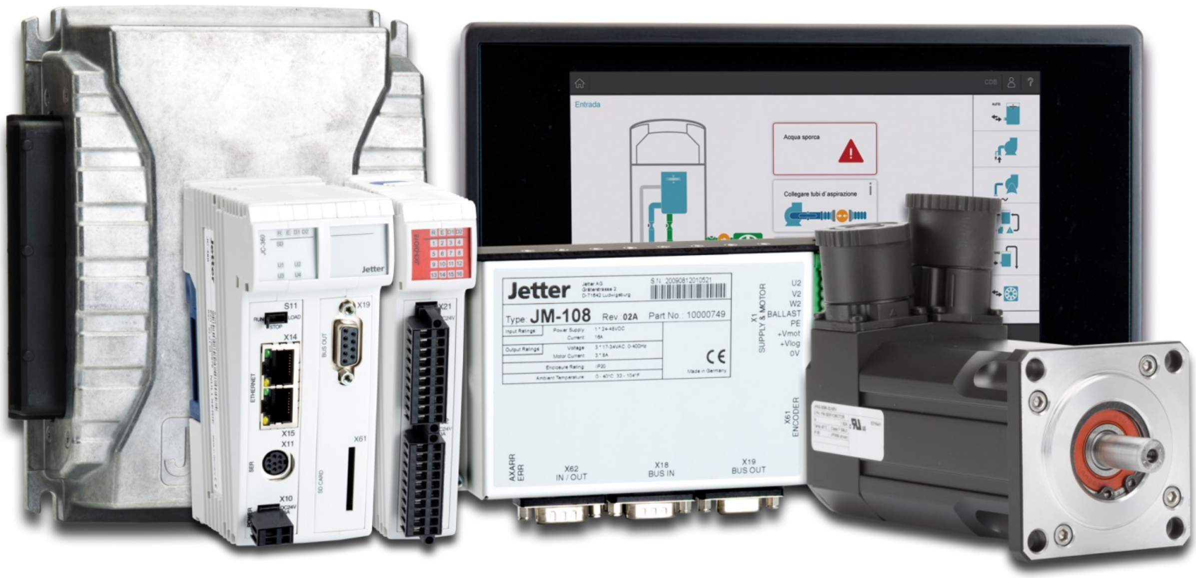
Jetter automation



Motion Control - Cam Definition

Application Note 052

608 847 55_00

We automate your success.

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg
Germany

www.jetter.de

Phone:

Switchboard	+49 7141 2550-0
Sales	+49 7141 2550-531
Technical hotline	+49 7141 2550-444

E-mail:	info@jetter.de
Technical hotline	hotline@jetter.de
Sales	vertrieb@jetter.de

Product name	Motion Control - Cam Definition
Document type	Application Note 052
Translation of the original German language document	
Document revision	1.00
Item number	608 847 55_00
Date of issue	2021-05-21

Table of Contents

1 Introduction

1.1 Prerequisites

- The following versions were used for the code and project examples as well as screen-shots:
    - JetSym 5.601
    - Motion API 1.0.0.16
    - JC365MC-OS 1.30.0.00
- Application Note 049: Motion Control - Technology Group

2 Cam Definition - Introduction

The following is shown in this application note:

- how a cam is basically defined;
- which elements a cam consists of;
- what types of segments are available;
- which options are available.

3 Cam

A cam is a motion profile. It describes the position of a follower axis relative to the position of a master axis.
A cam consists of one or more sections. These sections are called segments.
Segments use mathematical functions to describe the course within a segment.
A cam is defined via its "CamId" index, the axis and the technology group.

INFO

For example, cam 1 of axis 1 in the same technology group can have a different profile than cam 1 of axis 2 in the **same** technology group.

INFO

For example, cam 1 of axis 1 can have a different profile than cam 1 of axis 1 in **different** technology groups.

Depending on the controller type, the maximum number of cams and the maximum number of segments can vary. With JC365MC and JC440MC controllers, 30 cams with 100 segments each per cam follower axis are possible.
The cam index starts at 1 and must be less than the maximum number of cams.
The index of the cam disc does not necessarily have to start with "1", but can be freely selected within the above limits. The indexes do not have to be consecutive (e.g. 2, 3, 4, but can also be 2, 5, 6, 8 …).

3.1 Segments

Segments are identified via the "SegmentId" index.
These segments are clearly assigned to the respective cam via the technology group, the axis and the index of the "CamId" cam.
Each segment has a start position and is traversed from this position in positive master axis direction. The end of a segment is specified by the start position of the following segment.

4 Defining a cam

The following two steps are required to define a cam:

1. Creating the cam

<Techno>.Coupling.Cam.Create(Axis, CamId, SegmentCount)

This step is also the start of the cam definition and must be carried out before starting to define the individual segments.
The cam is uniquely assigned to the specified axis in this technology group.
To distinguish several cams, the "CamId" handle is used as an index in the cam memory.

INFO

If several axes have the same motion sequence, this must be defined separately for each axis using a separate cam.

INFO

If an axis in different technology groups has the same motion sequence, it must be defined as its own cam in the relevant technology group.

When creating the cam, the "SegmentCount", i.e. the number of segments, must be specified. It must therefore be known how many segments are required before defining the individual segments.

```
tecCam.Coupling.Cam.Create(xFollower, Cam, 2);
```

2. Describing the segments

After the cam has been created, the individual segments must be described.

```
tecCam.Coupling.Cam.DefineSegment(xFollower,1,1,MCTechnoCamSegmentTypes.AutoPoly1st, 0.0, MCTech-
noCamOptions.NoOption, 0.0, 0.0);
tecCam.Coupling.Cam.DefineSegment(xFollower,1,2,MCTechnoCamSegmentTypes.AutoPoly1st, 180.0, MCTech-
noCamOptions.NoOption, -180.0, 0.0);
```

Due to the unique assignment of the segments, it is not mandatory that the definition be in ascending order. Several cams can also be defined in a mixed way. The important thing is:

