

## Motion Control - Kurvenscheibendefinition

Application Note 052

608 847 07\_00

Dieses Dokument wurde von der Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg  
Deutschland

[www.jetter.de](http://www.jetter.de)

**Telefon:**

Zentrale	+49 7141 2550-0
Vertrieb	+49 7141 2550-531
Technische Hotline	+49 7141 2550-444

**E-Mail:**

Technische Hotline	info@jetter.de
Vertrieb	hotline@jetter.de
	vertrieb@jetter.de

Produktname	<b>Motion Control - Kurvenscheibendefinition</b>
Dokumentart	<b>Application Note 052</b>
Originaldokument	
Dokumentversion	1.00
Artikelnummer	608 847 07_00
Ausgabedatum	21.05.2021

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
1.1	Voraussetzungen .....	1
<b>2</b>	<b>Kurvenscheibendefinition - Einführung</b> .....	<b>2</b>
<b>3</b>	<b>Kurvenscheibe</b> .....	<b>3</b>
3.1	Segmente .....	3
<b>4</b>	<b>Definieren einer Kurvenscheibe</b> .....	<b>4</b>
<b>5</b>	<b>Definieren eines Segments mit „DefineSegment()“</b> .....	<b>5</b>
5.1	Unterstützte Funktionsarten .....	7
5.1.1	Polynom 1. Grades, gerade Linie .....	7
5.1.2	Polynom 3. Grades .....	7
5.1.3	Polynom 5. Grades .....	7
5.1.4	Polynom 7. Grades .....	7
5.1.5	Sinus 7 .....	7
5.1.6	AutoPoly .....	7
5.1.6.1	Autopoly1st: .....	9
5.1.6.2	Autopoly3rd: .....	10
5.1.6.3	Autopoly5th: .....	11
5.1.6.4	Autopoly7th: .....	12
5.1.6.5	Sin <sup>2</sup> .....	13
5.2	Segmentoptionen .....	14
<b>6</b>	<b>Beispiele</b> .....	<b>16</b>

# 1 Einleitung

## 1.1 Voraussetzungen

- Für die Code- und Projektbeispiele sowie Screenshots wurden folgende Versionen verwendet:
  - JetSym 5.601
  - Motion API 1.0.0.16
  - JC365MC-OS 1.30.0.00
- Application Note "*Technologieverbund*"

## 2 Kurvenscheibendefinition - Einführung

In dieser Anwendungsbeschreibung wird aufgezeigt, wie

- eine Kurvenscheibe grundsätzlich definiert wird;
- aus welchen Elementen eine Kurvenscheibe besteht;
- welche Typen von Segmenten zur Verfügung stehen;
- welche Optionen vorhanden sind.

## 3 Kurvenscheibe

Eine Kurvenscheibe ist ein Bewegungsprofil. Es beschreibt die Position einer Folgeachse bezogen auf die Position einer Leitachse.

Die Kurvenscheibe besteht hierbei aus einem oder mehreren Abschnitte. Diese Abschnitte werden als Segmente bezeichnet.

Diese Segmente beschreiben über mathematische Funktionen den Verlauf innerhalb eines Segments.

Eine Kurvenscheibe wird über ihren Index „CamId“, die Achse und den Technologieverbund definiert.

- Z.B. kann die Kurvenscheibe 1 der Achse 1 im gleichen Technologieverbund einen anderen Verlauf haben als die Kurvenscheibe 1 der Achse **2** im **gleichen** Technologieverbund.
- Z.B. kann die Kurvenscheibe 1 der Achse 1 einen anderen Verlauf haben als die Kurvenscheibe 1 der Achse 1 in **unterschiedlichen** Technologieverbunden.

Je nach Steuerungstyp können die maximale Anzahl an Kurvenscheiben und die maximale Anzahl an Segmenten unterschiedlich sein. Bei einer JC365MC und JC440MC sind 30 Kurvenscheiben mit je 100 Segmenten pro Kurvenscheibenfolgeachse möglich.

Der Index der Kurvenscheibe beginnt bei 1 und muss kleiner als die maximale Anzahl an Kurvenscheiben sein.

Der Index der Kurvenscheibe muss nicht zwangsläufig mit „1“ beginnen, sondern kann innerhalb der vorstehenden Grenzen frei gewählt werden. Die Indexe müssen nicht aufeinanderfolgend sein (z.B. 2, 3, 4 sondern können auch 2, 5, 6, 8 ... sein).

### 3.1 Segmente

Segmente werden über den Index „SegmentId“ identifiziert.

Diese Segmente werden über den Technologieverbund, die Achse und den Index der Kurvenscheibe „CamId“ eindeutig der jeweiligen Kurvenscheibe zugeordnet.

Jedes Segment hat eine Startposition und wird ab dieser Position in positiver Leitachsrichtung ausgeführt. Das Ende eines Segments wird durch die Startposition des Folgesegments vorgegeben.

## 4 Definieren einer Kurvenscheibe

Um eine Kurvenscheibe zu definieren braucht es die folgenden beiden Schritte:

### 1. Erstellen der Kurvenscheibe

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

Dieser Schritt ist auch der Start der Kurvenscheibendefinition und muss ausgeführt werden, bevor mit der Definition der einzelnen Segmente begonnen wird.

Die Kurvenscheibe wird in diesem Technologieverbund der angegebenen Achse eindeutig zugeordnet.

Die Unterscheidung mehrerer Kurvenscheiben erfolgt über das Handle „CamId“ als Index im Kurvenscheibenspeicher.

- Haben mehrere Achsen den gleichen Bewegungsablauf, muss dieser für jede Achse über eine eigene Kurvenscheibe separat definiert werden.
- Hat eine Achse in unterschiedlichen Technologieverbunden den gleichen Bewegungsablauf, muss dieser in den betreffenden Technologieverbunden als eigene Kurvenscheibe definiert werden.

Beim Erstellen der Kurvenscheibe muss die Anzahl der Segmente „SegmentCount“ angegeben werden. Es muss damit vor der Definition der einzelnen Segmente bekannt sein, wie viele Segmente benötigt werden.

