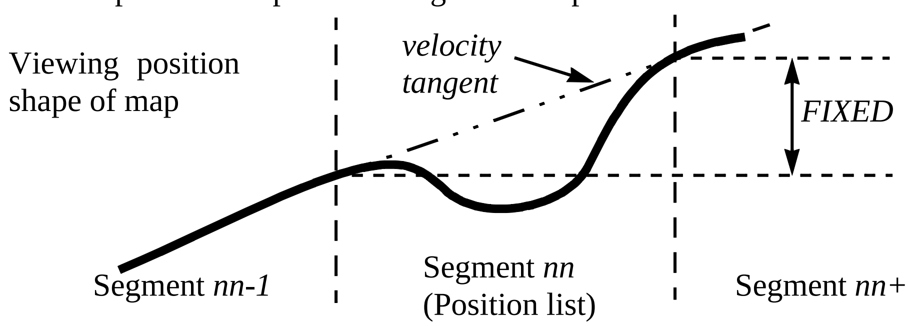
This is the facility that enables you to enter your own data to define the shape of a segment. Any position list segment requires a data file (normally with the extension '.SEG') containing values defining the shape. The format of this file is a plain text file (ASCII) (use PTS Note or NOTEPAD for example) with the following entries:

```
<start of file>
4   <first line defines the number of points in the list
(excluding first)>
0   <first slave position: should always be zero>
10  <second slave position, 10 units>
5
15
20  <last slave position>
<end of file>
```

You can have up to 360 entries in a position list. The motion generator then uses the list to create a smooth segment shape given the master (and slave) position change. For the PC based PTS Toolkit motion generator the position list will be assumed to be in the units that you are currently working in if there is non overall change in position (i.e. first and last values of the position list are identical), or otherwise will be normalised and scaled to suit the required slave position change across the segment. The PTS based motion generator stores position lists using the ET*nn* command (where *nn* is the segment number) using the same format as above, and these lists are held in encoder counts if there is no overall change in position, or otherwise in normalised units which are scaled to suit the required change in slave position.

Parameters:

- Segment type $F*nn*=12
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P, V, A & J all unspecified unless position list has net change in which case P can be specified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate *except* for two specific cases:
  1) the position list starts and ends at the *same* slave co-ordinate
  2) velocity matching is *on* at the start of the segment *and* the velocity tangent at the start of the position list passes through the end position co-ordinate of the list:



- File name for position list (PC based motion generator, and only required once)

4.13 Quadratic Spline Segment

A quadratic spline is a special case of a polynomial segment. It has continuity of both position and velocity from the previous segment at the beginning of the curve. The length and width of the segment can be changed. Having a number of these segments in series performs a quadratic spline between the data points (end segment co-ordinates).

Parameters:

- Segment type $F*nn*=13
- [Segment start constraints: P & V matching, A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate

4.14 Cubic Spline Segment

A cubic spline is a special case of a polynomial segment. It has continuity of position, velocity and acceleration from the previous segment at the beginning of the curve. The length and width of the segment can be changed. Having a number of these segments in series performs a cubic spline between the data points (end segment co-ordinates).

Parameters:

- Segment type $F*nn*=14
- [Segment start constraints: [P, V & A matching, J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate

4.15 Sine-Constant-Cosine Segment

This is a very adjustable segment and can often be used in place of other segments which have sine and cosine acceleration profiles. The acceleration profile is fixed to a sine wave followed by a constant portion followed by a cosine wave. It is however possible to modify the percentage of the segment taken by each of these portions, so creating quite complex shapes. The magnitude of the acceleration is set by the master position increment and the slave position change.

Parameters:

- Segment type $F*nn*=15
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate

- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Length of first portion of the segment given to sine acceleration $A*nn*=<value> in percent x10^6 (range 0 to 100% inclusive)
- Length of second portion of the segment given to constant acceleration $B*nn*=<value> in percent x10^6 (range 0 to 100% inclusive)
- Length of the third portion of the segment given to cosine acceleration $C*nn*=<value> in percent x10^6 (range 0 to 100% inclusive)
- Percentage of segment used: start at $X*nn*=<value> & end at $Y*nn*=<value> (see section 5.1 on page 38 for scaling information)

Note that $A*nn* + $B*nn* + $C*nn* = 100% always.