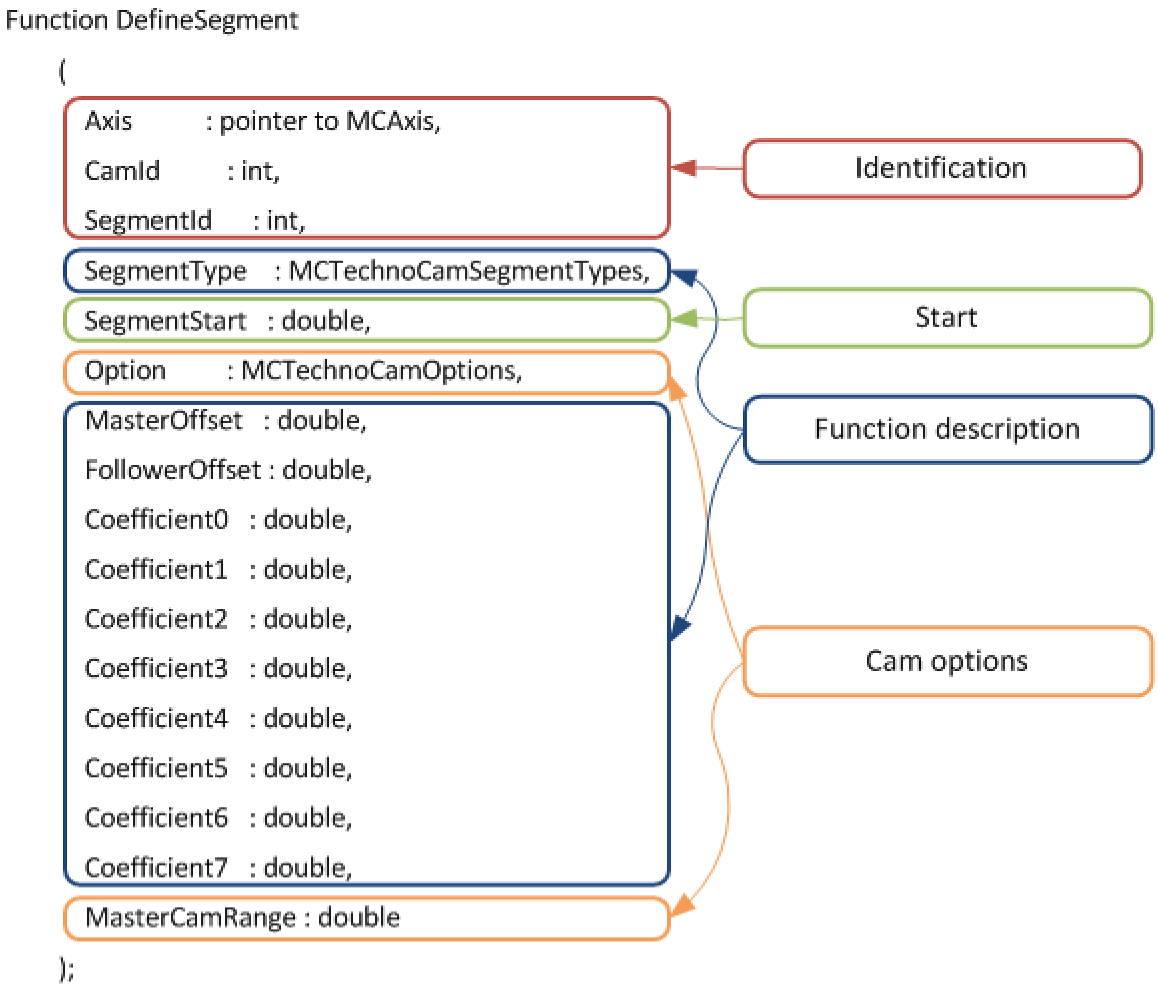
INFO

All segments must be defined before activating the cam.

5 Defining a segment using "DefineSegment()"

```
Function DefineSegment
    (
      Axis          : pointer to MCAxis,
      CamId         : int,
      SegmentId     : int,
      SegmentType   : MCTechnoCamSegmentTypes,
      SegmentStart  : double,
      Option        : MCTechnoCamOptions,
      MasterOffset  : double,
      FollowerOffset : double,
      Coefficient0  : double,
      Coefficient1  : double,
      Coefficient2  : double,
      Coefficient3  : double,
      Coefficient4  : double,
      Coefficient5  : double,
      Coefficient6  : double,
      Coefficient7  : double,
      MasterCamRange : double
    );
```

This powerful function can be divided into several sections, making it easier to understand.



Identification of the segment:

Techno:	Specifies which technology group this segment is to be assigned to.
Axis:	Specifies which axis this segment is to be assigned to.
CamId:	Index of the cam to which this segment is assigned.
SegmentId:	Index of the segment within the cam with respect to CamId.

Start:

Specifies from which master axis position this segment is valid.

Function description:

SegmentType:	Type of function of the segment
MasterOffset:	Offset of the master axis value that is added to the current master axis position
FollowerOffset:	Offset of the follower axis value which is added to the result of the function.
Coefficient0 ... 7:	Coefficients of the selected function according to the segment type

5.1 Supported function types

The following rule applies:
X_{Offset} : Masteroffset
Y_{Offset} : Followeroffset
c_0: Coefficient0
…
c_7: Coefficient7
$Y = y(t) – Y_{Offset}$
$X = x(t) + X_{Offset}$

5.1.1 1st order polynomial, straight line

$Y = c_1*X + c_0$

5.1.2 3rd order polynomial

$Y = c_3*X^3 + c_2* X^2 + c_1*X + c_0$

5.1.3 5th order polynomial

$Y = c_5 * X^5 + c_4 * X^4 + c_3 * X^3 + c_2 * X^2 + c_1 * X + c_0$

5.1.4 7th order polynomial

$Y = c_7 * X^7 + c_6 * X^6 + c_5 * X^5 + c_4 * X^4 + c_3* X^3 + c_2 * X^2 + c_1 * X + c_0$

5.1.5 Sine

$Y = c_0 * X + c_1 /\pi* \sin(c_2*X*\pi)$

5.1.6 AutoPoly

As the polynomial degree increases, the number of coefficients increases.
In order to facilitate the entry of polynomial functions, AutoPoly segment types can be used. In doing so, the polynomial is determined by boundary parameters.
Here, for example, the travel vector (dx, dy) is passed instead of the gradient for the 1st order polynomial.