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

### 2. Beschreibung der Segmente

Nach dem Erzeugen der Kurvenscheibe erfolgt die Beschreibung der einzelnen Segmente.

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

Aufgrund der eindeutigen Zuordnung der Segmente ist es nicht zwingend erforderlich, dass die Definition in aufsteigender Reihenfolge erfolgt. Es können auch mehrere Kurvenscheibe durcheinander definiert werden. Wichtig ist nur:

- Vor dem Aktivieren der Kurvenscheibe müssen alle Segmente definiert sein

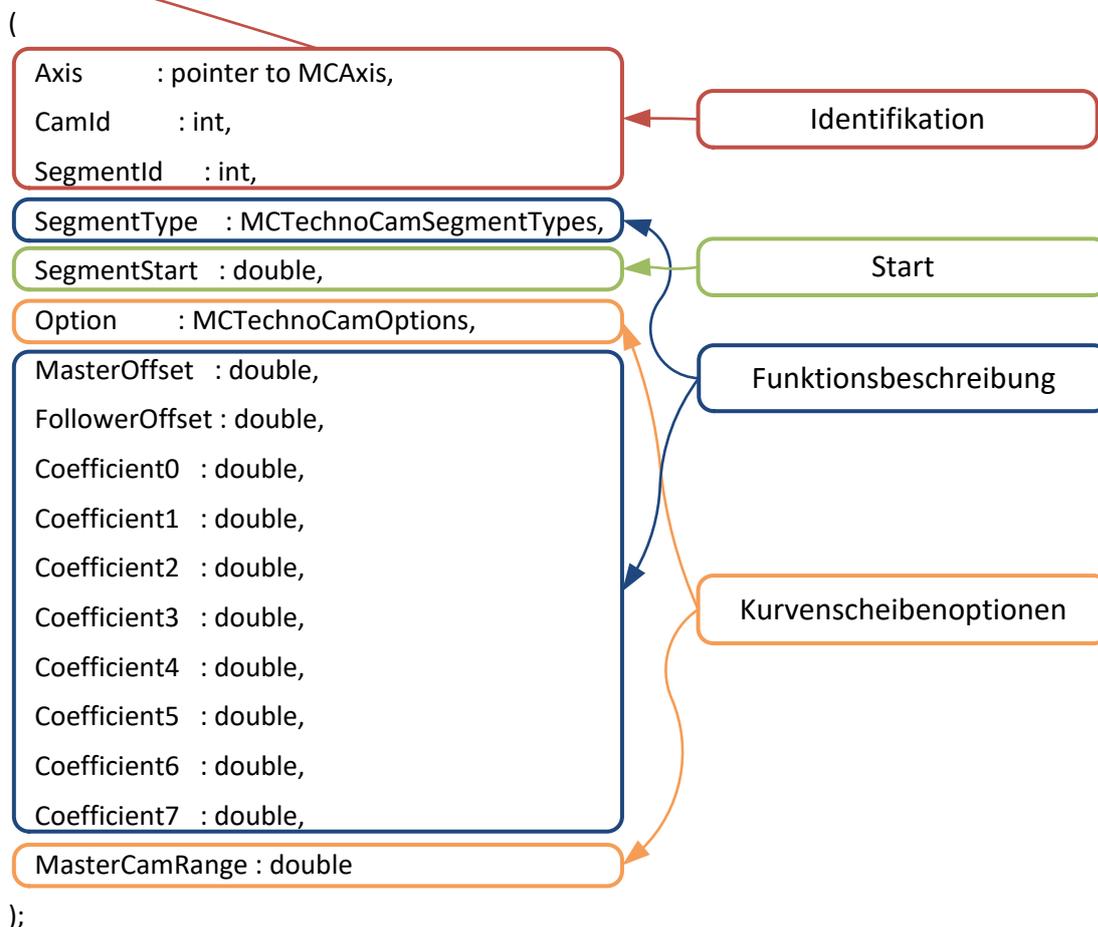
## 5 Definieren eines Segments mit „DefineSegment()“