4.16 Simple Harmonic Segment

Has an acceleration profile of the first half of a cosine wave. The magnitude of the acceleration profile is set by the master position increment and the slave position change. It is also possible to use only part of the segment, using the start and end percentages. This type of segment is useful for systems with some backlash at low to medium speeds.

Parameters:

- Segment type $F*nn*=16
- Segment start constraints: [P matching], V matching ($W*nn*=22000000) or unspecified ($W*nn*=20000000), [A & J unspecified]
- [Segment end constraints: P specified, V, A & J unspecified]
- Master end position $M*nn*=<value> sets segment end master position co-ordinate
- Slave end position $S*nn*=<value> sets segment end slave position co-ordinate
- Percentage of segment used: start at $X*nn*=<value> & end at $Y*nn*=<value> (see section 5.1 on page 38 for scaling information)

5. Motion Generator Features and Functions

This section contains detailed explanations of certain features of the motion generator, and some summaries of the information contained in the previous section, 'Detail of map segment types'.

5.1 Start and End Percentages of a Segment

For the following segment types it is possible to specify what proportion of the segment shape is used for the segment:

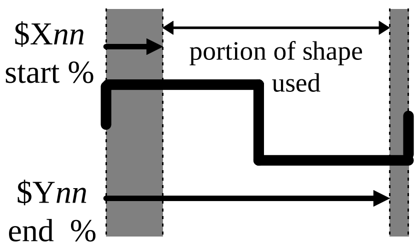
2: Constant Acceleration	6: Modified Sine
3: Sine-squared Velocity	10: Ramp
4: Cycloidal	15: Sine-constant-cosine
5: Modified Trapezoidal	16:Simple Harmonic

This is done using PTS variables $X*nn* and $Y*nn* for the PTS based motion generator and using the segment type tab of the segment settings dialogue box for the PC based PTS Toolkit motion generator.

Scalings:

- PC based: Percentage required, entered via the segment type tab of the segment settings dialogue box, as a real number
- PTS based: Percentage required x10^6, entered into $X*nn* and $Y*nn*, n.b. $X*nn*+$Y*nn*≤100%

The following diagram illustrates the effect of setting both the start and end percentages to non-zero values for the constant acceleration segment shape.



By adjusting the start and end percentage it is possible to create specialist shape profiles, for example a modified sine profile with no deceleration and finishing with a non-zero acceleration.

5.2 Summary of Segment Constraints

Constraint	Segment Types
P match **V A J** unspecified	0: Constant Position
P match **V** match *or* specified **A J** unspecified	1: Constant Velocity
P V match **A J** unspecified	13: Quadratic Spline
P V A match **J** unspecified	14: Cubic Spline
P match **V** match *or* unspecified **A J** unspecified	2: Constant Acceleration 3: Sine Squared Velocity 4: Cycloidal 5: Modified Trapezoidal 6: Modified Sine 7: Triple Harmonic 8: Sinusoidal 10: Ramp 11: Throw 12: Position List 15: Sine Constant Cosine 16: Simple Harmonic
P match **V A J** match *or* specified *or* unspecified	9: Polynomial

Table 6: Segment Start Constraints

- Match means that the derivative value doesn't change from the end of one segment to the beginning of the next.
- Specified means that the derivative value can be set at that point on the segment.
- Unspecified means that the derivative value cannot or has not been altered at that point on the segment.

Constraint	Segment Types
P V A J unspecified	0: Constant Position 1: Constant Velocity 8: Sinusoidal 11: Throw 12: Position list *
P specified **V A J** unspecified	2: Constant Acceleration 3: Sine Squared Velocity 4: Cycloidal 5: Modified Trapezoidal 6: Modified sine 7: Triple Harmonic 10: Ramp 13: Quadratic Spline 14: Cubic Spline 15: Sine Constant Cosine 16: Simple Harmonic
P specified **V A J** specified *or* unspecified	9: Polynomial

Table 7: Segment End Constraints

* **P** can be specified if position list has net slave position change.

