SIMOTION library LCamHdl

applications & TOOLS

Creation of cam disks at runtime

Version 1.3.3

SIEMENS

Warranty and liability

Note The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications − e.g. Catalogs − the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

User manual

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under http://www.siemens.com/industrialsecurity.

Table of contents

Warranty and liability **2**

1 Configuration of cam disks **6**

1.1 Tasks of the unit **6**

1.2 Fundamental explanations by means of a linear/circular guided output component **6**

1.2.1 Motion scheme 8
1.2.2 Laws of motion – motion transitions 11
1.2.3 Subdivision of laws of motion 13
1.2.4 Symmetrically / asymmetrically standardized laws of motion 13
1.2.5 Selection of laws of motion 14

2 Unit functions **15**

2.1 Application examples **17**

2.1.1 Example 1 17
2.1.2 Example 2 20

2.2 Function elements and integration **23**

2.3 Function block FBLCamHdlCreateCam **24**

2.3.1 Functionality 24
2.3.2 Schematic representation in LAD 24
2.3.3 Input and output parameters 25
2.3.4 Explanations to the different profile types 28
2.3.5 Time sequence chart 34
2.3.6 Error numbers 35

2.4 Function block FBLCamHdlCalcCamMinMaxValues **37**

2.4.1 Functionality 37
2.4.2 Schematic representation in LAD 37
2.4.3 Input and output parameters 38
2.4.4 Time sequence chart 39
2.4.5 Error numbers 40

2.5 Function block FBLCamHdlCalcAxisDynamics **41**

2.5.1 Functionality 41
2.5.2 Schematic representation in LAD 42
2.5.3 Input and output parameters 43
2.5.4 Time sequence chart 44
2.5.5 Error numbers 44

3 Definition of standardized laws of motion **45**

4 Continuity check and correction **48**

5 Literature index 50
6 Alterations / Author 50
7 Contacts 51

1 Configuration of cam disks

1.1 Tasks of the unit

The present application is to support the user with the configuration of cam disks (cam gears) by applying the laws of motion.

In general, cam disks are electronic gears at a non-constant transition, where, for example, a constant drive motion is converted into a non-constant drive motion by applying the laws of motion.

Within the scope of SIMOTION, there are two ways to configure cam disks:

- during the configuration in the engineering system SCOUT by means of a cam disk editor (CamEdit or CamTool)
- at runtime by means of the system functions _addPointToCam, _addPolynomialSegmentToCam, _addSegmentToCam and _interpolateCam

The present application is dedicated to the configuration of cam disks during the runtime by means of the runtime system.

1.2 Fundamental explanations by means of a linear/circular guided output component

Laws of motion mean analytic functions, which usually describe the drive motion in relation to the output motion.

Figure 1-1: Drive – output motion

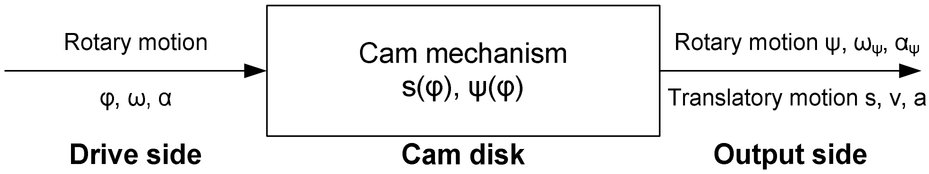
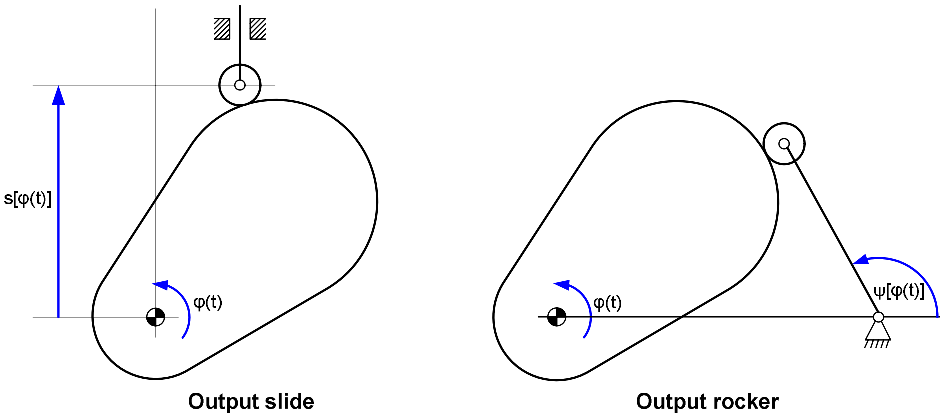


Figure 1-2: Differentiation of the output motion



Output slide: The rotational drive motion is converted into a sliding motion s[φ(t)].

Output rocker: The rotational drive motion is converted into a rotational motion ψ[φ(t)].

Motion equations

The prime ´ designates the deviation of the rotation angle φ of the drive, the dot · designates deviation according to the time t.

Drive component

Rotation angle	$\varphi(t)$
Angle velocity	$\dot{\varphi} = \frac{d\varphi}{dt} = \omega$
Angle acceleration	$\ddot{\varphi} = \frac{d\omega}{dt} = \alpha$

Drive slide

Travel	$s[\varphi(t)]$
Velocity	$v = \frac{ds}{dt} = \frac{ds}{d\varphi} \cdot \frac{d\varphi}{dt} = s' \cdot \omega$

Acceleration	$a = \frac{dv}{dt} = \frac{d}{dt}(s' \cdot \omega) =$ $\frac{ds'}{d\varphi} \cdot \frac{d\varphi}{dt} \cdot \omega + s' \cdot \frac{d\omega}{dt} =$ $s'' \cdot \omega^2 + s' \cdot \alpha$

Output rocker

Angle $\psi[\varphi(t)]$

Angle velocity $\omega_\psi = \frac{d\psi[\varphi(t)]}{dt} = \psi' \cdot \dot{\varphi} = \psi' \cdot \omega$

Acceleration $\alpha_\psi = \frac{d^2\psi[\varphi(t)]}{dt} = \psi'' \cdot \omega^2 + \psi' \cdot \alpha$

Special case of drive running at a constant rotational velocity (ω=constant, α = 0) →

Table 1-1: Output at a constant rotational velocity

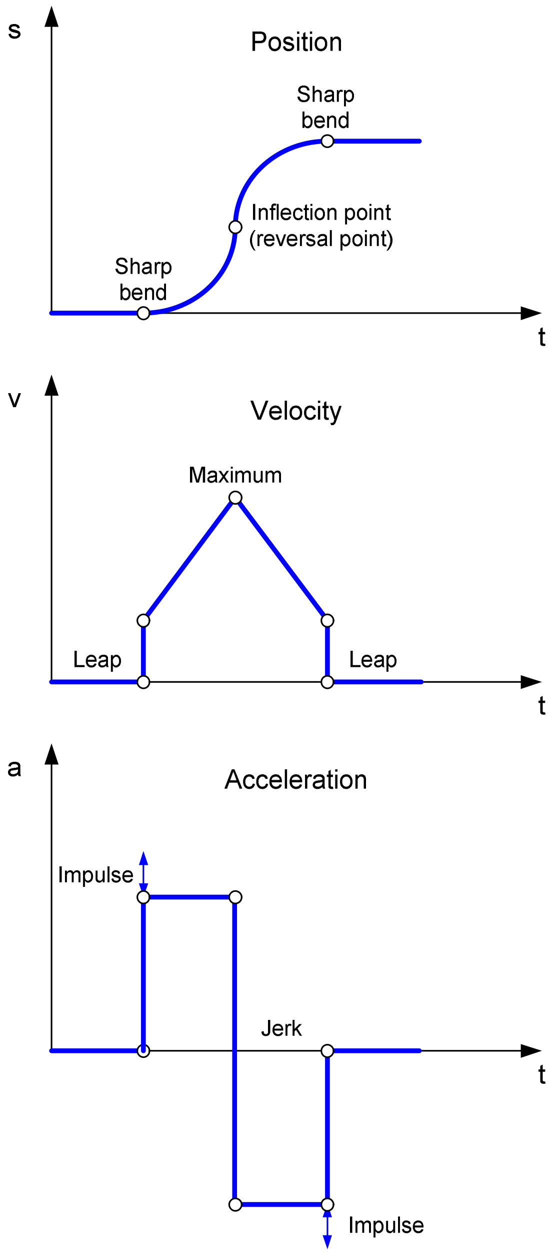
	Output slide	**Output rocker**
Travel/angle	$s[\varphi(t)]$	$\psi[\varphi(t)]$
Velocity/ Angle velocity	$v = s' \cdot \omega$	$\omega_\psi = \psi' \cdot \omega$
Acceleration/angle acceleration	$a = s'' \cdot \omega^2$	$\alpha_\psi = \psi'' \cdot \omega^2$

1.2.1 Motion scheme

Taking into consideration the given requirements concerning the motions of the output component, a motion scheme is generated to visualize the complete function of the transition.

To achieve a bumpless and jerk-free drive motion, a transition function without any break points or inflection points is required (see Figure 1-3).

Figure 1-3: Example of a bumpless and jerk-free motion



For the calculation according to a certain system, this motion scheme is divided into single sections (segments) which are again a partial section of the transition function (PTF).

The following example of an output slide is to make this clear.