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
);
```

Diese mächtige Funktion lässt sich in mehrere Abschnitte unterteilen, wodurch sie einfacher zu verstehen ist.

Function <Techno>.Coupling.Cam.DefineSegment



**Identifikation** des Segments:

**Techno:** Festlegung, welchem Technologieverbund dieses Segment zuzuordnen ist.  
**Axis:** Festlegung, welcher Achse dieses Segment zuzuordnen ist.  
**CamId:** Index der Kurvenscheibe, der dieses Segment zuzuordnen ist.  
**SegmentId:** Index des Segments innerhalb der Kurvenscheibe bzgl. CamId.

**Start:**

Festlegung, ab welcher Leitachsposition dieses Segment gültig ist.

**Funktionsbeschreibung:**

**SegmentType:** Typ der Funktion des Segments  
**MasterOffset:** Offsets des Leitachswertes, welcher zur aktuellen Leitachsposition hinzugezählt wird  
**FollowerOffset:** Offset des Folgeachswertes, welcher dem Funktionsergebnis hinzugezählt wird.  
**Coefficient0 ... 7:** Koeffizienten der ausgewählten Funktion entsprechend dem SegmentType

## 5.1 Unterstützte Funktionsarten

Es gilt:

$X_{\text{Offset}}$  : Masteroffset

$Y_{\text{Offset}}$  : Followeroffset

$c_0$ : Coefficient0

...

$c_7$ : Coefficient7

$Y = y(t) - Y_{\text{Offset}}$

$X = x(t) + X_{\text{Offset}}$

### 5.1.1 Polynom 1. Grades, gerade Linie

$$Y = c_1 * X + c_0$$

### 5.1.2 Polynom 3. Grades

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

### 5.1.3 Polynom 5. Grades

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