Key (for both tables)

P: Position **V**: Velocity

A: Acceleration **J**: Jerk

5.3 Summary of Segments and their Parameters

The following table summaries the PTS variable names used for each segment type, and, where applicable, gives a brief description of the meaning of the variable:

Segment Type	$M*nn*	$S*nn*	$F*nn*	$A*nn*	$B*nn*	$C*nn*	$X*nn*	$Y*nn*	$Z*nn*	$W*nn*
Constant Position	✓	×	0							
Constant Velocity	✓	×	1	Start Velocity*						✓
Constant Acceleration	✓	✓	2				Start %	End %		✓
Sine Squared Velocity	✓	✓	3				Start %	End %		✓
Cycloidal	✓	✓	4				Start %	End %		✓
Modified Trapezoidal	✓	✓	5				Start %	End %		✓
Modified Sine	✓	✓	6				Start %	End %		✓
Triple Harmonic	✓	✓	7	1st harmonic amplitude	2nd harmonic amplitude	3rd harmonic amplitude				✓
Sinusoidal	✓	×	8	Cycle start angle	Amplitude	No. of cycles				✓
Polynomial	✓	✓	9	Start Velocity	Start Accel.	Start Jerk	End Vel.	End Accel.	End Jerk	✓
Ramp	✓	✓	10	Start %	End %					✓
Throw	✓	✓	11	Extent of throw						✓
Position List	✓	✓	12							✓
Quadratic Spline	✓	✓	13							
Cubic Spline	✓	✓	14							
Sine Constant Cosine	✓	✓	15	Sine %	Constant %	Cosine %	Start %	End %		✓
Simple Harmonic	✓	✓	16				Start %	End %		✓

Key:

✓	standard definition applies
×	although variable may be present (and standard definition applies) it will not be used
<blank>	the variable does not apply to the segment type
<text>	brief description of the meaning of the variable specific to the segment type
*	Start velocity will only be required for a constant velocity segment if the velocity matching at start is switched off

5.4 Technical Information about the Motion Generator

This section details some technical information about the motion generator.

5.4.1 Maximum number of segments

PC based PTS Toolkit motion generator: limited by memory, but practical limits will also apply. PTS based motion generator: 99 (01 to 99 inclusive).

5.4.2 PTS variable names used (PTS based motion generator specific)

Map start co-ordinates:

- $M00: master position co-ordinate (range ±4194304, units encoder counts).
- $S00: slave position co-ordinate (range ±4194304, units encoder counts).

For each segment (where *nn* is 01 to 99 inclusive, with leading zero when appropriate):

- $F*nn*: segment type (range 0 to 16 inclusive, no units).
- $M*nn*: master position co-ordinate (range ±4194304, units encoder counts).
  NOTE: to end a map shape the motion generator looks for a value of $M*nn* that is *less* than the value of $M*nn-1*, and so calculates that the map shape has *nn-1* segments.
- $S*nn*: slave position co-ordinate (range ±4194304, units encoder counts).
- $W*nn*: constraint word for segment:

$W*nn* Entry	**Segment boundary**	**Boundary constraint**
viii	start	position
vii		velocity
vi		acceleration
v		jerk
iv	end	position
iii		velocity
ii		acceleration
i		jerk

Table 8: Constraint Word Entries

Where each entry can be 0: unspecified; 1: specified; 2: match (start boundary only); 3: don't care (uses previous, or default, value). Valid entries will depend upon segment type.

