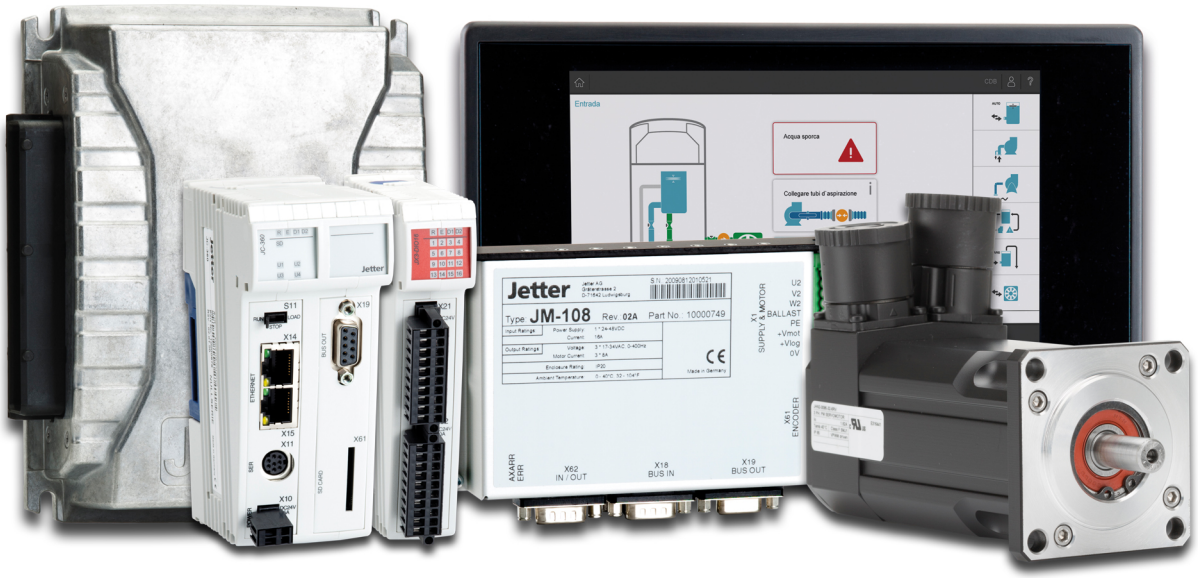
Jetter automation



Application Note - APN 047

Beschreibung des Motion-API-Befehls "Kurvenscheiben auslesen"

60878555

We automate your success.

Application Note: 047
Artikelnummer: 60878555
Version 1.01
Juni 2015 / Printed in Germany

Dieses Dokument hat die Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt.

Bei Änderungen, Weiterentwicklungen oder Erweiterungen bereits zur Verfügung gestellter Produkte wird ein überarbeitetes Dokument nur beigefügt, sofern dies gesetzlich vorgeschrieben oder von der Jetter AG für sinnvoll erachtet wird. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Die im Dokument aufgeführten Logos, Bezeichnungen und Produktnamen sind geschützte Marken der Jetter AG, der mit ihr verbundenen Unternehmen oder anderer Inhaber und dürfen nicht ohne Einwilligung des jeweiligen Inhabers verwendet werden.

Inhaltsverzeichnis

1 Auslesen von Kurvenscheiben 5

1.1 Definitionen der Kurvenscheibe 6
CamSegmentData 7
CamEvaluateData 11
CamAnalyseData 13
CamAnalyseDerivatesData 16

1 Auslesen von Kurvenscheiben

Einleitung

Das Kapitel beschreibt, wie Sie mit der Motion API die Kurvenscheiben auslesen. In der Motion API gibt es für Kurvenscheiben zwei Strukturen:

- Strukturen für die Definition von Kurvenscheiben
- Strukturen zum Auslesen von Kurvenscheiben

Inhalt

Thema	Seite
Definitionen der Kurvenscheibe	6

1.1 Definitionen der Kurvenscheibe

Einleitung

Die Motion-API-Bibliothek enthält Funktionen zur Auswertung von definierten Kurvenscheiben. Diese Funktionen geben Daten als Rückgabewert in entsprechenden Strukturen zurück. Das Kapitel beschreibt Details der verwendeten Strukturen.