### 5.1.4 Polynom 7. Grades

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

### 5.1.5 Sinus

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

### 5.1.6 AutoPoly

Mit steigendem Polynomgrad steigt die Anzahl der Koeffizienten.

Um die Eingabe der polynomialen Funktionen zu erleichtern, können Autopoly-Segmenttypen verwendet werden.

Hierbei wird z.B. beim Polynom 1. Grades anstatt der Steigung der Fahrvektor (dx, dy) übergeben.

Typ	Leitachsweg	Folgeachsweg	dy/dx Anfang	dy/dx Ende	d <sup>2</sup> y/dx <sup>2</sup> Anfang	d <sup>2</sup> y/dx <sup>2</sup> Ende	d <sup>3</sup> y/dx <sup>3</sup> Anfang	d <sup>3</sup> y/dx <sup>3</sup> Ende
Autopoly1st	X	X	-	-	-	-	-	-
Autopoly3rd	X	X	X	X	-	-	-	-
Autopoly5th	X	X	X	X	X	X	-	-
Autopoly7th	X	X	X	X	X	X	X	X

Mit steigendem Polynomgrad steigt die Anzahl der Randparameter.

Bedeutung der Randparameter:

dy/dx: 1. Ableitung bzw. Geschwindigkeit bezüglich der Leitachse: Bei einem Wert 1 fährt die Folgeachse in diesem Punkt mit der gleichen Geschwindigkeit wie die Leitachse.

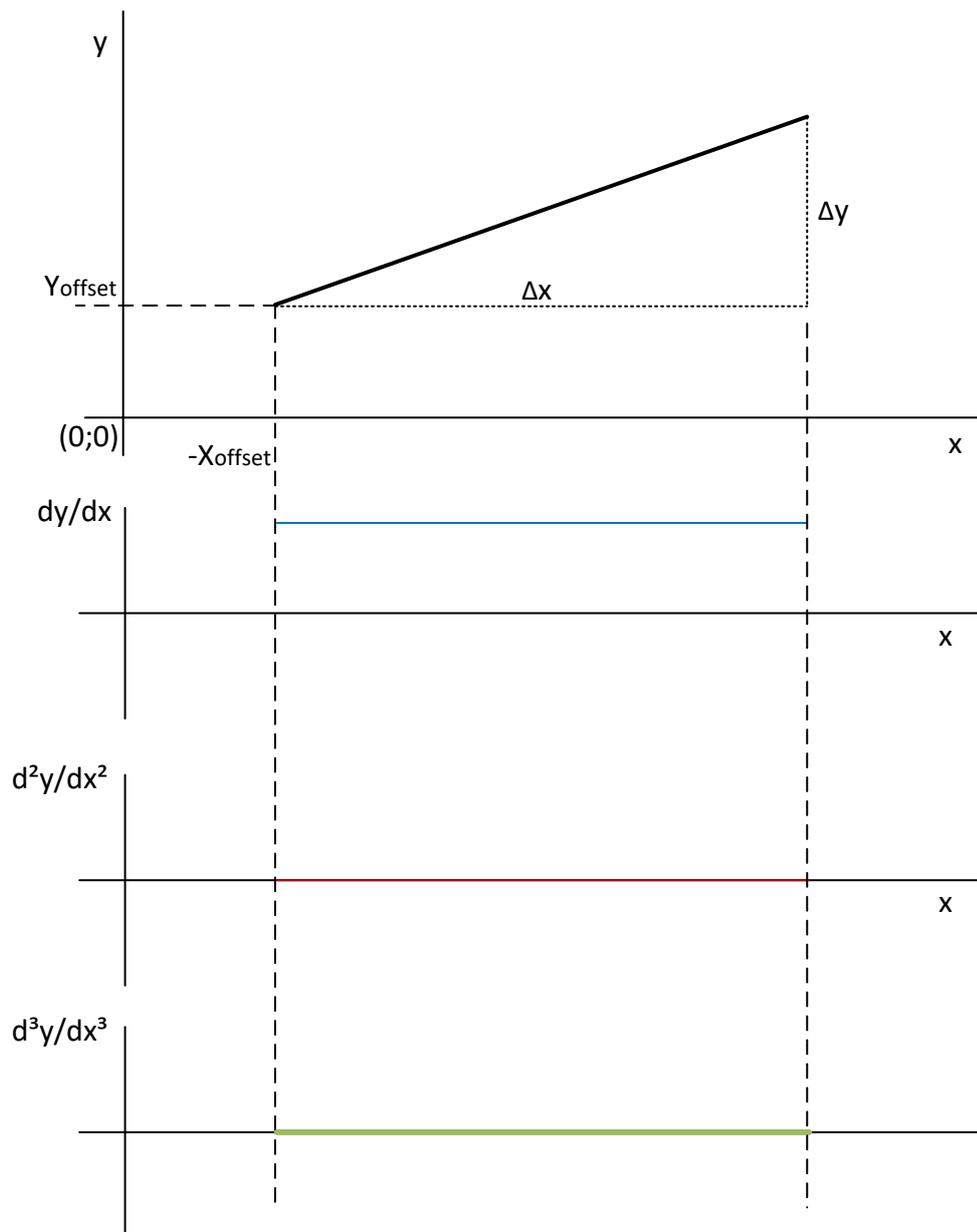
$d^2y/dx^2$ : 2. Ableitung bzw. Beschleunigung bezüglich der Leitachse.

$d^3y/dx^3$ : 3. Ableitung bzw. Ruck bezüglich der Leitachse.

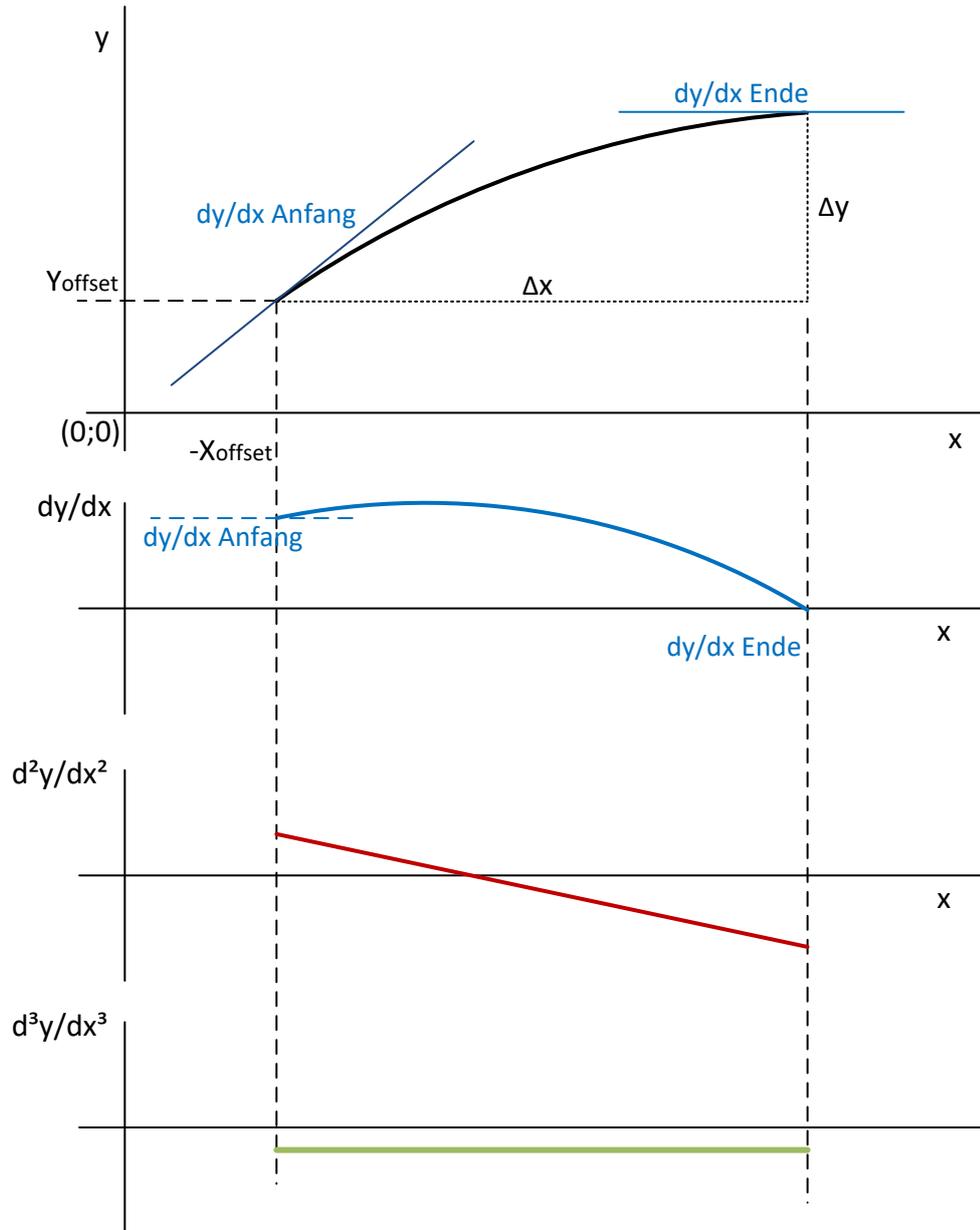
Die Bezeichnung Geschwindigkeit, Beschleunigung und Ruck beziehen sich immer auf die Leitachsenbewegung und sind nicht die von der Achse gefahrene Geschwindigkeit, Beschleunigung oder Ruck. Hierfür muss die Bewegung der Leitachse berücksichtigt werden.