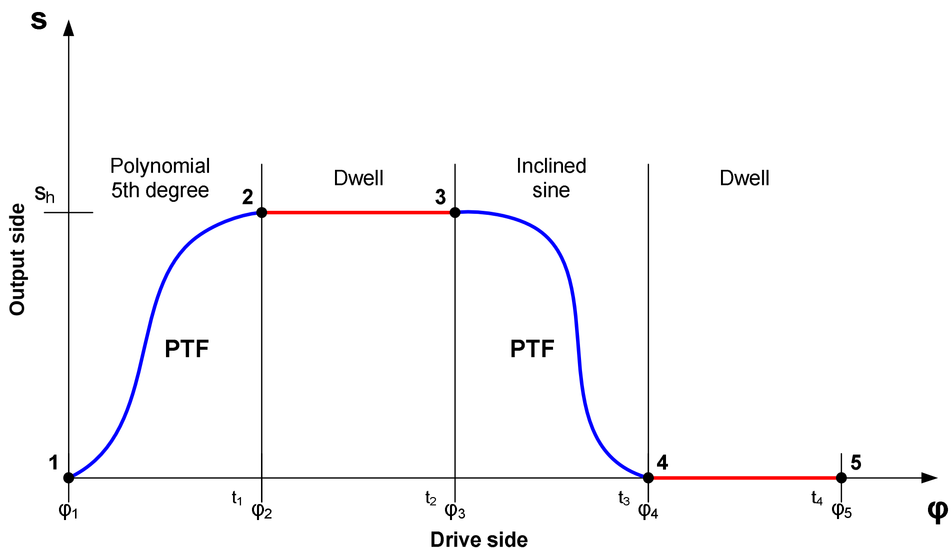
The limit points of the single segments are determined by the according distance ‚s' (output) and the circular angle φ (drive).

Example:

After a stroke (s_h) in a given time (t_1), the output component shall stop for a defined time (t_2) and then move to the opposite direction (negative stroke) in a given time (t_3) and stop for a certain time (t_4).

For this example, the laws of motion have been chosen in accordance to the criteria of the motion transitions being described in the following.

Figure 1-4: Motion diagram



1.2.2 Laws of motion – motion transitions

Considering the generally applied terms, the drive range is designated as master (x-axis) and the output range as slave (y-axis) in the following.

The limit points can be categorized by the velocity and the acceleration.

Table 1-1: Limit values of partial transition functions

Motion task	Limit values	
	Velocity	Acceleration
Dwell	$v = 0$	$a = 0$
Constant velocity	$v \neq 0$	$a = 0$
Reversal	$v = 0$	$a \neq 0$
Motion	$v \neq 0$	$a \neq 0$

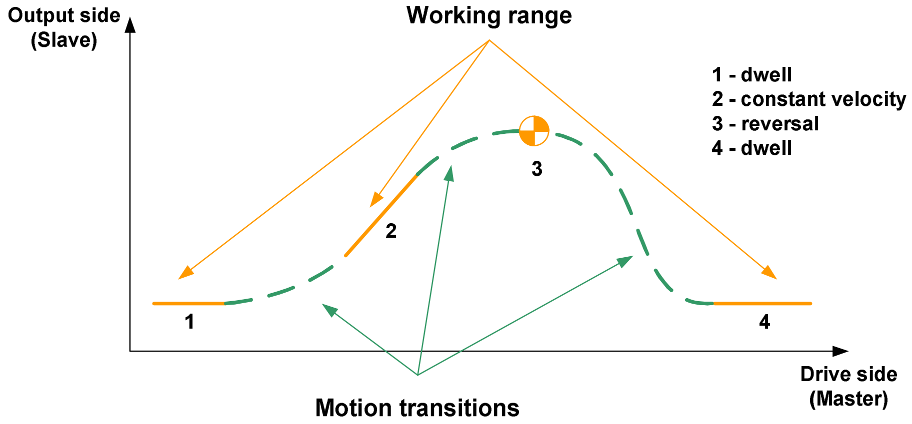
Table 1-2 shows the 16 resulting possibilities for motion transitions.

Table 1-2: Combinations of motion transitions

	D (dwell)	**C** (constant velocity)	**R** (reversal)	**M** (motion)
D (dwell)	v=0 a=0 v=0 a=0	v=0 a=0 v≠0 a=0	v=0 a=0 v=0 a≠0	v=0 a=0 v≠0 a≠0
C (constant velocity)	v≠0 a≠0 v=0 a=0	v≠0 a=0 v≠0 a=0	v=0 a≠0 v=0 a≠0	v≠0 a=0 v≠0 a≠0
R (reversal)	v=0 a≠0 v=0 a=0	v=0 a≠0 v≠0 a=0	v=0 a≠0 v=0 a≠0	v=0 a≠0 v≠0 a≠0
M (motion)	v≠0 a≠0 v=0 a=0	v≠0 a≠0 v≠0 a=0	v≠0 a≠0 v=0 a≠0	v≠0 a≠0 v≠0 a≠0

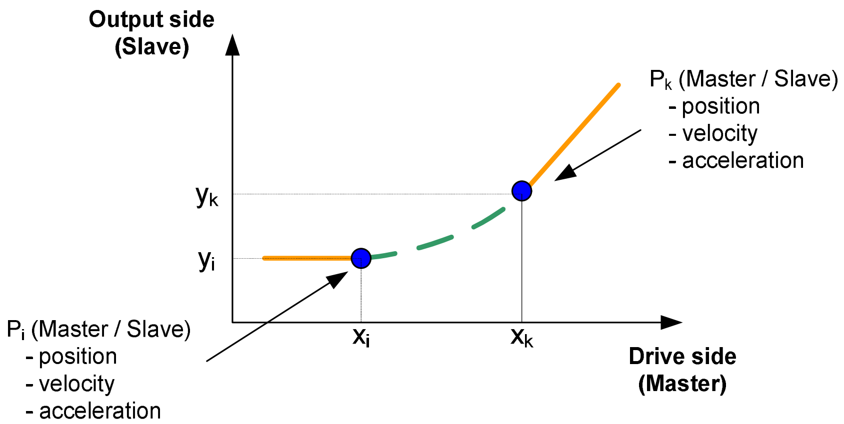
Segments for cam disks (partial transition functions) consist of working ranges and motion transitions. The working ranges are determined by the technological sequence within the machine.

Figure 1-5: working ranges and motion transitions for cam disks



The motion transitions between the working ranges have to comply with certain limit conditions (Figure 1-6), e.g. motion transition at a continuous velocity and acceleration. This guarantees that the drive runs smoothly, e.g. without jerks. The segments are described by mathematic functions between start and end point, e.g. polynomials.

Figure 1-6: limit conditions of an element



1.2.3 Subdivision of laws of motion

The laws of motion can be subdivided into:

- Power laws

$$f(z)=\sum_{i=0}^{n} A_i z^i = A_0 + A_1 z + A_2 z^2 + A_3 z^3 + \ldots + A_i z^i$$

- Trigonometry laws

$$f(z)=A\cos(vz)+B\sin(vz)$$

- Combination of those two mentioned above

1.2.4 Symmetrically / asymmetrically standardized laws of motion

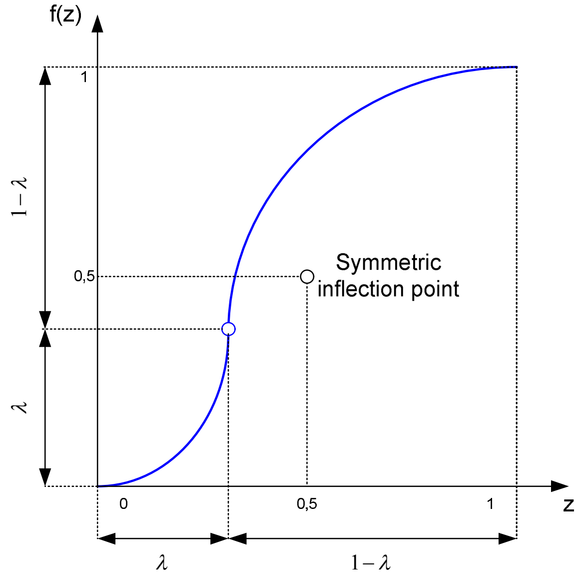
Symmetric laws of motion are characterized by an inflection point at $\lambda = 0{,}5$ within the standardized master range (0-1) – see chapter 3 of the annex.

Asymmetric laws of motion are created by displacing the inflection point on the straight line f(z) = z within the standardized law of motion. Resulting from this, there are different maximum values for the acceleration of the ranges

$0 \le z \le \lambda$ and

$\lambda \le z \le 1$

Figure 1-7: Asymmetric law of motion by displacing the inflection point



1.2.5 Selection of laws of motion

Table 1-2 shows appropriate laws of motion for the 16 possible transitions of motion according to VDI 2143, without any judgment of quality.

Table 1-3: Appropriate laws of motion for the possible motion transitions

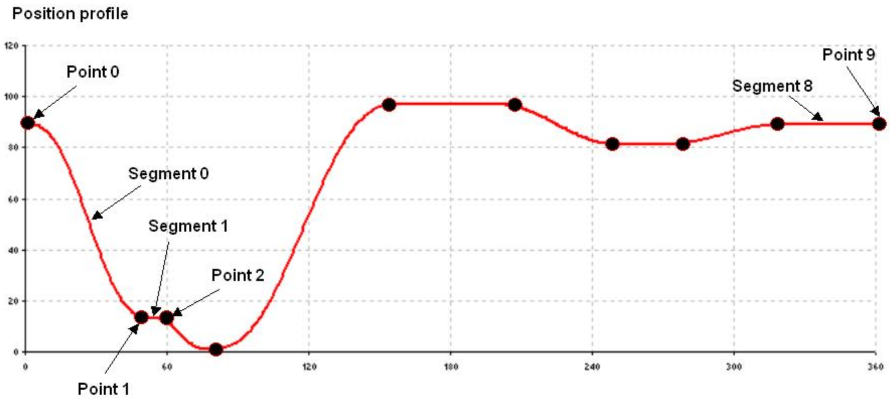
	Dwell **D**	**Constant velocity** **C**	**Reversal** **R**	**Motion** **M**
Dwell **D**	- Straight line - Quadratic parabola - Basic sine line - 5th degree polynomial - Inclined sine line - Mod. accel.trapezoid - Modified sine line	- 5th degree polynomial - Modified sine line	- Mod. accel. trapezoid - 5th degree polynomial - Harmonic combination	- 5th degree polynomial
Constant velocity **C**	- 5th degree polynomial - Modified sine line	- Straight line - 5th degree polynomial - Modified sine line	- Harmonic combination - 5th degree polynomial	- 5th degree polynomial
Reversal **R**	- Mod. accel.trapezoid - 5th degree polynomial - Harmonic combination	- Harmonic combination - 5th degree polynomial	- Straight sine line combination (velocity trapezoid) - Basic sine line - 5th degree polynomial	- 5th degree polynomial
Motion **M**	-5th degree polynomial	- 5th degree polynomial	- 5th degree polynomial	- 5th degree polynomial

You can find a detailed explanation of the above described circumstances in the guideline „VDI 2143 Motion rules for cam mechanisms“ [Literature 1].