Aufbau der Struktur

In der Motion-API haben Strukturen für das Auslesen von Kurvenscheiben folgende Namen:

```
{
    //...
    Struct CamSegmentData
    Struct CamEvaluateData
    Struct CamAnalyseData
    Struct CamAnalyseDerivatesData
    //...
}
```

Zugriff auf die Strukturen

Die Daten lesen Sie, indem Sie jeder Funktion der Kurvenscheibe die Profilnummer (CamID) und Segmentnummer (SegmentID) mitgeben. Um auf die Strukturen `MCTechnoCam` zuzugreifen, deklarieren Sie Variablen. Diese Variablen können Sie sich z. B. in einem Setup-Fenster anzeigen lassen.

```
Var
    MyCamSegmentData            : CamSegmentData;
    MyCamEvaluateData           : CamEvaluateData;
    MyCamAnalyseData            : CamAnalyseData;
    MyCamAnalyseDerivatesData   : CamAnalyseDerivatesData;
End_Var;
```

Inhalt

Thema	Seite
CamSegmentData	7
CamEvaluateData	11
CamAnalyseData	13
CamAnalyseDerivatesData	16

CamSegmentData

Einleitung

Der Rückgabewert der Funktion `ReadSegment` liefert folgende Struktur der Daten zum Auslesen:

- Beginn eines Segments
- Offsets der Leit- und Folgeachse
- Koeffizienten der Kurvenscheibe
- Leitachsbereich

Wenn Sie Daten prüfen, lesen Sie z. B. mit der Struktur `CamSegmentData` die Koeffizienten eines vorher definierten Polynoms oder lesen die Offset-Position einer Folgeachse aus.

Korrespondierender Motion-API-Befehl

Die Struktur `CamSegmentData` korrespondiert mit folgender Funktion:

- MCTechnoCam.**ReadSegment**()

`ReadSegment` ist für die Kontrolle von definierten Kurvenscheiben gedacht. Damit können Sie prüfen, ob der Verlauf der Kurvenscheibe sich wie geplant verhält. Sie können in JetSym z. B. diesen Verlauf auslesen und grafisch darstellen.

Definition der Struktur CamSegmentData

Die Struktur `CamSegmentData` ist in der Motion-API-Datei **MCTechnoCamStructs.stxp** definiert.

```
CamSegmentData : Struct
    SegmentType    : Int;
    SegmentStart   : Double;
    Option         : Int;
    MasterOffset   : Double;
    FollowerOffset : Double;
    Coefficient    : Array[0..7] Of Double;
    MasterCamRange : Double;
End_Struct;
```

Speicherstruktur CamSegmentData

Members	Beschreibung	Datentyp
SegmentType	Typ der Segmentfunktion: 1: Gerade 2: Polynom 3. Grads 3: Polynom 5. Grads 4: Polynom 7. Grads 5: Autopolynom 7. Grads 6: Sin 7: Autopolynom 3. Grads 8: Autopolynom 5. Grads 9: Autopolynom 1. Grads (Gerade)	Int
SegmentStart	Beginn des Segments	Double
Optionen	Mögliche Optionen: 0: Keine (default) 1: Segment automatisch anhängen 2: Anwenderspezifischer Leitachs-CAM-Bereich	Int
MasterOffset	Offset der Leitachse	Double
FollowerOffset	Offset der Folgeachse	Double
Coefficient	0: 1. Koeffizient/ Leitachsbereich 1: 2. Koeffizient 2: 3. Koeffizient 3: 4. Koeffizient 4: 5. Koeffizient 5: 6. Koeffizient 6: 7. Koeffizient 7: 8. Koeffizient	Double
MasterCamRange	Leitachsbereich	Double

Die Funktion `MCTechnoCam.ReadSegment()` liest Daten und speichert diese als Rückgabewert für die Struktur `CamSegmentData`.

Hinweise

- Bei der AutopolyX-Funktion werden nicht die Randparameter (Ableitungen), sondern die Polynomkoeffizienten rückgelesen.
- Für das erste Segment wird der Leitachsbereich gelesen, für weitere Segmente der Leitachsmodulobereich.