Beispiel:  $dy/dx = 1$  - Geschwindigkeit bzgl. Leitachse ist 1. Fährt die Leitachse mit einer Geschwindigkeit von  $100^\circ/s$ , so ist die Folgeachsengeschwindigkeit auch  $100^\circ/s$  (gesetzt den Fall, die Skalierungsfaktoren beim Aktivieren sind 1).

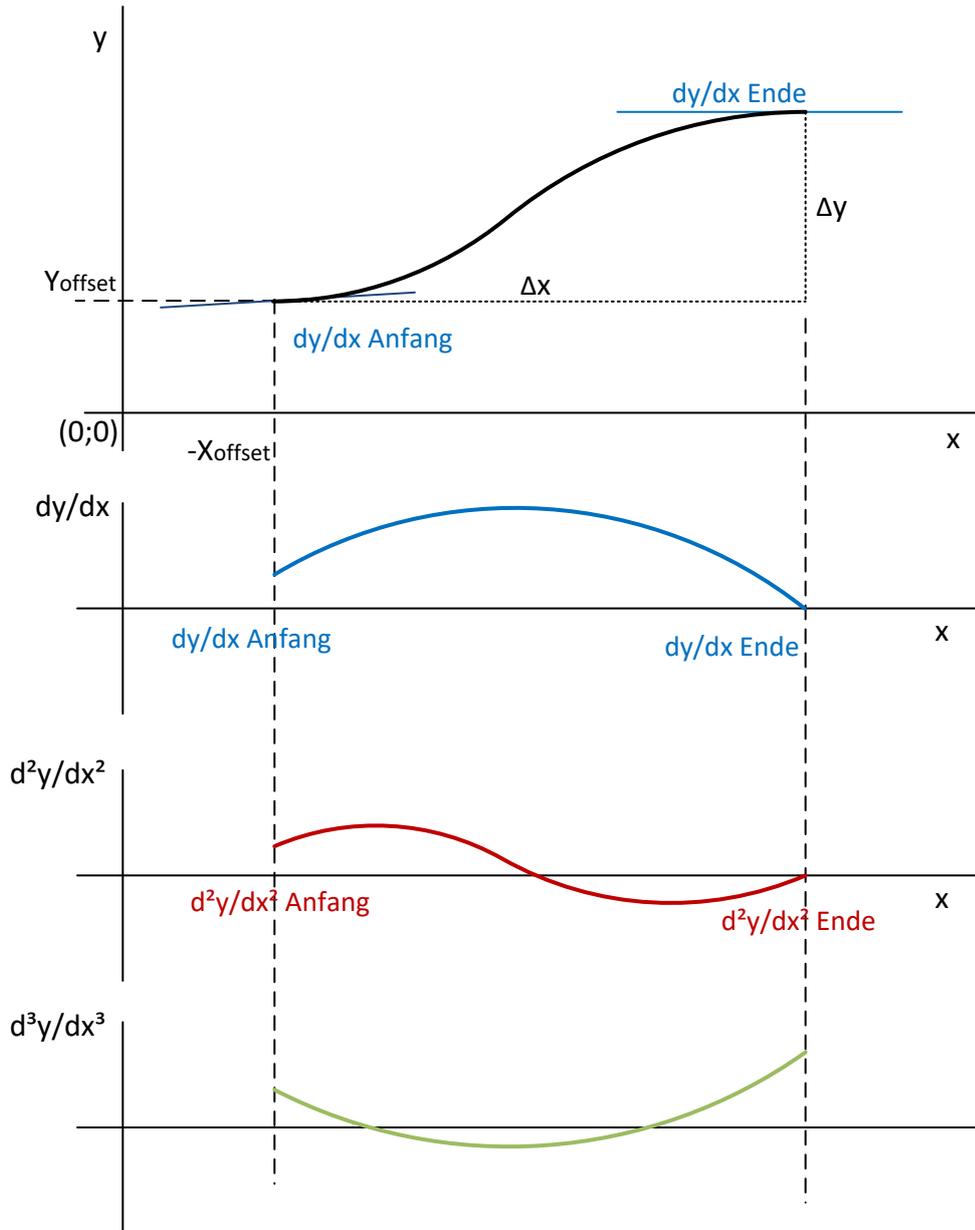
### 5.1.6.1 Autopoly1st:



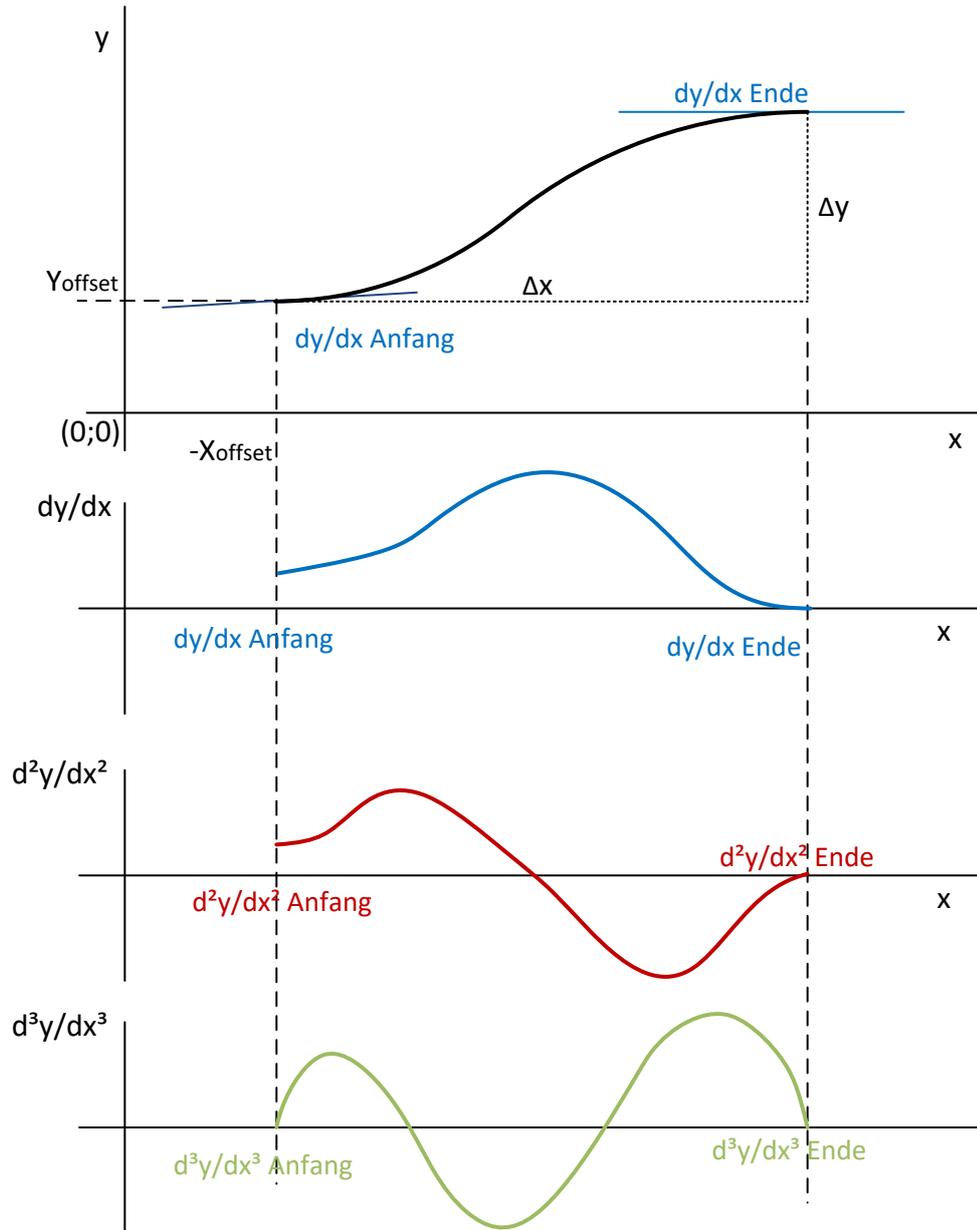
5.1.6.2 Autopoly3rd:



5.1.6.3 Autopoly5th:



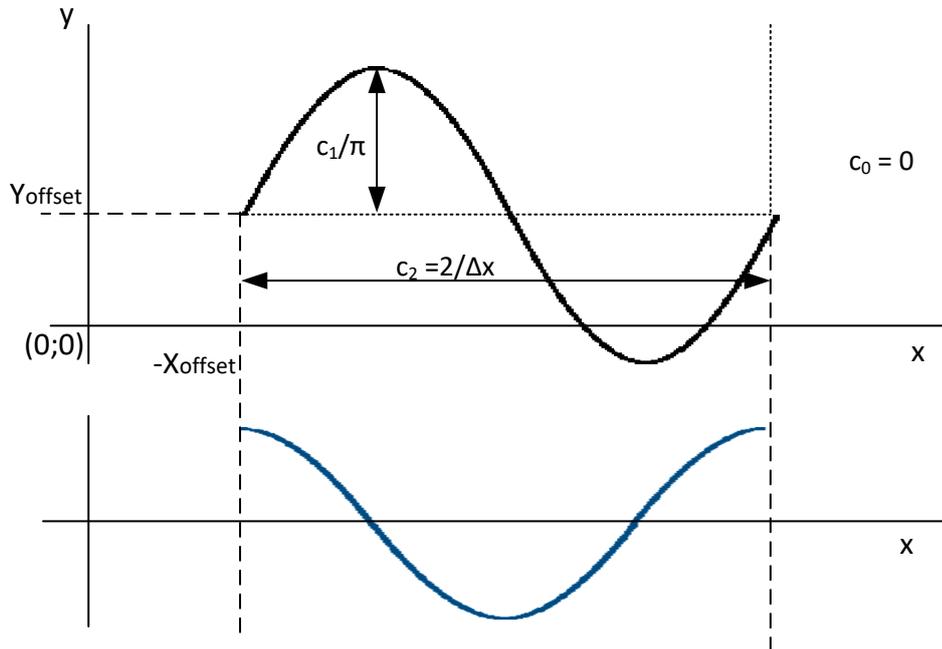
5.1.6.4 Autopoly7th:



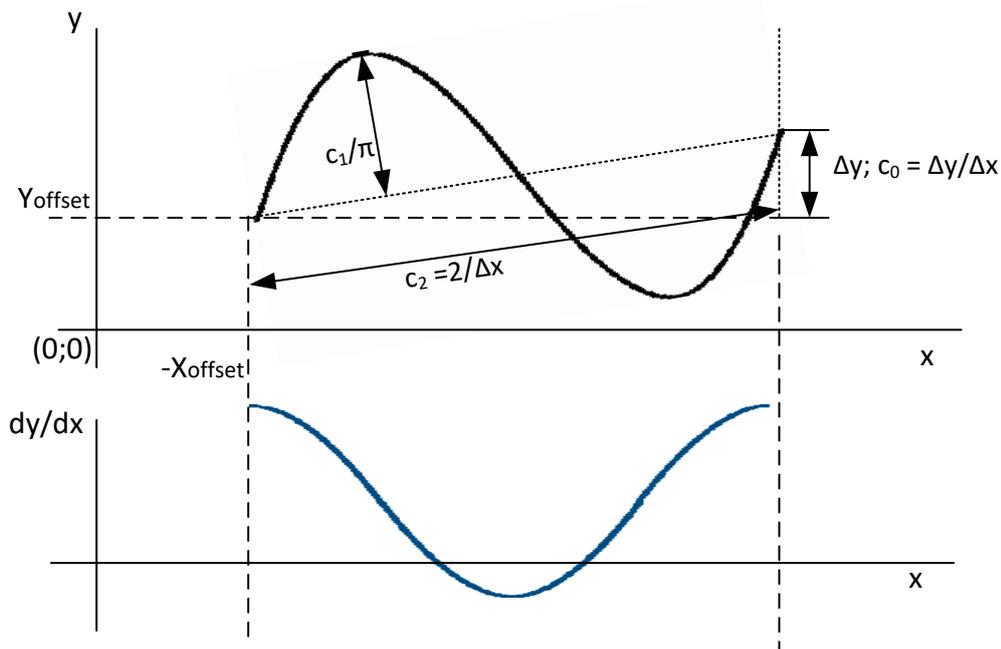
### 5.1.6.5 Sin<sup>2</sup>

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

Prinzipieller Kurvenverlauf als reiner Sinus ohne Linearanteil



Prinzipieller Kurvenverlauf als Sinus mit Linearanteil



Prinzipieller Kurvenverlauf als Sinus<sup>2</sup> ohne Linearanteil.

Die Funktion sin<sup>2</sup> lässt sich auch als folgendermaßen darstellen:

$$\sin^2 = 0,5 - 0,5 * \cos(2X) = 0,5 + 0,5 * \sin(2X + \pi/2)$$

$$2 * 0,5(x + X_0) = 2 * 0,5x + \pi/2$$

$$x + X_0 = x + \pi/2$$

$$X_0 = \pi/2$$

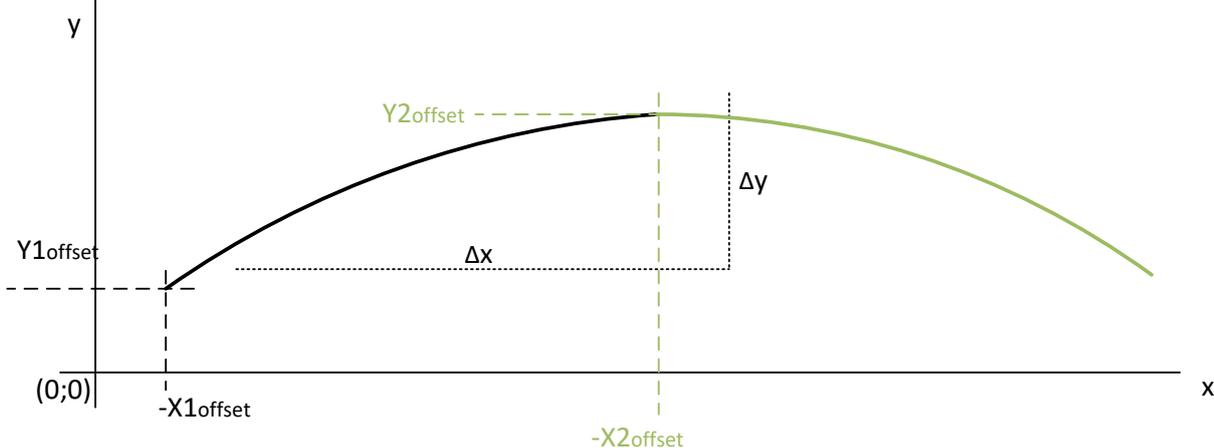
## 5.2 Segmentoptionen

Die Optionen der Segmentdefinition sind in der Enumeration „MCTechnoCamOptions“ zusammengefasst:

0: NoOption  
1: AutoAppend

Die Definition des Segments wird so genommen wie angegeben. Diese Option ist erst mit ab dem 2. Segment verfügbar. Sie soll die Definition dahingehend vereinfachen, dass der Folgeachsoffset der Segmente nicht berechnet werden muss.

Beispiel:



```
// 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,...)
```

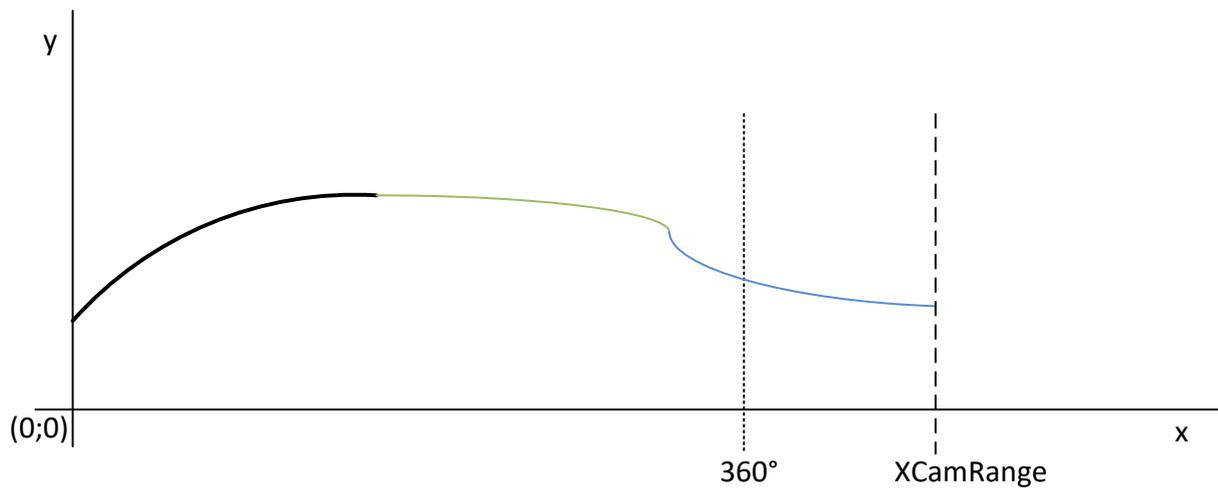
```
// With AutoAppend Option
tecCam.Coupling.Cam.DefineSegment(xFollower,1,1,MCTechnoCamSegmentTypes.AutoPoly1st, X1Offset,
MCTechnoCamOptions.NoOption, -1.0 * X1Offset, Y1Offset,...)
tecCam.Coupling.Cam.DefineSegment(xFollower,1,2,MCTechnoCamSegmentTypes.AutoPoly1st, X2Offset,
MCTechnoCamOptions.AutoAppend,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