- $A*nn*, $B*nn*, $C*nn*, $X*nn*, $Y*nn* & $Z*nn*: meaning of variable is dependent upon segment type, as follows

Segment Number	Applicable Variables	Description
1	$A*nn*	Start velocity, required only when velocity matching at start is turned off using $W*nn*. See section 5.4.3 on page 43 for scaling
2	$X*nn* & $Y*nn*	Start & End percentages x10^6
3	$X*nn* & $Y*nn*	Start & End percentages x10^6
4	$X*nn* & $Y*nn*	Start & End percentages x10^6
5	$X*nn* & $Y*nn*	Start & End percentages x10^6
6	$X*nn* & $Y*nn*	Start & End percentages x10^6
7	$A*nn*, $B*nn* & $C*nn*	Normalised amplitudes of the three harmonics of a three term fourier series x10^6
8	$A*nn*, $B*nn* & $C*nn*	Cycle start angle (degrees x10^6), amplitude (encoder counts) & No. cycles (no units x10^6)
9	$A*nn*, $B*nn*, $C*nn*, $X*nn*, $Y*nn* & $Z*nn*	Start & End derivative values (when applicable as determined by $W*nn*). See section 5.4.3 on page 43 for scalings
10	$X*nn* & $Y*nn*	Start & End percentages x10^6
11	$A*nn*	Extent of throw (encoder counts)
12	$A*nn*	Slave end position (encoder counts)
15	$A*nn*, $B*nn*, $C*nn*, $X*nn* & $Y*nn*	Percentage x10^6 of sine, constant and cosine portion of segment. Start & End percentages x10^6
16	$X*nn* & $Y*nn*	Start & End percentages x10^6

Table 9: Segments and Applicable Variables

Global to motion generator:

- $NUM sets map number (range 1to 255 inclusive, no units). Note: maps 1 to 200 inclusive will be saved in PTS NVM, maps 201 to 255 inclusive are not.
- $NPT sets the number of points in the generated map table (max. 2400, no units).
- $MTR is the command word for the motion generator. $MTR=1 triggers the motion generator: the relevant variables will be read and the resultant map table produced. $MTR=2 repeats the same process but produces a profile table rather than a map table. $MTR=0 will provide a list of the PTS variables that the motion generator finds without actually compiling and creating the map/profile table.

- \$MST is the variable used to return the success or failure of the motion generator: 0 indicates success, a non-zero value will indicate failure (see section 3.2 on page 24 for a list of error codes).

5.4.3 Map scale units

Where applicable the map scale units for the PC based PTS Toolkit motion generator are:

- position, velocity, acceleration and jerk: user specified, defaults are encoder counts for positional information and encoder counts per second for velocity information. Acceleration and jerk are higher derivatives of the velocity units. Optionally user supplies conversion factors between user defined units and default units, as determined by the mechanical construction of the machine.

Where applicable the map scale units for the PTS based motion generator are:

- Position: encoder counts. Individual positions are specified in encoder counts, maximum range is from -4194304 to +4194304 encoder counts.
  Maximum length of a master axis is 4194304 encoder counts.
  Maximum length of slave axis is from -4194304 to +4194304 encoder counts
  Resolution is 1 encoder count for both the master and slave axes.
- Velocity: slave encoder counts per $1x10^6$ master counts. Maximum range is -2147483648 to +2147483648 which equates to -2147.48 to +2147.48 slave encoder counts per master count. Resolution is 0.000001 ($1x10^{-6}$) slave encoder counts per master count.
- Acceleration: slave encoder counts per $1x10^8$ master counts^2. Maximum range is -2147483648 to +2147483648 which equates to -21.4748 to +21.4748 slave encoder counts per master count^2. Resolution is 0.00000001 ($1x10^{-8}$) slave encoder counts per master count^2.
- Jerk: slave encoder counts per $1x10^9$ master counts^3. Maximum range is -2147483648 to +2147483648 which equates to -2.14748 to +2.14748 slave encoder counts per master count^3. Resolution is 0.000000001 ($1x10^{-9}$) slave encoder counts per master count^3.

Appendix A. Motion Generator Calculation Engine Error Codes

The following error codes can be generated by the motion generator calculation engine itself (either the PC or the PTS based version):

- PC motion generator: Standard Windows message box informing of the error.
- PTS embedded motion generator: error message format (following text sent through the serial port, <*number*> returned in variable $MST):

Error <*number*> [$<*variable*>=<*value*>]

The variable and value will help in determining the segment number and therefore the segment type and parameter that caused this error. Note that an error message may have more than one error number associated with it.