Beispiel bei Geradengleichung

Sie haben eine Kurvenscheibe mit dem Funktionstyp **Gerade** definiert. Wenn Sie die Koeffizienten rücklesen, dann sind in `CamSegmentData.Coefficient[...]` zwei Koeffizienten für die Funktion gespeichert.

$$y(x) = a \cdot x + b$$

Gespeicherte Koeffizienten: a, b

Beispiel bei Polynom 7. Grads

Sie haben eine Kurvenscheibe mit den Funktionstyp **Polynom 7. Grads** definiert. Wenn Sie die Koeffizienten rücklesen, dann sind in **C**`amSegmentData.Coefficient[...]` acht Koeffizienten für die Funktion gespeichert.

$$y(x) = ax^7 + bx^6 + cx^5 + dx^4 + ex^3 + fx^2 + gx + h$$

Gespeicherte Koeffizienten: a, b, c, d, e, f, g, h

Reihenfolge der Koeffizienten

Die Koeffizienten sind in aufsteigender Reihenfolge der Potenz gespeichert, also enthält z. B. bei einem **Polynom 7. Grads** der achte Koeffizient die höchste Potenz:

```
CamSegmentData.Coefficient[0] = Koeffizient h
CamSegmentData.Coefficient[1] = Koeffizient g
...
CamSegmentData.Coefficient[6] = Koeffizient b
CamSegmentData.Coefficient[7] = Koeffizient a
```

Bei einer **Geraden** sind die Koeffizienten wie folgt gespeichert:

```
CamSegmentData.Coefficient[0] = Koeffizient b
CamSegmentData.Coefficient[1] = Koeffizient a
```

Nicht benötigte Koeffizienten enthalten den Wert Null.

JetSym-STX-Programm

Sie haben mit dem Hardware-Manager einen Technologieverbund `tecGruppe` aus der Leitachse `axLeitachse` und Kurvenscheiben-Folgeachse `axFolgeachse` angelegt. Das kurze Programm gibt Werte des Segments aus:

```
Task tReadSegment
Var
    CamSegData:     CamSegmentData;
    i:              Int;
End_Var;

// Lesen der Daten 'ReadSegment':
tecGruppe.Coupling.Cam.ReadSegment(axFolgeachse,1,1);

// Ausgeben der Daten:
Trace('Segmenttyp: ' + FloatToStr(CamSegData.SegmentType));
Trace('Segmenstart: ' + FloatToStr(CamSegData.SegmentStart));
Trace('Optionen: ' + FloatToStr(CamSegData.Option));
Trace('Offset der Leitachse: ' +
FloatToStr(CamSegData.MasterOffset));
Trace('Offset der Folgeachse: ' +
FloatToStr(CamSegData.FollowerOffset));
For i := 0 To 7 Do
    Trace('Koeffizient ' + IntToStr(i) + ' :' +
    FloatToStr(CamSegData.Coefficient[i]));
End_For;
Trace('Leitachsbereich: ' +
FloatToStr(CamSegData.MasterCamRange));
End_Task;
```

CamEvaluateData

Einleitung

Mit der Struktur lassen sich folgende Daten auslesen:

- Position der Folgeachse bei vorgegebener Leitachsposition
- Segmentnummer
- Bereich der Leitachse

Korrespondierender Motion-API-Befehl

Die Struktur `CamEvaluateData` korrespondiert mit folgender Funktion:

- MCTechnoCam.**Evaluate**()

Der Motion-API-Befehl `Evaluate` liest die Position der Folgeachse aus. Sie geben die Position der Leitachse vor.

Definition der Struktur CamEvaluateData

Die Struktur `CamEvaluateData` ist in der Motion-API-Datei **MCTechnoCamStructs.stxp** definiert.

```
CamEvaluateData : Struct
    FollowerPosition  : Double;
    SegmentNr         : Int;
    MasterCamRange    : Double;
End_Struct;
```

Speicherstruktur CamEvaluateData

Members	Beschreibung	Datentyp
FollowerPosition	Position der Folgeachse	Double
SegmentNr	Beginn des Segments	Int
MasterCamRange	Leitachsbereich	Double

Die Funktion `MCTechnoCam.Evaluate()` liest Daten und speichert diese Daten als Rückgabewert in die Struktur `CamEvaluateData`.