Type	Master axis travel	Follower axis trav-el	dy/dx start	dy/dx end	d^2y/dx^2 start	d^2y/dx^2 end	d^3y/dx^3 start	d^3y/dx^3 end
Autopoly1st	X	X	-	-	-	-	-	
Autopoly3rd	X	X	X	X	-	-	-	-
Autopoly5th	X	X	X	X	X	X	-	-
Autopoly7th	X	X	X	X	X	X	X	X

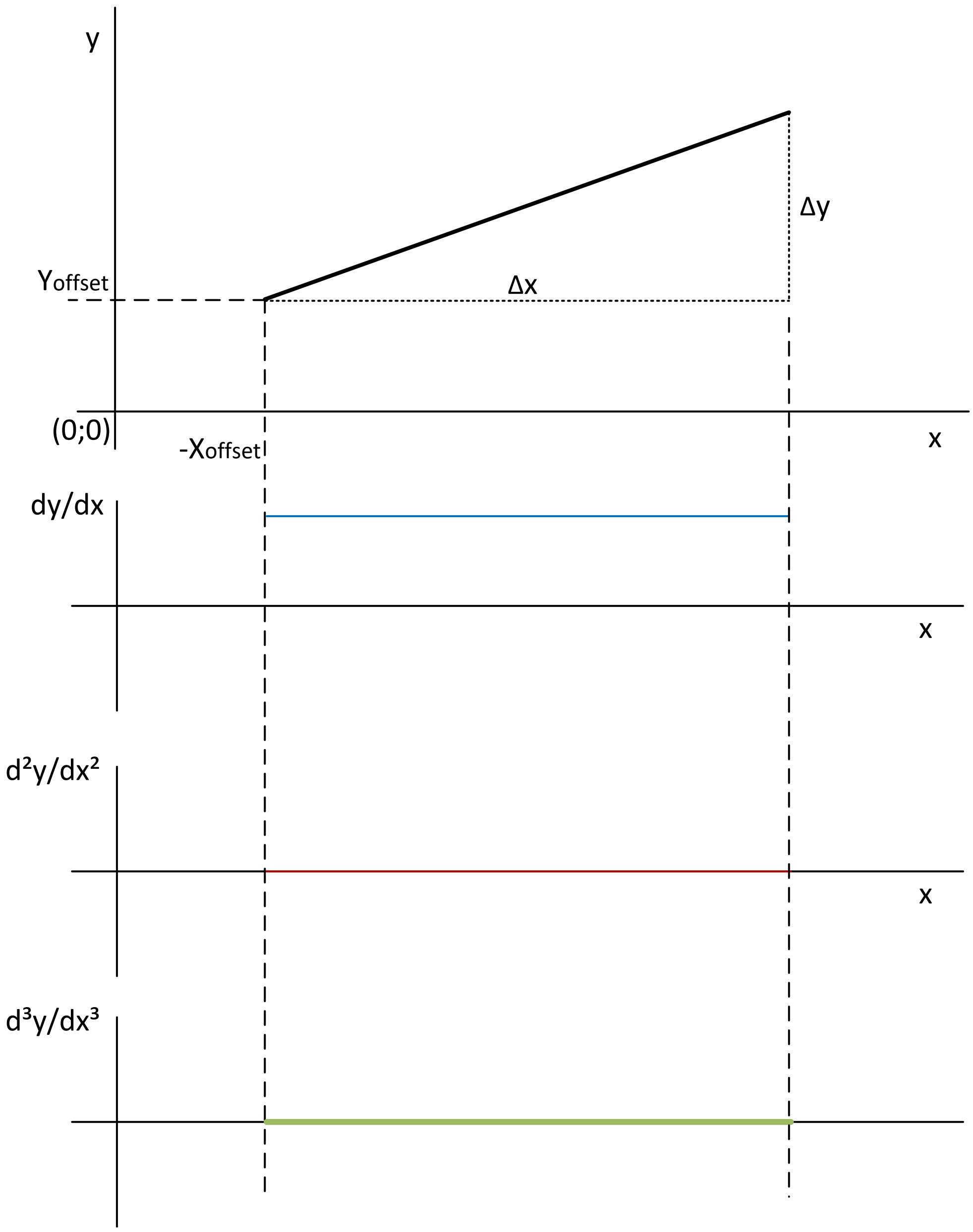
As the polynomial degree increases, the number of boundary parameters increases.

Meaning of the boundary parameters:

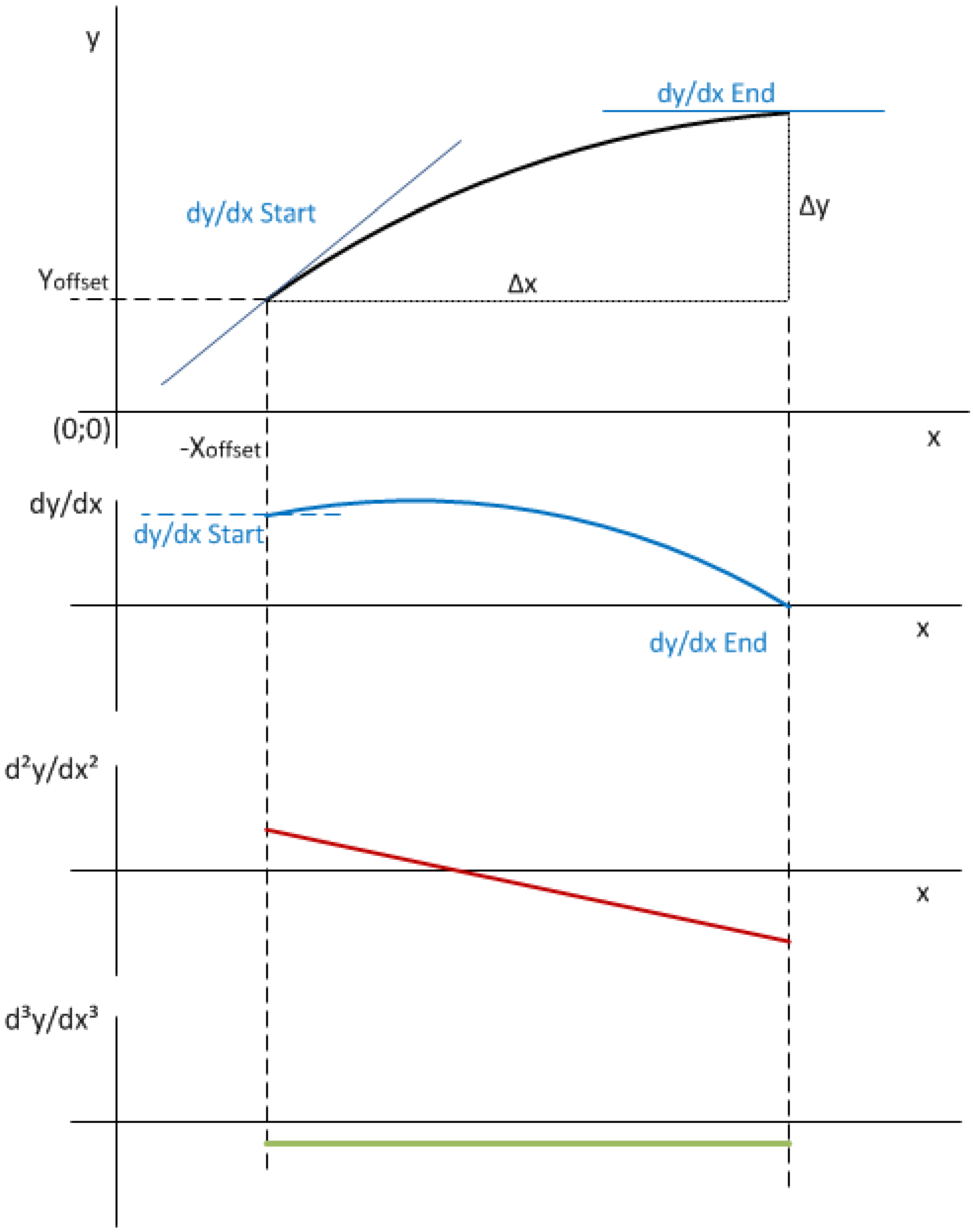
dy/dx: 1st derivation or velocity with respect to the master axis: If the value is 1, the follower axis moves at the same speed as the master axis at this point.
d^2y/dx^2: 2nd derivation or acceleration with respect to the master axis.
d^3y/dx^3: 3rd derivation or jerk with respect to the master axis.

The terms velocity, acceleration and jerk always refer to the master axis motion and are not the velocity, acceleration or jerk of the follower axis. For this purpose, the movement of the master axis must be taken into account.
Example: dy/dx = 1 - velocity with respect to the master axis is 1. If the master axis travels at a velocity of 100°/s, the follower axis velocity is also 100°/s (if the scaling factors are 1 when activating).

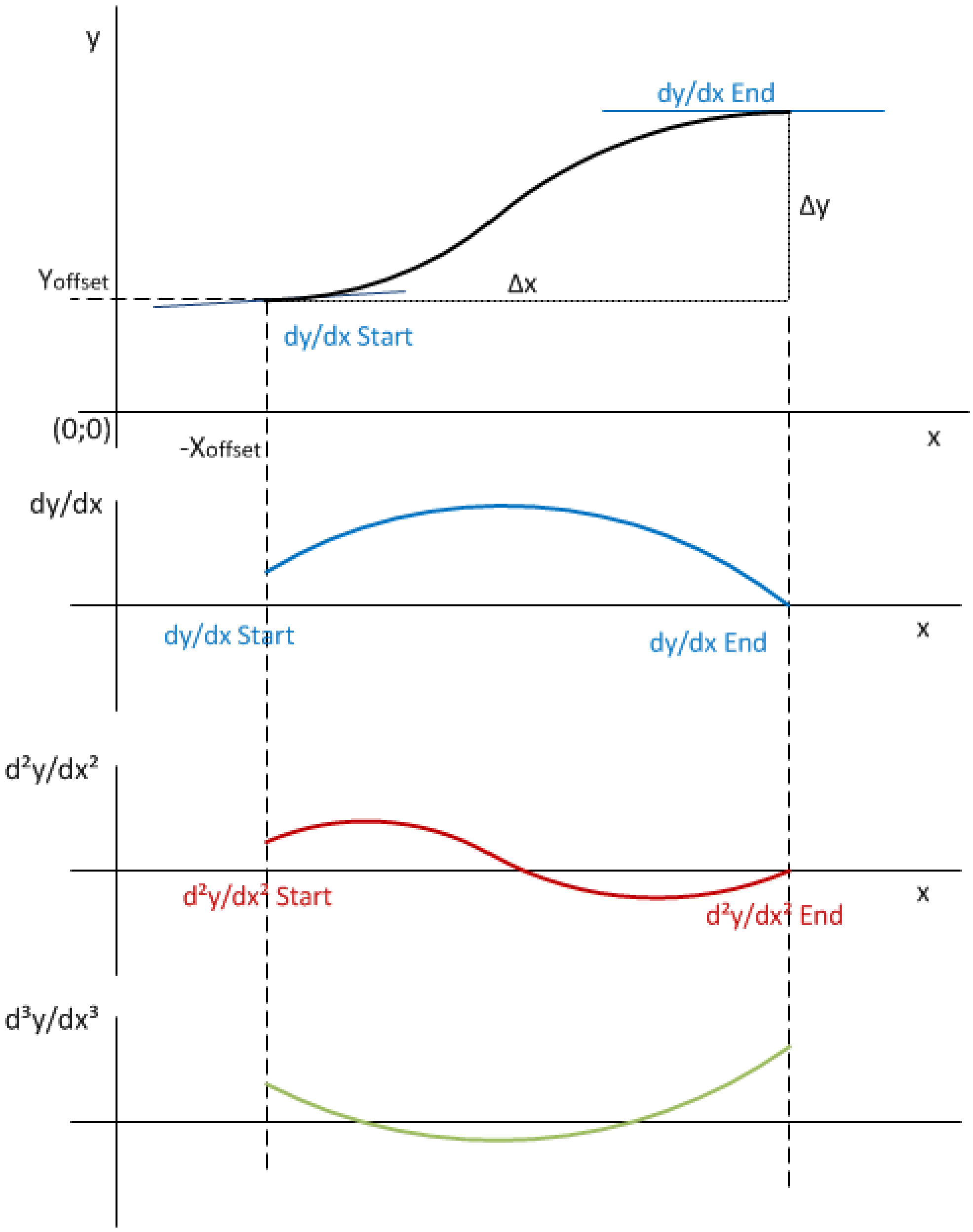
5.1.6.1 Autopoly1st:



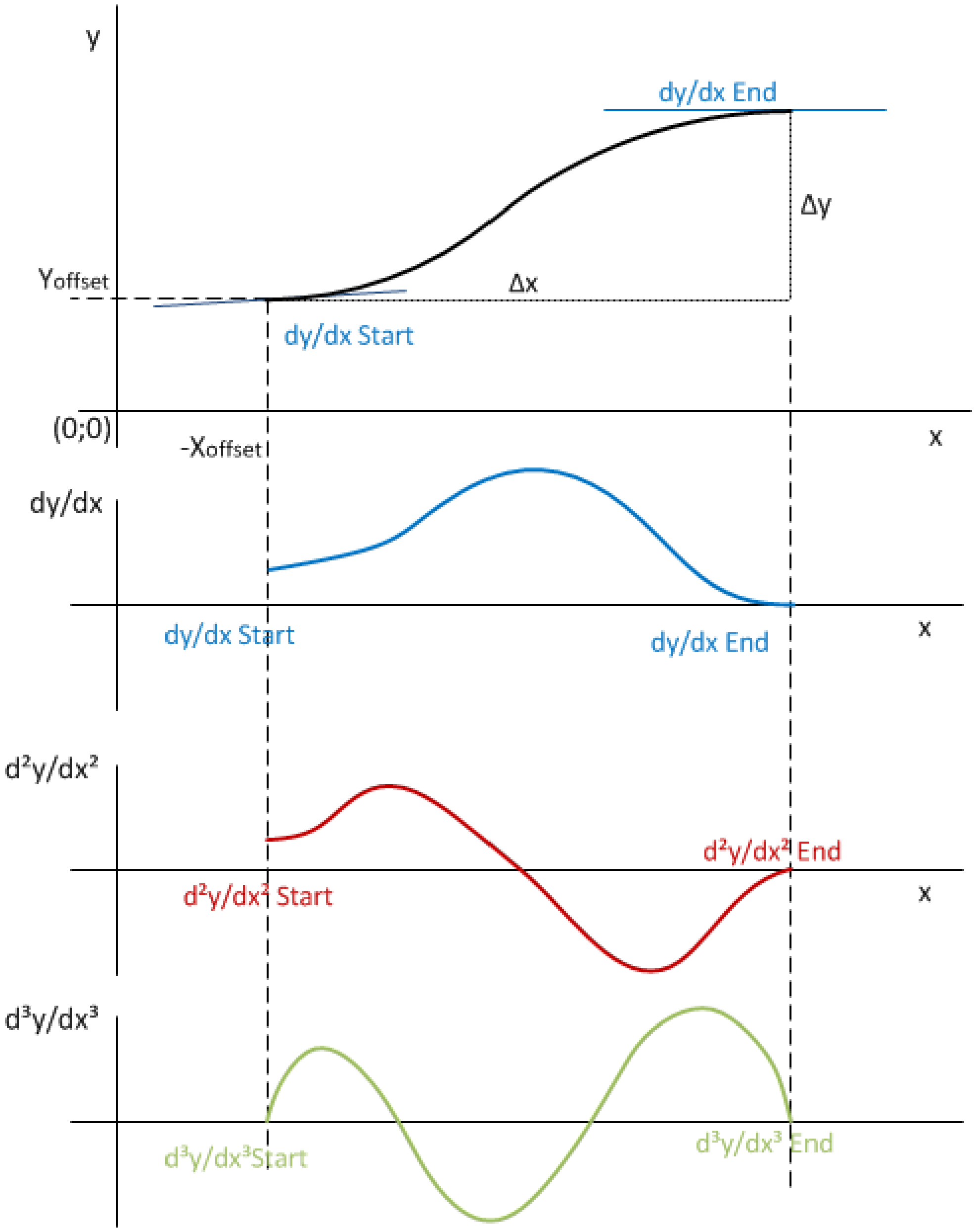
5.1.6.2 Autopoly3rd:



5.1.6.3 Autopoly5th:



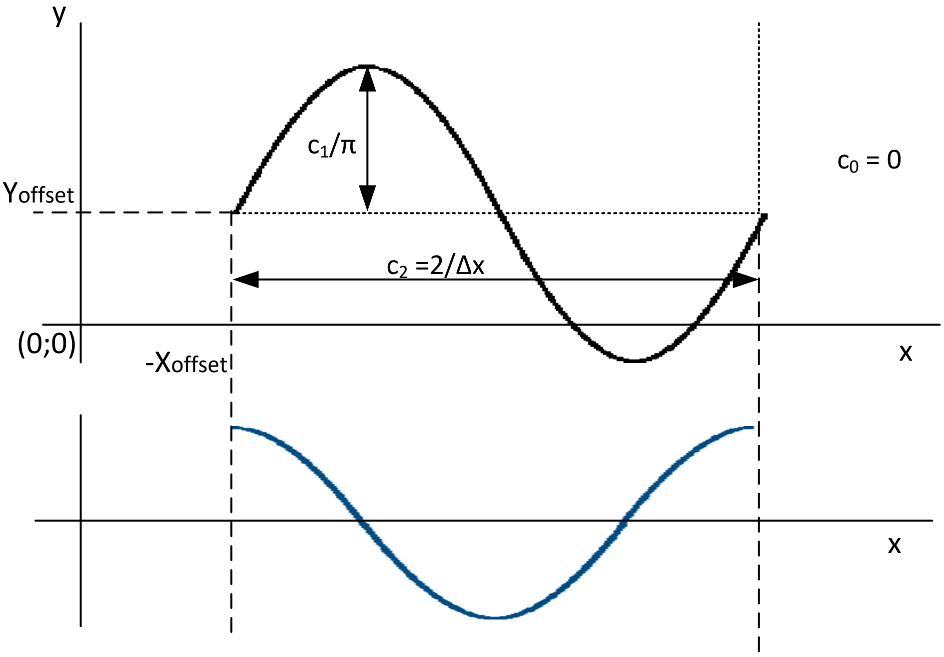
5.1.6.4 Autopoly7th:



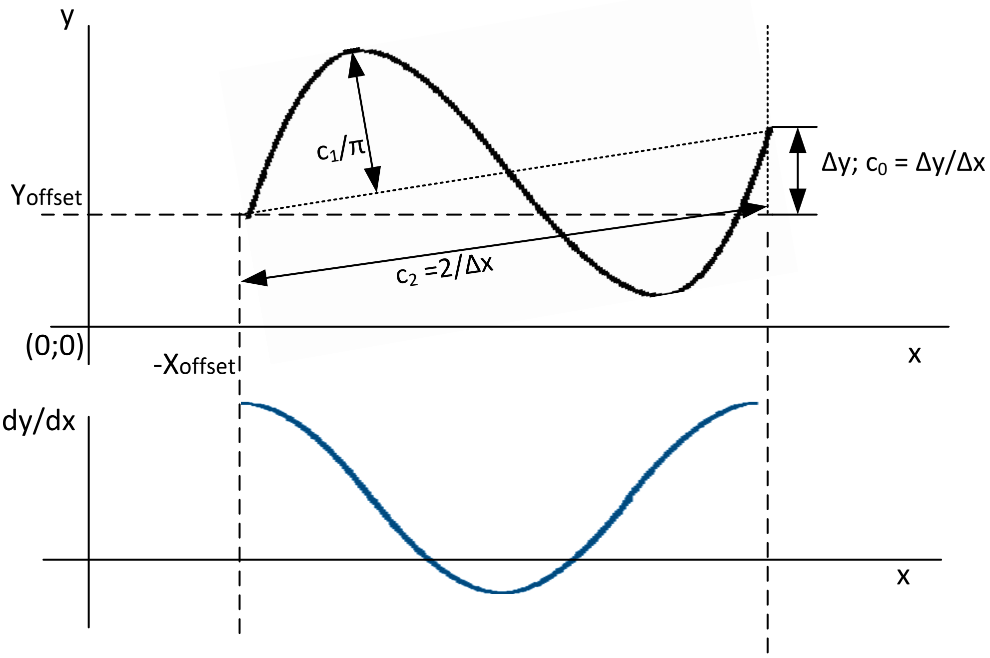
5.1.6.5 Sine²

$Y = c_0 * X + c_1 / \pi * \text{sine}(c_2 * X * \pi)$

Basic curve shape as pure sine without linear component



Basic curve shape as sine with linear component



Basic curve shape as sine² without linear component

The sine² function can also be represented as follows:
Sine² = 0.5 - 0.5*cos(2X) = 0.5 + 0.5*sine(2X+PI/2)

2*0.5(x+X0) = 2*0.5x+Pi/2
x+X0 = x+PI/2
X0 = PI/2

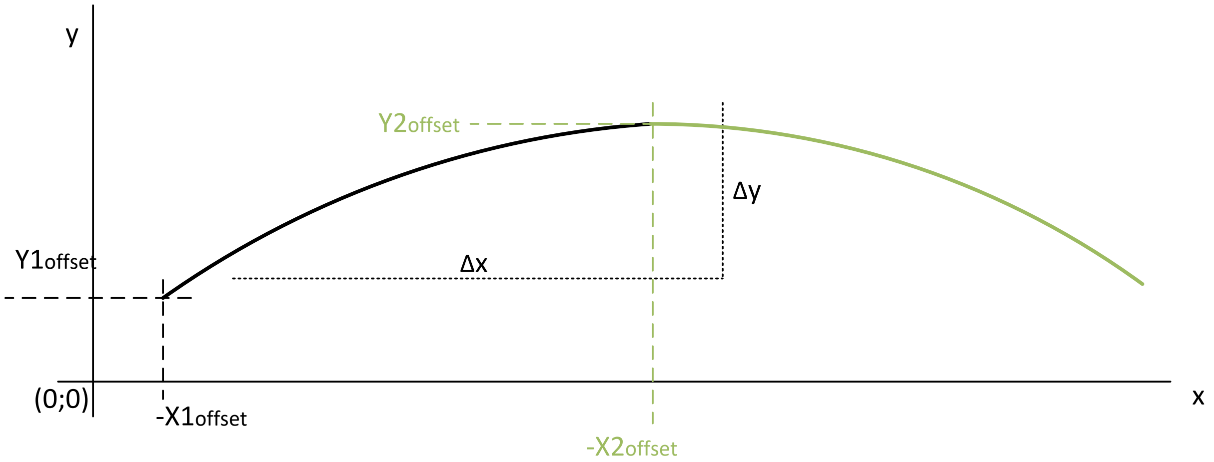
5.2 Segment options

The options of the segment definition are summarized in the enumeration "MCTechnoCam-Options":

0: NoOption — The segment definition is taken as specified.

1: AutoAppend — This option is only available from the 2nd segment. It is intended to simplify the definition so that the follower axis offset of the segments does not have to be calculated.