Error No.	Error Message (*PC based*)	Comments/ Information
501	master count out-of-bounds	
502	no segment with required master count	
503	append master count out-of-range	
504	insufficient memory for new segment	Working memory full (PC/PTS)
505	segment number too low	
506	insert segment width too small	
507	insert segment - insufficient memory	Working memory full (PC/PTS)
508	add segment not found	
509	delete segment not found	
510	delete segment - no current segment	
511	set master count - node number out-of-range	
512	set master count out-of-range	
513	negative node number	
514	node number too big	
515	negative node number	
516	node number too big	
517	set slave count - segment not found	
518	get slave count - negative node number	
519	get slave count - node number too big	
520	set Y segment number too low	Y refers to $Y*nn*
521	set Y segment number too big	
522	set Y segment not found	
523	set B segment number too low	B refers to $B*nn*
524	set B segment number too big	
525	set B segment not found	

Error No.	Error Message (*PC based*)	Comments/ Information
526	set Z segment number too low	Z refers to $Z*nn*
527	set Z segment number too big	
528	set Z segment not found	
529	set C segment number too low	C refers to $C*nn*
530	set C segment number too big	
531	set C segment not found	
532	set W segment number too low	W refers to $W*nn*
533	set W segment number too big	
534	set W segment not found	
535	set W segment number too low	
536	set W segment number too big	
537	set W segment not found	
538	set W segment number too low	
539	set W segment number too big	
540	set W segment not found	
541	set F segment number too low	F refers to $F*nn*
542	set F segment number too big	
543	set F segment not found	
544	set F function type not known	
545	set F out of memory	Working memory full (PC/PTS)
546	set spec segment number too low	
547	set spec segment number too big	
548	set spec segment not found	
549	get spec segment number too low	
550	get spec segment number too big	
551	get spec segment not found	
552	set siz value out-of-range	
553	set npts value out-of-range	npts refers to $NPT
554 *to* 556	number of steps too low	
557	set Pa segment number too big	Pa refers to $A*nn* parameter
558	set Pa segment not found	
559	set Pb segment number too low	Pb refers to $B*nn* parameter
560	set Pb segment number too big	
561	set Pb segment not found	
562	set Pc segment number too low	Pc refers to $C*nn* parameter
563	set Pc segment number too big	
564	set Pc segment not found	
565	set Pe segment number too low	Pe refers to $Y*nn* end %age
566	set Pe segment number too big	

Error No.	Error Message (*PC based*)	Comments/ Information
567	set Pe segment not found	
568	set Ps segment not found	Ps refers to $X*nn* start %age
569	set Ps segment number too big	
570	set Ps segment not found	
571	set throw segment number too low	
572	set throw segment number too big	
573	set throw not for this type of segment	
574	get throw segment number too low	
575	get throw segment number too big	
576	set throw not for this type of segment	
577	set ramp segment number too low	
578	set ramp segment number too big	
579	set ramp not for this type of segment	
580	get ramp segment number too low	
581	get ramp segment number too big	
582	get ramp not for this type of segment	
583	set triple segment number too low	
584	set triple segment number too big	
585	set triple not for this type of segment	
586	set sin segment number too low	
587	set sin segment number too big	
588	set sin not for this type of segment	
589	load points segment number too low	
590	load points segment number too big	
591	load points segment not found	
592	scale x number of steps too low	
593	get width segment not found	
594	get height segment not found	
595	new node number out-of-range	
596	new node master count out-of-range	
597	new node - out of memory	Working memory full (PC/PTS)
598	set T segment number too low	T is not used at present
599	set T segment number too big	
600	set T segment not found	
601	get T segment number too low	
602	get T segment number too big	
603	get T segment not found	
604	get W segment number too low	
605	get W segment number too big	
606	get X segment number too low	
607	get X segment number too big	