2 Unit functions

SIMOTION makes it possible to generate complex curve disks from n segments with (n+1) points during the runtime. For this, you can freely choose the law of motion for each single segment. You can find the example of a generated cam disk in the figure below:

Figure 2-1: cam disk consisting of several segments



If the working ranges change during the operation of a machine by, for example, a changed length of the product, it is necessary to adapt the motion transitions as well.

During the runtime, you can use the function block **FBLCamHdlCreateCam** to merge working ranges and motion transitions into one cam disk. Unlike the system function _addSegmentToCam, the FBs can be used without having to calculate the polynomial coefficient before.

It is not necessary either to standardize the single sections (segments) with the parameterization (see chapter 3 of the annex) – thus, the user works in the real section. This means that the necessary calculations for the standardization – in general, the FB is based on the equations according to VDI 2143 – are executed within the function block and do not have to be considered by the user.

Due to these facts, the user transfers directly the position as well as the geometric deviation from the **real section** (e.g. velocity, acceleration).

There are different mathematic functions available for the motion transitions (segments), subsequently called profile types. In addition to the already mentioned polynomials

- 3rd degree polynomial,
- 5th degree polynomial,
- 7th degree polynomial

the further profiles exist

- straight line,
- quadratic parabola,
- basic sine,
- inclined sine,
- modified acceleration trapezoid,
- modified sine,
- sine-straight line-combination – velocity trapezoid,
- harmonic combination

In addition to that, it is also possible to transfer single support points which makes it possible to generate cam disks with combined ranges consisting of transition functions and others consisting of single points.

As only certain profile types can be logically applied to the different motion transitions, only those certain profiles are available. The polynomials can always be applied to all motion transitions by getting a more or less good result. This is why no differentiation is made at this point.

2.1 Application examples

The examples require a cam disk named Cam_1 in the runtime system of SIMOTION.

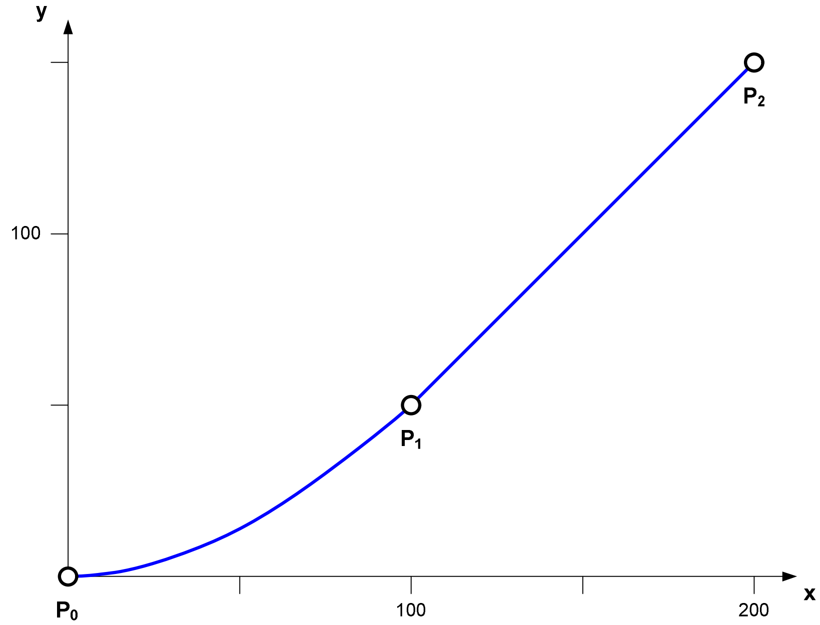
2.1.1 Example 1

Segment 1: dwell – constant velocity → 3rd degree polynomial

Segment 2: constant velocity → straight line

P_0:	$X_0 = 0$	$Y_0 = 0$	$m_0 = 0$
P_1:	$X_1 = 100$	$Y_1 = 50$	$m_1 = 0.75$
P_2:	$X_2 = 200$	$Y_2 = ?$	$m_2 = m_1$

Figure 2-2 – sketch example 1



$$Y_2 = Y_1 + (X_2 - X_1) \cdot m_1 = 50 + 100 \cdot 0.75 = 125$$

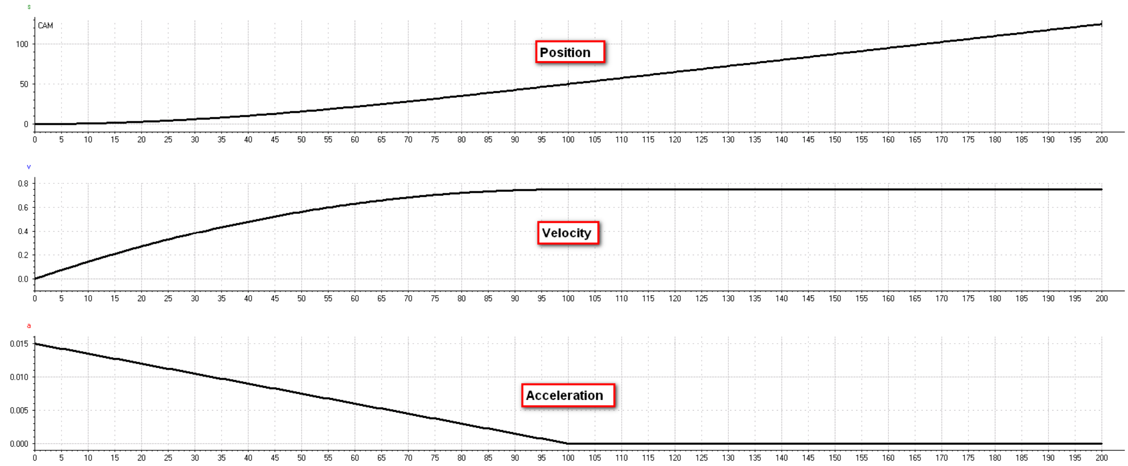
Source code

```
VAR
    instFBCreateCam    : FBLCamHdlCreateCam;
    sCamProfile        : sLCamHdlCamProfileType;
    boCalcOK           : BOOL;
END_VAR

// Initialize CAM
sCamProfile.eInterpolationMode   := LINEAR;
sCamProfile.r64MasterScale       := 0;
sCamProfile.r64SlaveScale        := 0;
sCamProfile.r64MasterShift       := 0;
sCamProfile.r64SlaveShift        := 0;

// Definition 1st segment
sCamProfile.asCamElement[0].r64InflectionPointParameter            := 0.5;
sCamProfile.asCamElement[0].eCamSegmentType                        := POLY_3;

sCamProfile.asCamElement[0].sSegmentParameter.r64MasterPosStart    := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64MasterPosEnd      := 100;
sCamProfile.asCamElement[0].sSegmentParameter.r64SlavePosStart     := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64SlavePosEnd       := 50.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoVeloStart      := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoVeloEnd        := 0.75;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoAccelStart     := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoAccelEnd       := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoJerkStart      := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoJerkEnd        := 0.0;

// Definition 2nd segment
sCamProfile.asCamElement[1].r64InflectionPointParameter            := 0.5;
sCamProfile.asCamElement[1].eCamSegmentType                        := CONST_VELO;

sCamProfile.asCamElement[1].sSegmentParameter.r64MasterPosStart    := 100.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64MasterPosEnd      := 200.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64SlavePosStart     := 50.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64SlavePosEnd       := 125.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoVeloStart      := 0.75;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoVeloEnd        := 0.75;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoAccelStart     := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoAccelEnd       := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoJerkStart      := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoJerkEnd        := 0.0;

sCamProfile.i16NumberOfElements := 2;


instFBCreateCam(execute          := TRUE
                ,reset           := FALSE
                ,toCam           := Cam_1
                ,camProfile      := sCamProfile);
```

Cam disk

You can see the generated cam in the following figure.

Figure 2-3: cam disk: curve of position, velocity and acceleration



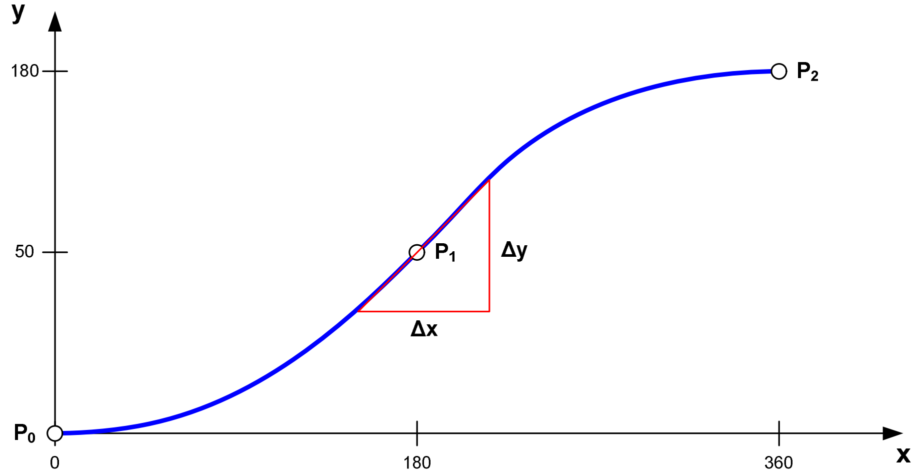
2.1.2 Example 2

Segment 1: dwell – constant velocity → modified sine line

Segment 2: constant velocity – dwell → 5th degree polynomial

P_0:	$X_0 = 0$	$Y_0 = 0$	$m_0 = 0$
P_1:	$X_1 = 180$	$Y_1 = 50$	$m_1 = ?$
P_2:	$X_2 = 360$	$Y_2 = 100$	$m_2 = 0$