Example:




```
// Without AutoAppend Option
tecCam.Coupling.Cam.DefineSegment(xFollower,1,1,MCTechnoCamSegmentTypes.AutoPoly1st, X1Offset,
MCTechnoCamOptions.NoOption, -1.0 * X1Offset, Y1Offset,...)
tecCam.Coupling.Cam.DefineSegment(xFollower,1,2,MCTechnoCamSegmentTypes.AutoPoly1st, X2Offset,
MCTechnoCamOptions.NoOption, -1.0 * X2Offset, Y2Offset,...)
```

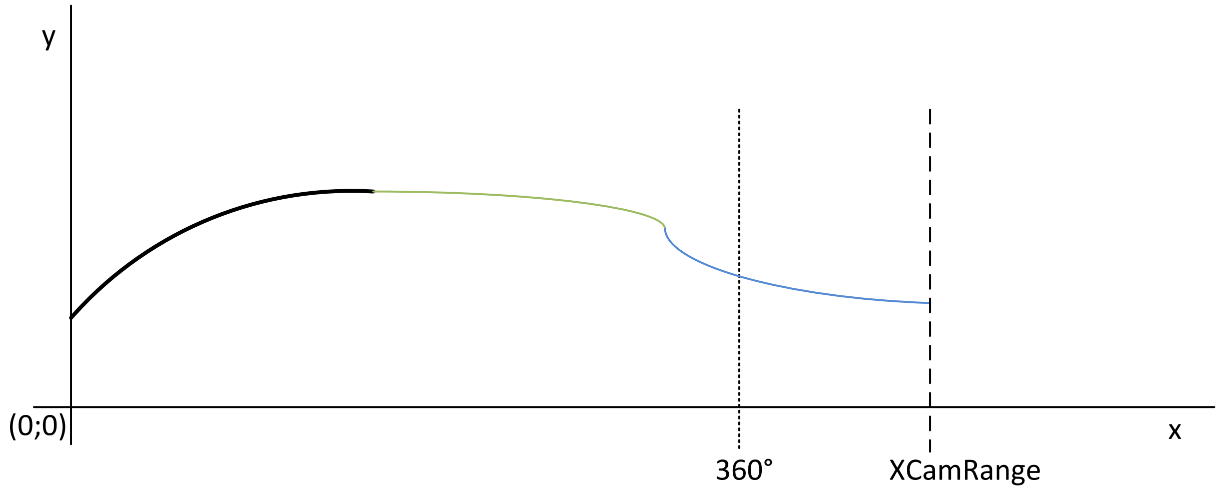
```
// With AutoAppend Option
tecCam.Coupling.Cam.DefineSegment(xFollower,1,1,MCTechnoCamSegmentTypes.AutoPoly1st, X1Offset,
MCTechnoCamOptions.NoOption, -1.0 * X1Offset, Y1Offset,...)
tecCam.Coupling.Cam.DefineSegment(xFollower,1,2,MCTechnoCamSegmentTypes.AutoPoly1st, X2Offset,
MCTechnoCamOptions.AutoAppend, -1.0 * X2Offset, 0.0,...)
```

2: UserDefinedMasterRange — This option is only available in the 1st segment. Here you define for the complete cam that the master axis range of the cam is not equal to the modulo range of the master axis.

Example:
The modulo range of the master axis is 360°, but the cam disc of the follower axis is to be defined in a master axis range of 540°. In this case, the "UserDefinedMasterRange" option must be selected.

INFO

If the cam is to be stretched/compressed in the master axis direction during use, the "UserDefinedMasterRange" option must also be selected here, even if the specified MasterCamRange corresponds to the modulo traversing range of the master axis.




```
tecCam.Coupling.Cam.DefineSegment(xFollower,1,1,MCTechnoCamSegmentTypes.AutoPoly1st, X1Offset,
MCTechnoCamOptions.UserDefinedMasterCamRange, -1.0 * X1Offset, Y1Offset,X1Range,
Y1Range,,,,,,,XCamRange);
tecCam.Coupling.Cam.DefineSegment(xFollower,1,2,MCTechnoCamSegmentTypes.AutoPoly1st, X2Offset,
MCTechnoCamOptions.AutoAppend, -1.0 * X2Offset, 0.0,...)
tecCam.Coupling.Cam.DefineSegment(xFollower,1,3,MCTechnoCamSegmentTypes.AutoPoly1st, X3Offset,
MCTechnoCamOptions.AutoAppend, -1.0 * X3Offset, 0.0,...)
```

6 Examples

In the following example, 3 cams are defined. The timing diagram of the curve is the same. Only the definitions are different:

CAM Sample 1: Definition of the cam with auto polynomials without further options
CAM Sample 2: Definition of the cam with auto polynomials with the "AutoAppend" option
CAM Sample 3: Definition of cams with polynomials with the specification of coefficients without further options

```
// CAM Sample 1
tecCam.Coupling.Cam.Create(xFollower, 1, 4);

tecCam.Coupling.Cam.DefineSegment(xFollower, 1, 1, MCTechnoCamSegmentTypes.AutoPoly3rd, 0.0,
MCTechnoCamOptions.NoOption,    0.0,   0.0, 90.0,   60.0,  0.0,  1.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 1, 2, MCTechnoCamSegmentTypes.AutoPoly5th, 90.0,
MCTechnoCamOptions.NoOption,  -90.0,  60.0, 90.0,   60.0,  1.0,  0.5, 0.0, 0.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 1, 3, MCTechnoCamSegmentTypes.AutoPoly5th, 180.0,
MCTechnoCamOptions.NoOption, -180.0, 120.0, 90.0,   45.0,  0.5, -0.5, 0.0, 0.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 1, 4, MCTechnoCamSegmentTypes.AutoPoly3rd, 270.0,
MCTechnoCamOptions.NoOption, -270.0, 165.0, 90.0, -165.0, -0.5,  0.0);


// CAM Sample 2
tecCam.Coupling.Cam.Create(xFollower, 2, 4);

tecCam.Coupling.Cam.DefineSegment(xFollower, 2, 1, MCTechnoCamSegmentTypes.AutoPoly3rd, 0.0,
MCTechnoCamOptions.NoOption,      0.0, 0.0, 90.0,   60.0,  0.0,  1.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 2, 2, MCTechnoCamSegmentTypes.AutoPoly5th, 90.0,
MCTechnoCamOptions.AutoAppend,  -90.0, 0.0, 90.0,   60.0,  1.0,  0.5, 0.0, 0.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 2, 3, MCTechnoCamSegmentTypes.AutoPoly5th, 180.0,
MCTechnoCamOptions.AutoAppend, -180.0, 0.0, 90.0,   45.0,  0.5, -0.5, 0.0, 0.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 2, 4, MCTechnoCamSegmentTypes.AutoPoly3rd, 270.0,
MCTechnoCamOptions.AutoAppend, -270.0, 0.0, 90.0, -165.0, -0.5,  0.0);

// CAM Sample 3
tecCam.Coupling.Cam.Create(xFollower, 3, 4);

tecCam.Coupling.Cam.DefineSegment(xFollower, 3, 1, MCTechnoCamSegmentTypes.Poly3rd, 0.0,   MCTech-
noCamOptions.NoOption,    0.0,   0.0, 0.0,  0.0,  0.01111111, -4.115226e-005, 0.0, 0.0);
tecCam.Coupling.Cam.DefineSegment(xFollower, 3, 2, MCTechnoCamSegmentTypes.Poly5th, 90.0,  MCTech-
noCamOptions.NoOption,  -90.0,  60.0, 0.0,  1.0,  0.0,        -0.0001646091,  2.057613e-006, -
7.620789e-009);
tecCam.Coupling.Cam.DefineSegment(xFollower, 3, 3, MCTechnoCamSegmentTypes.Poly5th, 180.0, MCTech-
noCamOptions.NoOption, -180.0, 120.0, 0.0,  0.5,  0.0,         0.0004938272, -9.602195e-006,
4.572474e-008);
tecCam.Coupling.Cam.DefineSegment(xFollower, 3, 4, MCTechnoCamSegmentTypes.Poly3rd, 270.0, MCTech-
noCamOptions.NoOption, -270.0, 165.0, 0.0, -0.5, -0.050,       0.0003909465,  0.0, 0.0);
```