Error No.	Error Message (*PC based*)	Comments/ Information
608	get X segment not found	
609	set A segment number too low	A refers to $A*nn*
610	set A segment number too big	
611	set A segment not found	
612	get A segment number too low	
613	get A segment number too big	
614	get A segment not found	
615	set Y segment number too low	
616	set Y segment number too big	
617	set Y segment not found	
618	get Y segment number too low	
619	get Y segment number too big	
620	get Y segment not found	
621	set B segment number too low	
622	set B segment number too big	
623	set B segment not found	
624	get B segment number too low	
625	set B segment number too big	
626	set B segment not found	
627	set Z segment number too low	
628	set Z segment number too big	
629	set Z segment not found	
630	get Z node number too low	
631	get Z node number too big	
632	get Z segment not found	
633	set C segment number too low	
634	set C segment number too big	
635	set C segment not found	
636	get C segment number too low	
637	get C segment number too big	
638	get C segment not found	
639	get W segment number too low	
640	get W segment number too big	
641	get W segment not found	
642	get CW segment number too low	CW refers to $W*nn*
643	get CW segment number too big	
644	get CW segment not found	
645	get F segment number too low	
646	get F segment number too big	
647	get F segment not found	
648	node number negative	

Error No.	Error Message (*PC based*)	Comments/ Information
649	node number too big	
650 *to* 653	negative number of steps	
654	get Pabc segment number too low	Get is used by the PC version only
655	get Pabc segment number too big	
656	get Pabc segment not found	
657	get Pes segment number too low	
658	get Pes segment number too big	
659	get Pes segment not found	
660	get triple segment number too low	
661	get triple segment number too big	
662	get triple not for this type of segment	
663	get sin segment number too low	
664	get sine segment number too big	
665	get sin not for this type of segment	
666	set nupts segment number too low	nupts refers to $NPT
667	set nupts segment number too big	
668	set nupts segment not found	
669	set point segment number too low	
670	set point segment number too big	
671	set point segment not found	
672	invalid map number	
673	set X segment number too low	
674	set X segment not found	
675	set X segment number too low	
676	set X segment number too big	
677	set X segment number too low	
678	set X segment not found	
679	set A segment number too low	
680	set A segment number too big	
681	set A segment not found	
682	set X segment number too low	
683	set X segment number too big	
901	set width too low	
924	not allowed for this segment type	
925	set width too low	
926	crop ratio out-of-range	
927	unknown boundary condition	
928 *to* 930	uncompiled polynomial segment	

Error No.	Error Message (*PC based*)	Comments/ Information
931	uncompiled segment	
932		
933	SCCA set width too low	
934	SCCA segment split out-of-place	
935		
936 *to* 938	Pa,Pb,Pc parameter out-of-range	
939	unknown set Parameter operation	
940	unknown parameter type	
941	uncompiled segment	
942	CACR segment split out-of-place	
943		
944 *to* 951	segment split out-of-place	
952	boundary condition unacceptable	
953	segment split out-of-place	
954		
1001 *to* 1004	uncompiled segment	
1005	set width too low	
1006	crop not allowed for sinusoidal segment	
1007		
1008 *to* 1011	uncompiled segment	
1012	crop not allowed for ramp	
1013		
1014 *to* 1017	uncompiled ramp segment	
1018 *to* 1024	ramp parameters unacceptable	
1025	ramp set B parameter unacceptable	
1026	crop not allowed for throw segment	
1027		
1028 *to* 1031	uncompiled throw segment	
1032	set position list width too low	
1033	crop not allowed for position list	
1034		

Error No.	**Error Message (*PC based*)**	**Comments/ Information**
1035 *to* 1038	uncompiled position list segment	
1039	No. pts in table unacceptable	
1040	table index out-of-range	
1041	not allowed for position list segments	
1042 *to* 1044	uncompiled position list segment	
1045	boundary condition unacceptable for quad spline	
1046	boundary condition unacceptable for cubic spline	
1047 *to* 1050	no memory for table points	Working memory full (PC/PTS)

Q

R

S

T

U