Figure 2-4 – sketch example 2



$$m_1 = \frac{\Delta y}{\Delta x} = \frac{11}{20} = 0{,}55$$ (exemplary)

Source code:

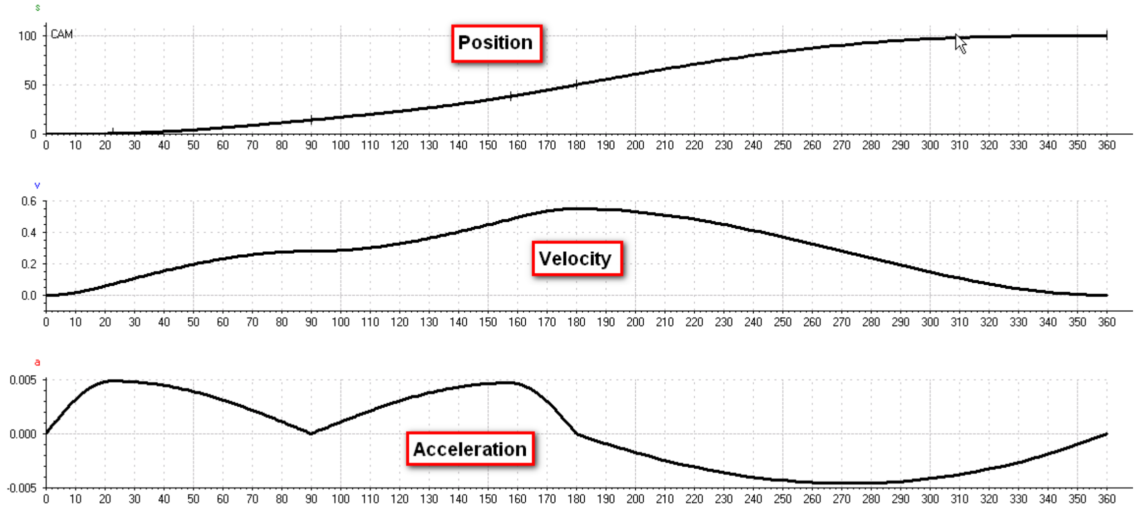
```
VAR
    instFBCreateCam    : FBLCamHdlCreateCam;
    sCamProfile        : sLCamHdlCamProfileType;
    boCalcOK           : BOOL;
END_VAR

// Initialize CAM
sCamProfile.eInterpolationMode   := LINEAR;
sCamProfile.r64MasterScale       := 0;
sCamProfile.r64SlaveScale        := 0;
sCamProfile.r64MasterShift       := 0;
sCamProfile.r64SlaveShift        := 0;

// Definition 1st segment
sCamProfile.asCamElement[0].r64InflectionPointParameter           := 0.5;
sCamProfile.asCamElement[0].eCamSegmentType                       := D_C_MOD_SINE;
sCamProfile.asCamElement[0].sSegmentParameter.r64MasterPosStart   := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64MasterPosEnd     := 180.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64SlavePosStart    := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64SlavePosEnd      := 50.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoVeloStart     := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoVeloEnd       := 0.55;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoAccelStart    := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoAccelEnd      := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoJerkStart     := 0.0;
sCamProfile.asCamElement[0].sSegmentParameter.r64GeoJerkEnd       := 0.0;

// Definition 2nd segment
sCamProfile.asCamElement[1].r64InflectionPointParameter           := 0.5;
sCamProfile.asCamElement[1].eCamSegmentType                       := POLY_5;

sCamProfile.asCamElement[1].sSegmentParameter.r64MasterPosStart   := 180.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64MasterPosEnd     := 360.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64SlavePosStart    := 50.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64SlavePosEnd      := 100.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoVeloStart     := 0.55;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoVeloEnd       := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoAccelStart    := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoAccelEnd      := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoJerkStart     := 0.0;
sCamProfile.asCamElement[1].sSegmentParameter.r64GeoJerkEnd       := 0.0;

sCamProfile.i16NumberOfElements := 2;

instFBCreateCam(execute         := TRUE
                ,reset          := FALSE
                ,toCam          := Cam_1
                ,camProfile     := sCamProfile);
```

User manual

Cam disk

You can see the generated cam in the following figure:

Figure 2-5: curve of position, velocity and acceleration



2.2 Function elements and integration

Table 2-1: Elements and integration

Source	fCreaCam	**Programming language**	ST
Library	LCamHdl	**Know-how protection**	Yes

Function / function block program	Property / function	Adaptation to application necessary
FBLCamHdlCreateCam	Callable in all cyclic tasks from SCOUT V4.2 on	**No**
FBLCamHdlCalcCamMinMaxValues	Callable in all cyclic tasks from SCOUT V4.2 on	**No**
FBLCamHdlCalcAxisDynamics	Callable in all cyclic tasks from SCOUT V4.2 on	**No**

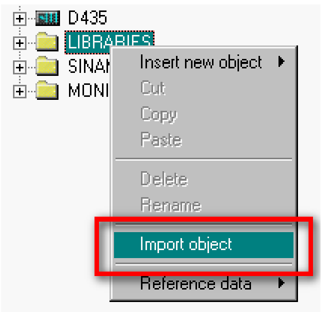
Necessary technology packages / -objects

The CAM technology package is a minimum requirement for the usage of this library as well as a technology object with type cam.

Integration of the library

The LCamHdl library is supplied as an XML export file which can be imported into the SIMOTION project by right clicking the „libraries“ folder, see Figure 2-6.

Figure 2-6: Importing of the library



Please consider that the interpolation of the cam disk within the function block *FBLCamHdlCreateCam* can have a very long runtime, depending on the number of the elements. Therefore, it might become necessary or it is recommended to use the function block in a motion task. However, this motion task has to be run in cycles. Furthermore, the creation of the cam disk can be configured either to be finished within one cycle of the FB or to be distributed to several function block cycles. Consider that the creation of the cam disk within one cycle leads to a higher system load.

2.3 Function block FBLCamHdlCreateCam

2.3.1 Functionality

The curve profile, which has been defined before is transferred to the input *camProfile.* The cam disk object to be used, is made available at the input *toCam.* The output variable *busy* indicates the value TRUE during the calculation of the cam disk. Via *done* = TRUE, the error free operation of the function block is indicated. Status and error messages are indicated at the outputs *error, errorID* and *errorInfo*

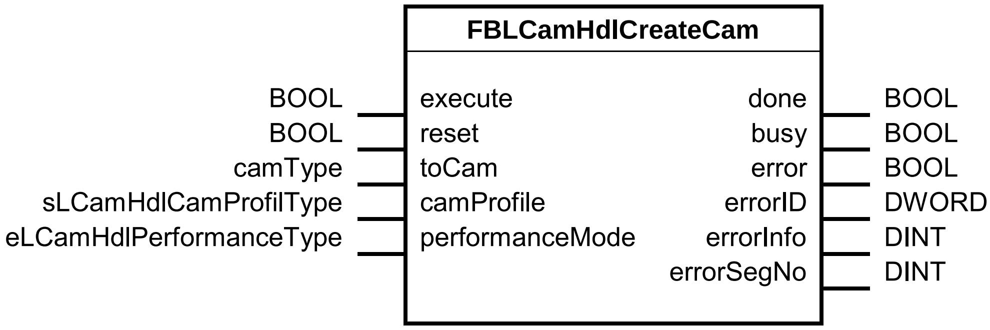
After each segment calculation, the system function *_addSegmentToCam* is called internally. Finally, the cam disk is generated by the system function *_interpolateCam.* Therefore, it is not possible to add further elements at a later date.

Note:

At first, the maximum number of points of the cam disk is limited to 32 in unit *cPublic.* A higher number of points lead to a longer runtime accordingly.

2.3.2 Schematic representation in LAD

Figure 2-7: Schematic representation in LAD



2.3.3 Input and output parameters

Table 2-2: Input and output parameters

Element	P type [1)]	Data type	M/O [2)]	Initial value	Meaning
execute	IN	BOOL	M	FALSE	Positive edge to start the calculation
reset	IN	BOOL	M	FALSE	TRUE: Reset of the function block
toCam	IN	CamType	M	-	Cam disk object to be generated
camProfile	IN	sLCamHdlCam ProfileType	M	-	Definition of the whole curve profile
performance Mode	IN	eLCamHdlPer formanceType	O	BALANCED	Performance mode of the cam disk creation BALANCED: the creation of the cam disk is distributed to several function block cycles HIGH: the creation of the cam disk is finished within one function block cycle
done	OUT	BOOL	-	FALSE	Function successfully completed
busy	OUT	BOOL	-	FALSE	Function active
error	OUT	BOOL	-	FALSE	Function aborted with error
errorID	OUT	DWORD	-	16#0	Error code
errorInfo	OUT	DINT	-	0	Error additional information (error in the internally called system functions - if available)
errorSegNo.	OUT	DINT	-	0	Error additional information (number of segment in which the error occurred - if available)

[1)] Parameter types: IN = input parameters, OUT = output parameters,
IN/OUT = in-out parameters
[2)] Parameter mode: M = mandatory parameter, O = optional parameter

Table 2-3: Elements of structure *sLCamHdlCamProfileType*

Element	Data type	Meaning
i16NumberOfElements	INT	Number of elements / points of the cam disk
eCamModify	EnumCamModify	WITH_INTERPOLATION (default): The currently active cam is deleted and new geometry definitions are written to it. Afterwards the cam is interpolated. WITHOUT_INTERPOLATION: The currently active cam is deleted and new geometry definitions are written to it. The cam is **not** interpolated. The user is responsible of performing the interpolation afterwards. WITHOUT_INTERPOLATION_AND_ HOLD_ACTIVE_CAM: The active cam is retained, however the new geometry definitions are written in parallel in the background. The cam is **not** interpolated. The user is responsible of performing the interpolation afterwards, with _interpolateCam() the cam programmed in the background is activated. → See SIMOTION documentation (*SIMOTION Technology Objects Synchronous Operation, Cam*). The parameter camData of function *_resetCam()* is preset to CAM_DATA_RESET and cannot be changed.
eInterpolationMode	EnumCamInterpolation Mode	Interpolation module for the interpolation of the single points