Timing diagram:

Blue: Set position of the master axis
Red: Setpoint position of the follower axis
Green: Set velocity of the follower axis
Orange: Set acceleration of the follower axis
Pink: Number of the active cam profile
Light blue: Number of the active cam segment

The oscilloscope clearly shows that despite different definitions, the curves are identical.
The change of cams can be recognized by the number of the active cam profile (pink).
The individual sections within the cam can be recognized not only by the number of the active cam segment (light blue) but also, for example, by the jumps in acceleration (orange).
Jumps in acceleration often lead to an unsteady behavior which can be avoided by explicitly specifying the acceleration at the transition points when using the 5th order polynomial.
Peaks in the acceleration curve at the start and end of the motion reflect the acceleration and deceleration of the master axis.

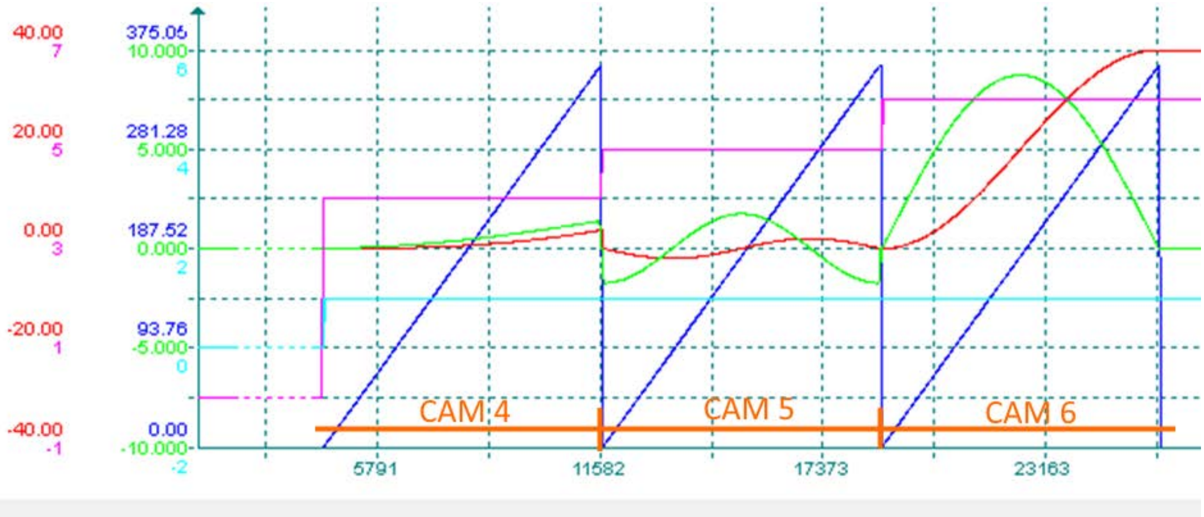
CAM Sample 4 ... 6: Various definitions of a "sine" type cam.

```
// CAM Sample 4: Part of Sine
tecCam.Coupling.Cam.Create(xFollower, 4, 1);
tecCam.Coupling.Cam.DefineSegment(xFollower, 4, 1, MCTechnoCamSegmentTypes.Sine, 0.0, MCTechnoCam-
Options.NoOption,
0.0,
0.0,
10.0/360.0,
-2.0*cPI*cPI, // Amplitude
-0.5 / 360.0,
);

// CAM Sample 5: Pure Sine
tecCam.Coupling.Cam.Create(xFollower, 5, 1);
tecCam.Coupling.Cam.DefineSegment(xFollower, 5, 1, MCTechnoCamSegmentTypes.Sine, 0.0, MCTechnoCam-
Options.NoOption,
0.0,
0.0,
0.0,
2.0*cPI, // Amplitude
2.0 / 360.0,
);

// CAM Sample 6: Part of Sine
```

```
tecCam.Coupling.Cam.Create(xFollower, 6, 1);
tecCam.Coupling.Cam.DefineSegment(xFollower, 6, 1, MCTechnoCamSegmentTypes.Sine, 0.0, MCTechnoCam-
Options.NoOption,
180.0, // Offset-Shift to get specific part of sine
20.0,  // Follower-Offset-Shift to start at 0, since sine begins here at -20
0.0,
20.0*cPI, // Amplitude
1.0 / 360.0,
);
```



By specifying different parameters when defining a "sine" type cam segment, different sections of a sine can be configured and described.
As with CAM 4, this can be used, for example, as an acceleration segment (for CAM 5 as a sinusoidal oscillating movement and for CAM 6 as positioning).

Jetter automation

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg
www.jetter.de

E-mail info@jetter.de
Phone: +49 7141 2550-0

We automate your success.