2: UserDefinedMasterRange

Diese Option ist nur im 1. Segment verfügbar. Hier wird für die komplette Kurvenscheibe definiert, dass der Leitachsbereich der Kurvenscheibe ungleich des Modulobereichs der Leitachse ist.

Beispiel: Der Modulobereich der Leitachse ist 360°, die Kurvenscheibe der Folgeachse soll aber in einem Leitachsbereich von 540° definiert werden. In diesem Fall muss die Option „UserDefinedMasterRange“ ausgewählt werden.

- Soll im Gebrauch die Kurvenscheibe in Leitachsrichtung gestreckt/gestaucht werden, so muss auch hier die Option „UserDefinedMasterRange“ gewählt werden, auch wenn der angegebene MasterCamRange dem Modulo-Verfahrbereich der Leitachse entspricht.



```
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 Beispiele

Im nachfolgenden Beispiel werden 3 Kurvenscheiben definiert. Der Verlauf der Kurve ist identisch. Nur die Definitionen sind unterschiedlich:

CAM Sample 1: Definition der Kurvenscheibe mit Autopolynomen ohne weitere Optionen

CAM Sample 2: Definition der Kurvenscheibe mit Autopolynomen mit der Option „AutoAppend“

CAM Sample 3: Definition der Kurvenscheiben mit Polynomen mit der Angabe der Koeffizienten ohne weitere Optionen

```
// 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);