Element	Data type	Meaning
eCamMode	EnumCamMode	NO_CONSTRAINTS: Not constant at the boundary points CYCLIC_ABSOLUTE: Position-continuous in the boundary points CYCLIC_RELATIVE: Constant velocity in the boundary points For details see SIMOTION documentation
sContinuityCheck	sLCamHdlContinuity CheckType	Check for continuity and correction in the definition and value range (X-axis, Y-axis), see annex.
r64MasterScale	LREAL	Scaling of the master axis (X-axis)
r64SlaveScale	LREAL	Scaling of the slave axis (Y-axis)
r64MasterShift	LREAL	Shifting of the master axis
r64SlaveShift	LREAL	Shifting of the slave axis
boUserDefinedMasterRange	BOOL	FALSE: The master range corresponds to the range which was taken to calculate the cam disk - Default TRUE: The master range corresponds to the range being defined by *r64LeadingRangeStartPoint* and – *EndPoint*
r64LeadingRangeStartPoint	LREAL	Start value for the master range with boUserDefinedMasterRange = TRUE
r64LeadingRangeEndPoint	LREAL	End value for the master range with boUserDefinedMasterRange = TRUE
asCamElement	ARRAY[0..LCAMHDL_MAX_NUMBER_OF_ELEMENTS - 1] OF sLCamHdlCamElement Type	Definition of the single elements/points (positions, velocities, accelerations etc.)

Table 2-4: Elements of structure sLCamHdlContinuityCheckType

Element	Data type	Meaning
r64MinimumDeviation LeadingRange	LREAL	Minimum value of spacing between segments for the definition range (master range, x-axis)
r64MaximumDeviation LeadingRange	LREAL	Maximum value of spacing between segments for the definition range (master range, x-axis)
r64MinimumDeviation FollowingRange	LREAL	Minimum value of spacing between segments for the value range (slave range, y-axis)
r64MaximumDeviation FollowingRange	LREAL	Minimum value of spacing between segments for the definition range (slave range, y-axis)

Table 2-5: Elements of structure sLCamHdlCamElementType

Element	Data type	Meaning
sSegmentParameter	sLCamHdlCam Segment ParameterType	Parameter of the segment
r64InflectionPointPara meter	LREAL	Inflection point parameter (λ) – Default: 0,5 standardized
r64ModVeloTrapezoid Parameter	LREAL	Segment parameter (c) – sine-straight line- combination
r64ModSineMaxAccel CaStar	LREAL	Special case of modified sine – Presetting by Ca* standardized
eCamSegmentType	eLCamHdlCamSe	Profile types for the curve segments

Element	Data type	Meaning
	gmentTypeType	

Table 2-6: Elements of structure sLCamHdlCamSegmentParameterType

Element	Data type	Meaning
r64MasterPosStart	LREAL	Position of master at the start position of the segment
r64MasterPosEnd	LREAL	Position of master at the end position of the segment
r64SlavePosStart	LREAL	Position of slave at the start position of the segment
r64SlavePosEnd	LREAL	Position of slave at the end position of the segment
r64GeoVeloStart	LREAL	Velocity at the start position of the segment (real – standardization is done internally)
r64GeoVeloEnd	LREAL	Velocity at the end position of the segment (real – standardization is done internally)
r64GeoAccelStart	LREAL	Acceleration at the start position of the segment (real – standardization is done internally)
r64GeoAccelEnd	LREAL	Acceleration at the end position of the segment (real – standardization is done internally)
r64GeoJerkStart	LREAL	Jerk at the start position of the segment (real – standardization is done internally)
r64GeoJerkEnd	LREAL	Jerk at the end position of the segment (real – standardization is done internally)

Table 2-7: Elements of the ENUM eLCamHdlCamSegmentTypeType

Element	Meaning
EMPTY	empty
POINT	Transition of a point
DWELL	Special case of dwell
CONST_VELO	Special case of constant velocity (even for v=0 – dwell after dwell)
POLY_3	3rd degree polynomial
POLY_3_OPT_ACCEL_START	3rd degree polynomial acceleration zero at the start (Velocity[Cv] - Dwell)
POLY_3_OPT_ACCEL_END	3rd degree polynomial acceleration zero at the end (Dwell - Velocity[Cv])
POLY_5	5th degree polynomial
POLY_7_JERK_ZERO	7th degree polynomial – limit condition jerk at start and end point = 0
D_D_PARABEL	dwell → dwell - quadratic parabola
D_D_BASIC_SINE	dwell → dwell - basic sine line
D_D_INCLINED_SINE	dwell → dwell - inclined sine line
D_D_MOD_ACCEL_TRAPEZOID	dwell → dwell – modified acceleration trapezoid
D_D_MOD_SINE	dwell → dwell - modified sine line
C_C_MOD_SINE	constant velocity → constant velocity - modified sine line
R_R_MOD_VELO_TRAPEZOID	reversal → reversal - sine-straight line - combination
R_R_BASIC_SINE	reversal → reversal – basic sine line
D_C_MOD_SINE	dwell → constant velocity - modified sine line
C_D_MOD_SINE	constant velocity → dwell - modified sine line
D_R_MOD_ACCEL_TRAPEZOID	dwell → reversal - modified acceleration trapezoid
D_R_HARMONIC_ COMBINATION	dwell → reversal - harmonic combination

Element	Meaning
R_D_MOD_ACCEL_TRAPEZOID	reversal → dwell - modified acceleration trapezoid
R_D_HARMONIC_ COMBINATION	reversal → dwell - harmonic combination
C_R_HARMONIC_ COMBINATION	constant velocity → reversal - harmonic combination
R_C_HARMONIC_ COMBINATION	reversal → constant velocity - harmonic combination

2.3.4 Explanations to the different profile types

The following table shows all profile types. The user has to consider the necessary limit conditions.

Please pay special attention to the subject concerning reversal of velocity (problematical with some machine types) as well as to the meaning of „geometric velocity / acceleration / …“.

Deviations in the limit points

The deviations have to be parameterized with real values, the necessary standardization due to the laws of motion according to VDI 2143 is realized within the function block.

Figure 2-8

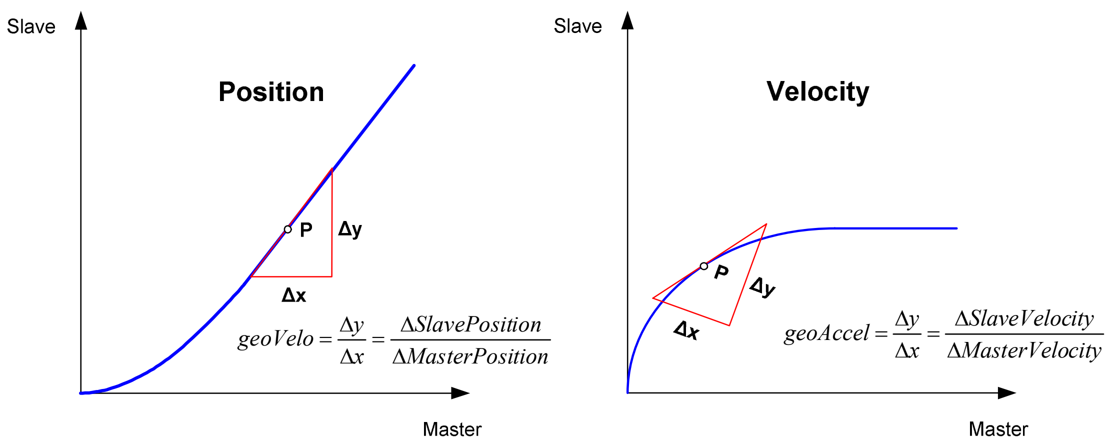


Table 2-8: Explanations of the different profile types

Profile types	General information	Necessary conditions
POINT:	**Application:** Input of a support point	*r64MasterPosStart* *r64SlavePosStart*
DWELL:	**Application:**	*r64MasterPosStart*

Profile types	General information	Necessary conditions
	dwell-transition	*r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd*
CONST_VELO: Straight line	**Application:** *Constant velocity* **Disadvantages:** Motion with impacts, vibrations, noise, wear	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd*
PARABOLA:	**Application:** *dwell – dwell* **Addition:** Shifting of the inflection point (λ) **Advantages:** Lowest Ca value, i.g. low inertial forces **Disadvantages:** Due to acceleration steps, vibrations, noise and wear, stronger feather keys are required for the frictional connection.	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* *r64GeoVeloStart = 0* *r64GeoVeloEnd = 0* Addition: *0 < r64InflectionPointParameter < 1*
BASIC_SINE: basic sine oscillation	**Application:** dwell – dwell reversal - reversal **Addition:** 1. Shifting of the inflection point (λ) 2. Presetting for starting and ending acceleration (λ = 0), for reversal - reversal **Advantages:** low Cv-value low Ca-value **Disadvantages:** acceleration steps, vibrations, noise and wear	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* Addition: *1.* *0 < r64InflectionPointParameter < 1* *2.* *r64InflectionPointParameter = 0* *r64GeoAccelStart <> 0* *r64GeoAccelEnd <> 0*
MOD_VELO_TRAPEZOID: Sine-straight line-combination	**Application:** reversal - reversal	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd*