JetSym-STX-Programm

Sie haben mit dem Hardware-Manager einen Technologieverbund `tecGruppe` aus der Leitachse `axLeitachse` und Kurvenscheiben-Folgeachse `axFolgeachse` angelegt. Der Task liest Daten der Folgeachse bei vorgegebener Leitachsposition aus.

```
Var
    CamEvalData:        CamEvaluateData;
    tecGruppe:          Pointer To MCTechno;
End_Var;

Task tEvalData
// Coupling.Cam.Evaluate wird mit folgenden Parametern aufgerufen:
// Coupling.Cam.Evaluate(FollowerAxis, CamId, MasterPosition)
// Die Leitachsposition wird vorgegeben, die Folgeachsposition,
// Segmentnummer und Leitachsbereich werden ausgelesen.
// Aufruf der Funktion:
// Auslesen der Werte für Kurvenscheibe 2 an Leitachsposition 150.0:
CamEvalData:= tecGruppe.Coupling.Cam.Evaluate(axFolgeachse, 2,
150.0);
// Ausgabe der Ergebnisse:
Trace('Position der Folgeachse: '+
FloatToStr(CamEvalData.FollowerPosition);
Trace('SegmentNr: ', + IntToStr(CamEvalData.SegmentNr));
Trace('Leitachsbereich: ' +
FloatToStr(CamEvalData.MasterCamRange));
End_Task;
```

CamAnalyseData

Einleitung

Mit der Struktur lassen sich folgende Daten auslesen:

- Beginn eines Segments
- Gewählte Option im analysierten Segment
- Offsets der Leitachse
- Offset der Folgeachse
- Minimalwert des Segments
- Maximalwert des Segments

Korrespondierender Motion-API-Befehl

Die Struktur `CamAnalyseData` korrespondiert mit folgender Funktion:

- MCTechnoCam.**CamAnalyseSegment()**

Definition der Struktur CamAnalyseData

Die Struktur `CamAnalyseSegment` ist in der Motion-API-Datei **MCTechnoCamStructs.stxp** definiert.

```
CamAnalyseData : Struct
    SegmentType    : Int;
    SegmentStart   : Double;
    Option         : Int;
    MasterOffset   : Double;
    FollowerOffset : Double;
    MinValue       : Double;
    MaxValue       : Double;
End_Struct;
```

Speicherstruktur CamAnalyseData

Members	Beschreibung	Datentyp
SegmentType	Typ der Segmentfunktion: 1: Gerade 2: Polynom 3. Grads 3: Polynom 5. Grads 4: Polynom 7. Grads 5: Autopolynom 7. Grads 6: Sin 7: Autopolynom 3. Grads 8: Autopolynom 5. Grads 9: Autopolynom 1. Grads (Gerade)	Int
SegmentStart	Beginn des Segments	Double
Optionen	Mögliche Optionen: 0: Keine (default) 1: Segment automatisch anhängen 2: Anwenderspezifischer Leitachsbereich	Int (0 ... 2)
MasterOffset	Offset der Leitachse	Double
FollowerOffset	Offset der Folgeachse	Double
MinValue	Kleinster Wert im Segment	Double
MaxValue	Größter Wert im Segment	Double

Die Funktion `MCTechnoCam.AnalyseSegment()` liest Werte für die Struktur `CamAnalyseData` aus.

JetSym-STX-Programm

Sie haben mit dem Hardware-Manager einen Technologieverbund `tecGruppe` aus der Leitachse `axLeitachse` und Kurvenscheiben-Folgeachse `axFolgeachse` angelegt.

```
Task tAnalyseCAM
Var
    CamAnalyseSegData:    CamAnalyseData;
End_Var;

// Aufruf der Funktion
// Auslesen der Werte für Kurvenscheibe 1, Segment 1:
tecGruppe.Coupling.Cam.AnalyseSegment(axFolgeachse,1,1);
// Auswertung:
Trace('Segmenttyp: ' + IntToStr(CamAnalyseSegData.SegmentType));
Trace('Beginn des Segments: ' +
FloatToStr(CamAnalyseSegData.SegmentStart));
Trace('Option: ' + IntToStr(CamAnalyseSegData.Option ));
Trace('Offset der Leitachse: ' +
FloatToStr(CamAnalyseSegData.MasterOffset));
Trace('Offset der Folgeachse: ' +
FloatToStr(CamAnalyseSegData.FollowerOffset));
Trace('Kleinster Wert im Segment: ' +
FloatToStr(CamAnalyseSegData.MinValue));
Trace('Größter Wert im Segment: ' +
FloatToStr(CamAnalyseSegData.MaxValue));
End_Task;
```

CamAnalyseDerivatesData

Einleitung

Mit der Struktur lassen sich folgende Daten auslesen:

- Beginn eines Segments
- Gewählte Option im analysierten Segment
- Offsets der Leitachse
- Offset der Folgeachse
- Minimalwerte der Ableitungen
- Maximalwerte der Ableitungen
- Leitachsbereich

Anwendungsbeispiel

Wenn Sie den Funktionstyp **Autopolynom** in der Definition der Kurvenscheibe verwendet haben, lesen Sie segmentweise die Ableitungen aus. Auf diese Weise prüfen Sie z. B. die Stetigkeit der Funktion in einem Segment.