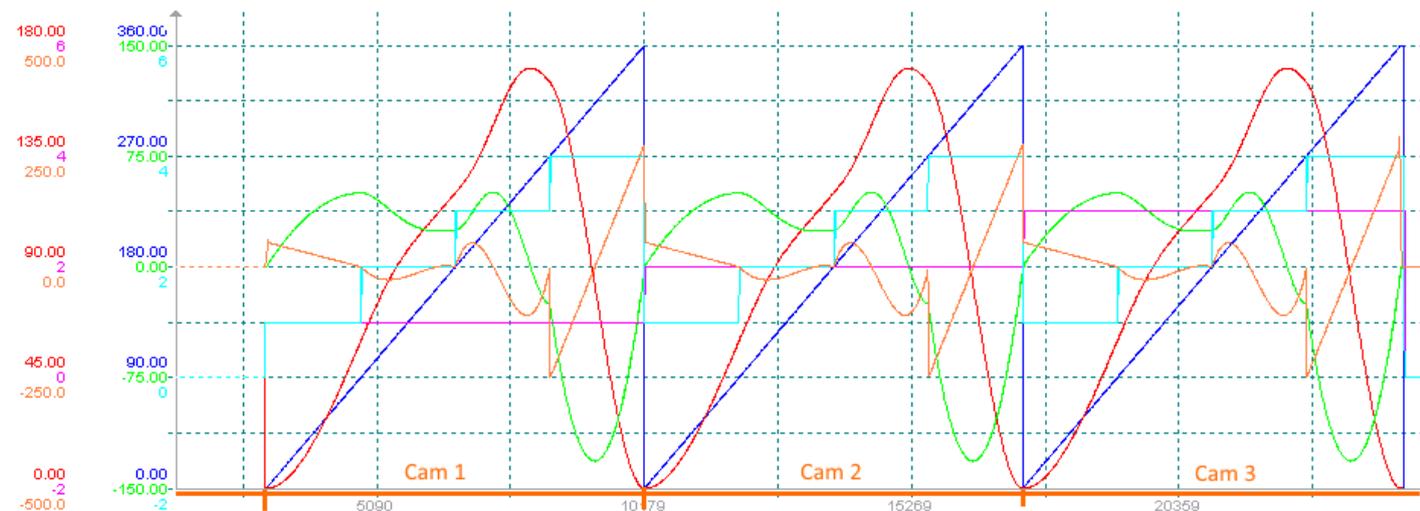
// CAM Sample 2
tecCam.Coupling.Cam.Create(xFollower, 2, 4);

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);
```

## Verlauf:



- Blau:** Sollposition der Leitachse
- Rot:** Sollposition der Folgeachse
- Grün:** Sollgeschwindigkeit der Folgeachse
- Orange:** Sollbeschleunigung der Folgeachse
- Pink:** Nummer des aktiven Kurvenscheibenprofils
- Hellblau:** Nummer des aktiven Kurvenscheibensegments

Im Oszilloskop ist deutlich zu sehen, dass trotz unterschiedlicher Definitionen der Kurvenverlauf identisch ist.

Der Wechsel der Kurvenscheiben ist an der Nummer des aktiven Kurvenscheibenprofils (pink) zu erkennen.

Die einzelnen Abschnitte innerhalb der Kurvenscheibe erkennt man nicht nur an der Nummer des aktiven Kurvenscheibensegments (hellblau) sondern z.B. auch an den Sprungstellen in der Beschleunigung (orange).

Sprungstellen in der Beschleunigung führen oft zu einem unruhigen Verhalten, welches durch die explizite Vorgabe der Beschleunigung an den Übergangsstellen bei der Verwendung des Polynoms 5. Grades vermeiden kann.

Spitzen im Beschleunigungsverlauf am Anfang und Ende der Bewegung spiegeln die Beschleunigung und Verzögerung der Leitachse wider.

## CAM Sample 4 ... 6: Verschiedene Definitionen einer Kurvenscheibe des Types „Sinus“.

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

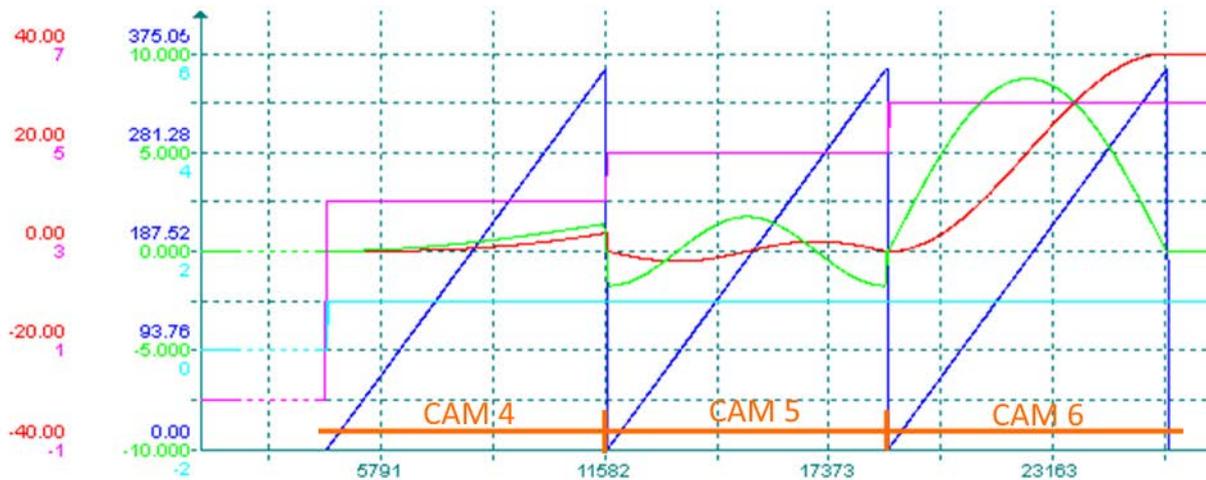
// CAM Sample 5: Pure Sine
tecCam.Coupling.Cam.Create(xFollower, 5, 1);
tecCam.Coupling.Cam.DefineSegment(xFollower, 5, 1, MCTechnoCamSegmentTypes.Sine, 0.0, MCTechnoCamOptions.NoOption,
0.0,
0.0,
0.0,
2.0*cPI, // Amplitude
```

```

2.0 / 360.0,
);

// CAM Sxample 6: Part of Sine
tecCam.Coupling.Cam.Create(xFollower, 6, 1);
tecCam.Coupling.Cam.DefineSegment(xFollower, 6, 1, MCTechnoCamSegmentTypes.Sine, 0.0, MCTechnoCamOptions.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,
);

```



Durch die Angabe unterschiedlicher Parameter bei der Definition eines Kurvenscheibensegments vom Typ „Sinus“ lassen sich unterschiedliche Abschnitte eines Sinus konfigurieren und beschreiben.

Wie bei CAM 4 kann dies z.B. als Beschleunigungssegment (bei CAM 5 als Sinus-oszillierende Bewegung und bei CAM 6 als Positionierung) verwendet werden.

Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg  
[www.jetter.de](http://www.jetter.de)

E-Mail [info@jetter.de](mailto:info@jetter.de)  
Telefon +49 7141 2550-0