Profile types	General information	Necessary conditions
	segment parameter (c) – Sine-straight line-combination 0 < c <= 1 **Example:** c= 0,2 20% Sine (split up among start and ending) 80% straight line **Addition:** 1. Shifting of the inflection point (λ) 2. Presetting of start and ending of acceleration (λ = 0) **Advantage:** Especially low Cv-value, i.e. low drive torque due to static charge **Disadvantage:** Ca-value is higher than it is with a simple sine line	Addition: *1.* *0 < r64InflectionPointParameter < 1* *2.* *r64InflectionPointParameter = 0* *r64GeoAccelStart <> 0* *r64GeoAccelEnd <> 0*
POLY_3: 3rd degree polynomial Optional: **POLY_3_OPT_ACCEL_START** **POLY_3_OPT_ACCEL_END**	**Application:** Transition at continuous velocity **Addition:** Shifting of the inflection point (λ) **Advantages:** Can be used for many motion tasks – Presetting of position and velocity **Disadvantages:** Undesired overshoot with unfavorable limit conditions Optional: POLY_3_OPT_ACCEL_START → a_{Start} = 0 $v_{End(standardized)}$ = 1,5 only for λ = 0,5 POLY_3_OPT_ACCEL_END → a_{End} = 0 $v_{Start(standardized)}$ = 1,5	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* *r64GeoVeloStart* *r64GeoVeloEnd* Addition: *0 < r64InflectionPointParameter < 1*

Profile types	General information	Necessary conditions
	only for λ = 0,5	
POLY_5: 5th degree polynomial	**Application:** Constant transition of velocity and acceleration **Addition:** Shifting of the inflection point (λ) **Advantages:** Can be used for nearly all motion tasks – presetting of position, velocity and acceleration **Disadvantages:** Undesired overshoot with unfavorable limit conditions	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* *r64GeoVeloStart* *r64GeoVeloEnd* *r64GeoAccelStart* *r64GeoAccelEnd* Addition: *0 < r64InflectionPointParameter < 1*
POLY_7_JERK_ ZERO: 7th degree polynomial	**Application:** Constant transition of velocity, acceleration and jerks (the jerk is always = 0 in the limit point) **Addition:** Shifting of the inflection point (λ) **Advantages:** Can be used for nearly all motion tasks – presetting of position, velocity, acceleration and fixed jerk at the limit points of zero **Disadvantages:** Undesired overshoot with unfavorable limit conditions The 7th degree polynomial is indicated internally by two 5th degree polynomials	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* *r64GeoVeloStart* *r64GeoVeloEnd* *r64GeoAccelStart* *r64GeoAccelEnd* Addition: *0 < r64InflectionPointParameter < 1*
INCLINED_SINE: inclined sine	**Application:** dwell – dwell **Addition:** Shifting of the inflection point (λ)	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd*

Profile types	General information	Necessary conditions
	Advantage: Especially low Cj-value, low vibration, well adapted for high speed **Disadvantage:** Cv, Ca, Cmdyn value is higher than POLY_5	*r64GeoVeloStart = 0* *r64GeoVeloEnd = 0* Addition: *0 < r64InflectionPointParameter < 1*
MOD_ACCEL_ TRAPEZOID: modified acceleration trapezoid	**Application:** dwell - dwell dwell – reversal reversal – dwell **Addition:** dwell - dwell Shifting of the inflection point (λ) dwell - reversal / reversal - dwell **Addition:** 1. Shifting of the inflection point (λ) 2. Presetting of acceleration start/end (λ = 0) **Advantage:** Especially low Ca-value, i.e. low drive torque **Disadvantage:** Cv, Cj and Cmdyn values are higher than those with the harmonic combination	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* *r64GeoVeloStart = 0* *r64GeoVeloEnd = 0* Addition: dwell - dwell *0 < r64InflectionPointParameter < 1* dwell – reversal *1.* *0 < r64InflectionPointParameter < 1* *2.* *r64InflectionPointParameter = 0* *r64GeoAccelEnd <> 0* reversal – dwell *1.* *0 < r64InflectionPointParameter < 1* *2.* *r64InflectionPointParameter = 0* *r64GeoAccelStart <> 0*
MOD_SINE: Sinusoidal acceleration	**Application:** dwell especially low Cv-value, i.e. low drive torque due to static charge **Disadvantage:** Ca-value is higher than it is with a simple sine line – dwell constant velocity - constant velocity dwell – constant velocity constant velocity – dwell	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* *r64GeoVeloStart* *r64GeoVeloEnd* Example: dwell – dwell *r64GeoVeloStart = 0* *r64GeoVeloEnd = 0*

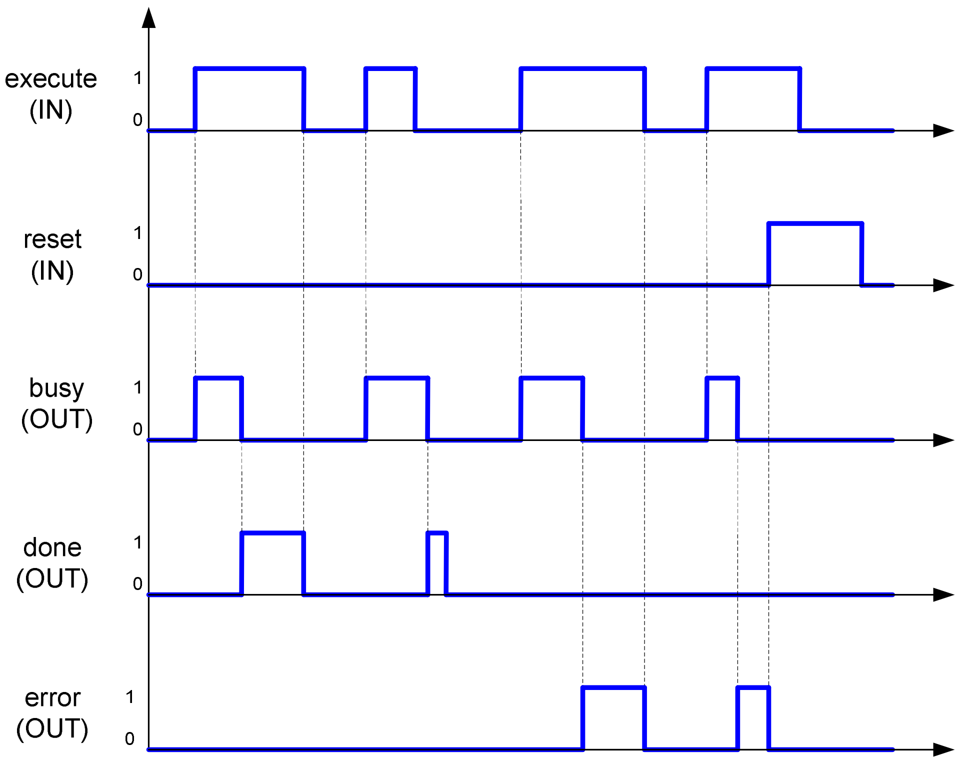
Profile types	General information	Necessary conditions
	Addition: 1. Shifting of the inflection point (λ) 2. Presetting of Ca* (λ = 0) – except dwell-dwell – Ca* standardized size (see annex) **Advantage:** Well adapted for high speed, low Cv, Ca und Cmdyn Wert **Disadvantage:** Cj-value is higher than inclined sine line	Addition: *1.* *0 < r64InflectionPointParameter < 1* (except dwell-dwell) *2.* *r64InflectionPointParameter = 0* *r64ModSineMaxAccelCaStar <> 0*
HARMONIC_ COMBINATION: Harmonic combination consists of harmonic parts only	**Application:** constant velocity – reversal reversal – constant velocity **Addition:** Shifting of the inflection point (λ) **Advantage:** Especially low Cv and Cmdyn values, favorable to high inertia mass at the drive **Disadvantage:** Ca-value is higher than modified acceleration trapezoid	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* constant velocity – reversal *r64GeoVeloStart <> 0* *r64GeoVeloEnd = 0* reversal - constant velocity *r64GeoVeloStart = 0* *r64GeoVeloEnd <> 0* Addition: *0 < r64InflectionPointParameter < 1*
	Application: dwell - reversal reversal - dwell **Addition:** 1. Shifting of the inflection point (λ) 2. Presetting of acceleration Start/end (λ = 0) **Advantage:** especially low Cv and Cmdyn values, favorable to high inertia mass at the drive **Disadvantage:**	*r64MasterPosStart* *r64SlavePosStart* *r64MasterPosEnd* *r64SlavePosEnd* Addition: dwell – reversal *1.* *0 < r64InflectionPointParameter < 1* *2.* *r64InflectionPointParameter = 0* *r64GeoAccelEnd <> 0* reversal – dwell *1.*

Profile types	General information	Necessary conditions
	Ca-value higher than modified acceleration trapezoid	*0 < r64InflectionPointParameter < 1* *2.* *r64InflectionPointParameter = 0* *r64GeoAccelStart <> 0*

2.3.5 Time sequence chart

The time sequence chart for the output variable *busy* is shown in the figure below. During the calculation of the cam disk, the value of the variable *busy* is TRUE.

Figure 2-9: time sequence chart



2.3.6 Error numbers

In the following table you can find all status / error values of the output variable *errorID*.