Korrespondierender Motion-API-Befehl

Die Struktur `CamAnalyseDerivatesData` korrespondiert mit folgender Funktion:

- MCTechnoCam.**AnalyseSegmentDerivates()**

Definition der Struktur CamAnalyseDerivatesData

Die Struktur `CamAnalyseDerivatesData` ist in der Motion-API-Datei **MCTechnoCamStructs.stxp** definiert.

```
CamAnalyseDerivatesData: Struct
    SegmentType    : Int;
    SegmentStart   : Double;
    Option         : Int;
    MasterOffset   : Double;
    FollowerOffset : Double;
    MinValue       : Array[0..2] Of Double;
    MaxValue       : Array[0..2] Of Double;
    MasterCamRange : Double;
End_Struct;
```

Speicherstruktur CamAnalyseDerivates Data

Members	Beschreibung	Datentyp
SegmentType	Typ der Segmentfunktion: 1: Gerade 2: Polynom 3. Grads 3: Polynom 5. Grads 4: Polynom 7. Grads 5: Autopolynom 7. Grads 6: Sin 7: Autopolynom 3. Grads 8: Autopolynom 5. Grads 9: Autopolynom 1. Grads (Gerade)	Int
SegmentStart	Beginn des Segments	Double
Optionen	Mögliche Optionen: 0: keine (Default) 1: Segment automatisch anhängen 2: anwenderspezifischer Leitachs-CAM-Bereich	Int (0 ... 2)
MasterOffset	Offset der Leitachse	Double
FollowerOffset	Offset der Folgeachse	Double
MinValue	Kleinster Wert der Ableitungen	Array[0..2] of Double
MaxValue	Größter Wert der Ableitungen	Array[0..2] of Double

Die Funktion `MCTechnoCam.AnalyseSegmentDerivates()` liest Werte für die Struktur `CamAnalyseDerivates` aus.

JetSym-STX-Programm

Sie haben mit dem Hardware-Manager einen Technologieverbund `tecGruppe` aus der Leitachse `axLeitachse` und Kurvenscheiben-Folgeachse `axFolgeachse` angelegt.

```
Task tAnalyseCAMDerivates

Var
  CamAnalyseDerivates:    CamAnalysederivatesData;
  i:                      Int;
End_Var;

// Aufruf der Funktion
// Auslesen der Werte für Kurvenscheibe 1, Segment 1:
tecGruppe.Coupling.Cam.AnalyseSegmentDerivates(axFolgeachse,1,1)
;

// Auswertung:
Trace('Segmenttyp: ' +
IntToStr(CamAnalyseDerivates.SegmentType));
Trace('Beginn des Segments: ' +
FloatToStr(CamAnalyseDerivates.SegmentStart));
Trace('Option: ' + IntToStr(CamAnalyseDerivates.Option));
Trace('Offset der Leitachse: ' +
FloatToStr(CamAnalyseDerivates.MasterOffset));
Trace('Offset der Folgeachse: ' +
FloatToStr(CamAnalyseDerivates.FollowerOffset));
For i := 0 To 2 Do
    Trace('Kleinster Wert der ' + IntToStr(i) + ' Ableitung ist: '
    + FloatToStr(CamAnalyseDerivates.MinValue[i]));
    Trace('Größter Wert der ' + IntToStr(i) + ' Ableitung ist: ' +
    FloatToStr(CamAnalyseDerivates.MaxValue[i]));
End_For;
End_Task;
```

Jetter automation

Jetter AG
Gräterstraße 2
71642 Ludwigsburg | Germany

Tel +49 7141 2550-0
Fax +49 7141 2550-425
info@jetter.de
www.jetter.de

We automate your success.