Table 2-9: Status / error indication *errorID*

errorID [HEX]	Meaning
0	Error free
4000	Internal error, invalid state
4001	Wrong transition values in segment no., see *errorSegNo*
4002	0<= r64ModVeloTrapezoidParameter >1 in segment no., see *errorSegNo*
4003	Invalid value for *r64ModSineMaxAccelCaStar* in segment no., see *errorSegNo*
4004	Invalid acceleration preset *r64GeoAccelStart*/*End* in segment no., see *errorSegNo*
4005	Invalid parameter for shifting the inflection point (0< λ<1) *r64InflectionPointParameter* in segment no., see *errorSegNo*
4006	Due to the parameterization, the calculated shifting of the inflection point (λ) is outside the range (0< λ<1) in segment no., see *errorSegNo*
4007	The range of motion at the master is <= 0 in segment no., see *errorSegNo*
4008	The range of motion at the slave is 0 in segment no., see *errorSegNo*
4009	No reference has been transferred to a cam disk
400A	No segments have been transferred to calculate a cam disk
400B	System command _resetCam was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
400C	System command _interpolateCam was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
400D	System command _setCamScale for the master axis was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
400E	System command _setCamScale for the slave axis was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
400F	System command _setCamOffset for the master axis was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
4010	System command _setCamOffset for the slave axis was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
4015	Invalid parameterization of HARMONIC_COMBINATION (const. velocity – reversal / reversal – const. velocity), characteristic value Ca becomes invalid.
4016	Positive edge at input *execute* detected while input *reset* is active.

errorID [HEX]	Meaning
4017	System command _setAddPointToCam was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
4018	System command _ addSegmentToCam was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*
4019	System command _ addPolynomialSegmentToCam was interrupted by an error. The feedback value of the system command can be found in *errorInfo.*

2.4 Function block FBLCamHdlCalcCamMinMaxValues

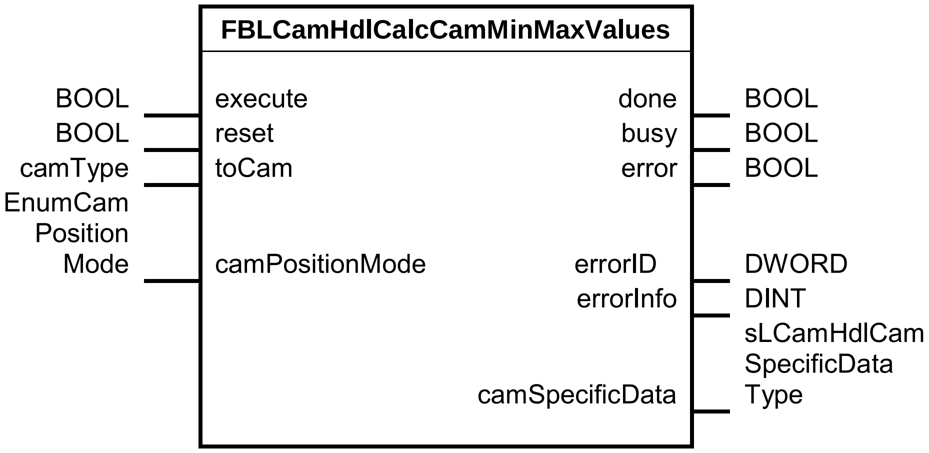
2.4.1 Functionality

The function block **FBLCamHdlCalcCamMinMaxValues** calculates the occurring minimum and maximum values for position, velocity, acceleration and jerk at master and slave for the transferred cam disk. The cam disk to be used is made available at the input *toCam.* During the calculation of the cam disk, the output variable *busy* indicates the value TRUE. The error free operation of the function block is indicated by *done* = TRUE. Status and error messages are output at the outputs *error, errorID* and *errorInfo.*

The system function *_getCamFollowingMinMax* is called internally.

2.4.2 Schematic representation in LAD

Figure 2-10: Schematic representation in LAD



2.4.3 Input and output parameters

Table 2-10: Input and output parameters

Element	P type [1)]	Data type	M/O [2)]	Initial value	Meaning
execute	IN	BOOL	M	FALSE	Positive edge to start the calculation
reset	IN	BOOL	M	FALSE	TRUE: Reset of the function block
toCam	IN	CamType	M	-	Cam disk object to be examined
camPosition Mode	IN	EnumCam PositionMode	O	ACTUAL	ACTUAL – with scale and offset BASIC – without scale and offset
done	OUT	BOOL	-	FALSE	Function successfully completed
busy	OUT	BOOL	-	FALSE	Function active
error	OUT	BOOL	-	FALSE	Function aborted with error
errorID	OUT	DWORD	-	16#0000	Error code
errorInfo	OUT	DINT	-	0	Error additional information / error in the internally called system functions
camSpecificData	OUT	sLCamHdlCamSpecificDataType	-	-	Calculated minimum and maximum values of the transferred cam disk

[1)] Parameter types: IN = input parameters, OUT = output parameters, IN/OUT = in-out parameters
[2)] Parameter mode: M = mandatory parameter, O = optional parameter

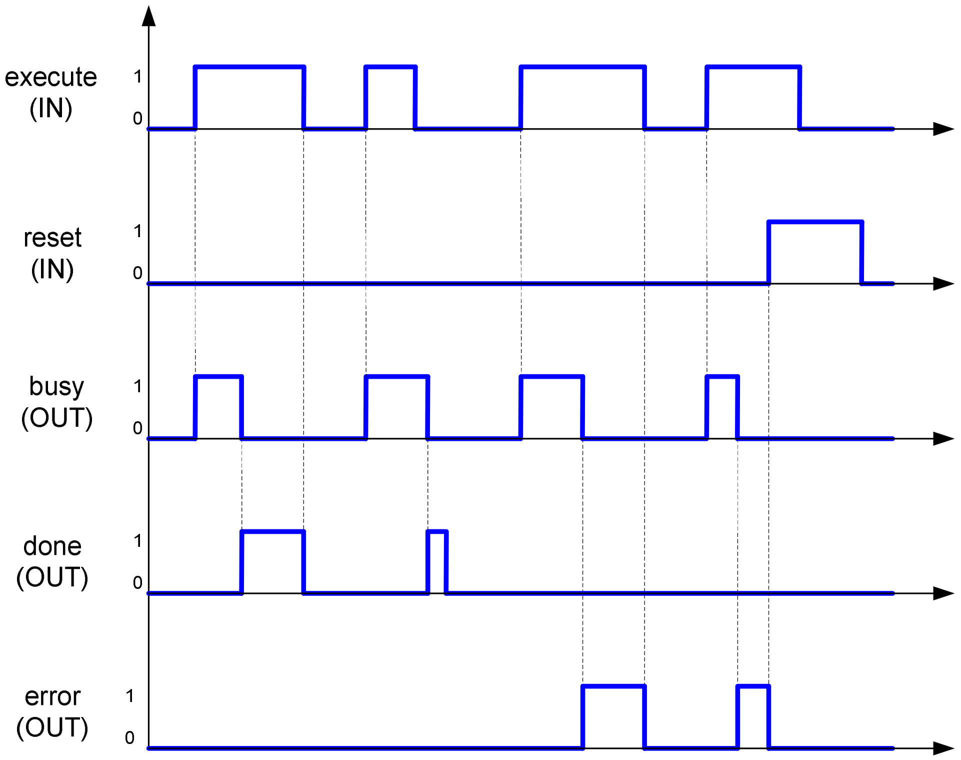
Table 2-11: Elements of structure sLCamHdlCamSpecificDataType

Element	Data type	Meaning
r64MasterPosMax	LREAL	Maximum position of the master within the cam disk
r64MasterPosMin	LREAL	Minimum position of the master within the cam disk
r64MasterCvMax	LREAL	Maximum velocity of the master within the cam disk
r64MasterCvMin	LREAL	Minimum velocity of the master within the cam disk
r64MasterCaMax	LREAL	Maximum acceleration of the master within the cam disk
r64MasterCaMin	LREAL	Minimum acceleration of the master within the cam disk
r64MasterCjMax	LREAL	Maximum jerk of the master within the cam disk
r64MasterCjMin	LREAL	Minimum jerk of the master within the cam disk
r64SlavePosMax	LREAL	Maximum position of the slave within the cam disk
r64SlavePosMin	LREAL	Minimum position of the slave within the cam disk
r64SlaveCvMax	LREAL	Maximum velocity of the slave within the cam disk
r64SlaveCvMin	LREAL	Minimum velocity of the slave within the cam disk
r64SlaveCaMax	LREAL	Maximum acceleration of the slave within the cam disk
r64SlaveCaMin	LREAL	Minimum acceleration of the slave within the cam disk
r64SlaveCjMax	LREAL	Maximum jerk of the slave within the cam disk
r64SlaveCjMin	LREAL	Minimum jerk of the slave within the cam disk

2.4.4 Time sequence chart

The time sequence chart for the output variable *busy* is shown in the figure below. During the calculation of the cam disk, the value of the variable *busy* is TRUE.

Figure 2-11: time sequence chart



2.4.5 Error numbers

In the following table you can find all status / error values of the output variable *errorID*.

Table 2-12: Status / error indication *errorID*

errorID [HEX]	Meaning
0	Error free
4000	Internal error, invalid state
4009	No reference has been transferred to a cam disk
4011	An error has occurred during the calculation of the maximum values for the position. (for error value ov _getCamFollowingMinMax, please see *errorInfo*)
4012	An error has occurred during the calculation of the maximum values for the velocity. (For error value of _getCamFollowingMinMax , please see *errorInfo*)
4013	An error has occurred during the calculation of the maximum values for the acceleration. (For error value of _getCamFollowingMinMax , please see *errorInfo*)
4014	An error has occurred during the calculation of the maximum values for the jerk. (For error value of _getCamFollowingMinMax , please see *errorInfo*)
4016	Positive edge at input *execute* detected while input *reset* is active

2.5 Function block FBLCamHdlCalcAxisDynamics

2.5.1 Functionality

The function block **FBLCamHdlCalcAxisDynamics** calculates the maximum occurring dynamics (velocity, acceleration and jerk) on the following axis for the special case having a constant master velocity – potential acceleration or deceleration of the master axis is not taken into account. Any synchronization dynamics are also not considered.

Furthermore the maximum possible master velocity can be calculated with respect to the maximum dynamics (velocity, acceleration and jerk) of the following axis. Again, prerequisite for this is that the cam disk is used at a constant master velocity and not taking into account any synchronization dynamics during synchronous operation. In case a jerk value of ≤ 0 is set (*slaveAxisDynamics.r64MaxJerk*), this value is ignored for the calculation of the master velocity.

The cam disk data, e.g. calculated with function block *FBLCamHdlCalcCamMinMaxValues*, are handed over via input *camSpecificData*.

During calculation the *busy* output has the value TRUE. The error free completion of the function block is indicated with *done* = TRUE. Status and error messages are displayed at the outputs *error, errorID* and *errorInfo.*

calcMode = CALC_MAX_SLAVE_DYNAMICS or CALC_ALL

For the calculation of the resulting dynamics of the following axis, the constant master velocity must be assigned at the input *masterVelocity*. The calculation is based on the equations in Table 1-1.

The maximum calculated dynamics (velocity, acceleration and jerk) of the following axis are available at output structure *calcMaxSlaveAxisDynamics* after error free completion.

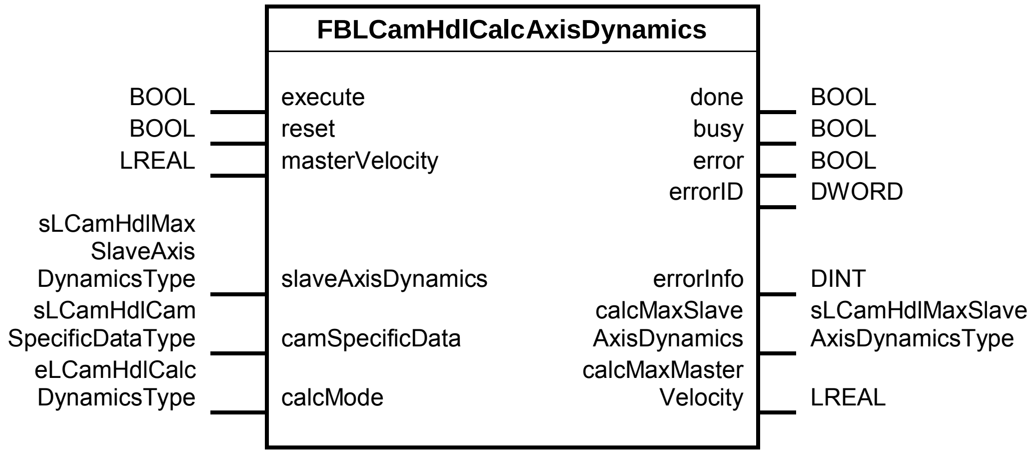
calcMode = CALC_MAX_MASTER_VELOCITY or CALC_ALL

The maximum possible master velocity is calculated with respect to the maximum dynamics (velocity, acceleration and optional jerk) of the following axis, see input *slaveAxisDynamics*. The calculation is based on the equations in Table 1-1.

The maximum calculated master velocity is available at output *calcMaxMasterVelocity* after error free completion.

2.5.2 Schematic representation in LAD

Figure 2-12: Schematic representation in LAD



2.5.3 Input and output parameters

Table 2-13: Input and output parameters

Element	P type [1)]	Data type	M/O [2)]	Initial value	Meaning
execute	IN	BOOL	M	FALSE	Positive edge to start the calculation
reset	IN	BOOL	M	FALSE	TRUE: Reset of the function block
master Velocity	IN	LREAL	O	0.0	Constant master velocity for the calculation of the resulting dynamics of the following axis (*calcMode = CALC_MAX_SLAVE_DYNAMICS*)
slaveAxis Dynamics	IN	sLCamHdlMaxSlave AxisDynamicsType	O	-	Maximum dynamics of the following axis for the calculation of the maximum possible master velocity (*calcMode = CALC_MAX_MASTER_VELOCITY*)
camSpecific Data	IN	sLCamHdlCam SpecificDataType	O	-	Structure with the minimum and maximum dynamics values of the cam disk – see FBLCamHdlCalcCamMinMaxValues
calcMode	IN	eLCamHdlCalc DynamicsType	O	CALC_ALL	Calculation modes (see chapter 2.5.1) CALC_ALL CALC_MAX_SLAVE_DYNAMICS CALC_MAX_MASTER_VELOCITY
done	OUT	BOOL	-	FALSE	Function successfully completed
busy	OUT	BOOL	-	FALSE	Function active
error	OUT	BOOL	-	FALSE	Function aborted with error
errorID	OUT	DWORD	-	16#0000	Error code
errorInfo	OUT	DINT	-	0	Additional error information / error in the internally called system functions
calcMaxSlave AxisDynamics	OUT	sLCamHdlMaxSlave AxisDynamicsType	-	-	Calculated dynamics of the following axis (*calcMode = CALC_MAX_SLAVE_DYNAMICS*)
calcMax MasterVelocity	OUT	LREAL	-	0.0	Calculated maximum master velocity (*calcMode = CALC_MAX_MASTER_VELOCITY*)

[1)] Parameter types: IN = input parameters, OUT = output parameters,
IN/OUT = in-out parameters
[2)] Parameter mode: M = mandatory parameter, O = optional parameter

Table 2-14: Elements of structure *sLCamHdlMaxSlaveAxisDynamicsType*

Element	Data type	Meaning
r64MaxVelocity	LREAL	Maximum velocity
r64MaxAcceleration	LREAL	Maximum acceleration
r64MaxJerk	LREAL	Maximum jerk

Table 2-15: Elements of enum *eLCamHdlCalcDynamicsType*

Element	Meaning
CALC_ALL	Calculate all
CALC_MAX_SLAVE_DYNAMICS	Calculate the maximum occurring dynamics (velocity, acceleration and jerk) on the following axis using the cam disk specific data and a constant master velocity

Element	Meaning
CALC_MAX_MASTER_VELOCITY	Calculate the maximum possible master velocity using the cam disk specific data and the maximum dynamics (velocity, acceleration and optional jerk) of the following axis

2.5.4 Time sequence chart

The time sequence chart for the output variable *busy* behaves as shown in Figure 2-11.

2.5.5 Error numbers

The following table shows all status / error values of the output variable *errorID*.

Table 2-16: Status / error indication *errorID*

errorID [HEX]	Meaning
0	Error free
4000	Internal error, invalid state
4016	Positive edge at input *execute* detected while input *reset* is active
4050	Invalid cam disk data for CvMax / CvMin (camSpecificData.r64SlaveCvMax = 0 and camSpecificData.r64SlaveCvMin = 0)

Annex

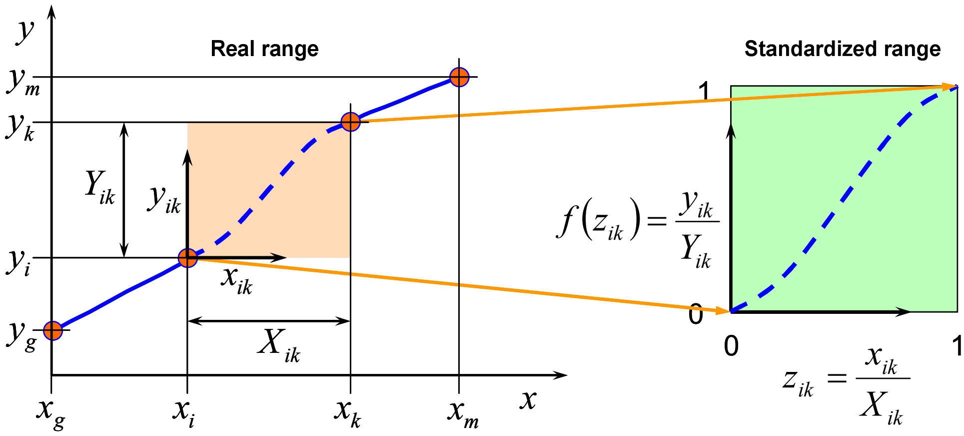
3 Definition of standardized laws of motion

To define motion transitions, a standardized notation without dimensions is used. To do so, the master range x_{ik} is referred to the whole range X_{ik} and the slave range y_{ik} is referred to the whole range Y_{ik}.in the according motion segments.

Master range $$z_{ik} = \frac{x_{ik}}{X_{ik}}$$

Slave range $$f(z_{ik}) = \frac{y_{ik}}{Y_{ik}}$$

Figure 3-1: relation between real and standardized law of motion



Hence, for the standardized transition functions and their deviation follows:

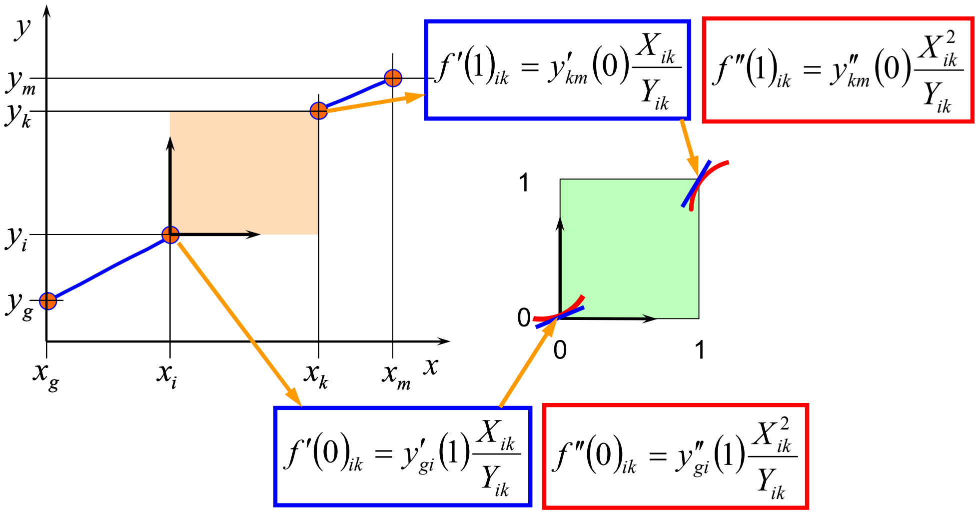
Table 3-1: Standardized transition function

Standardization	Transition Function
Position	$f_{ik} = f(z_{ik}) \rightarrow y_{ik} \cdot \frac{1}{Y_{ik}}$
Velocity	$f'_{ik} = \frac{df_{ik}}{dz_{ik}} \rightarrow y'_{ik} \cdot \frac{X_{ik}}{Y_{ik}}$

Standardization	Transition Function
Acceleration	$f''_{ik} = \dfrac{d^2 f_{ik}}{dz^2{}_{ik}} \rightarrow y''_{ik} \cdot \dfrac{X^2{}_{ik}}{Y_{ik}}$
Jerk	$f''_{ik} = \dfrac{d^3 f_{ik}}{dz^3{}_{ik}} \rightarrow y'''_{ik} \cdot \dfrac{X^3{}_{ik}}{Y_{ik}}$

Figure 3-2 shows the edge points of the first and second degree deviations of the standardized law of motion.

Figure 3-2: Edge points of the standardized transition function



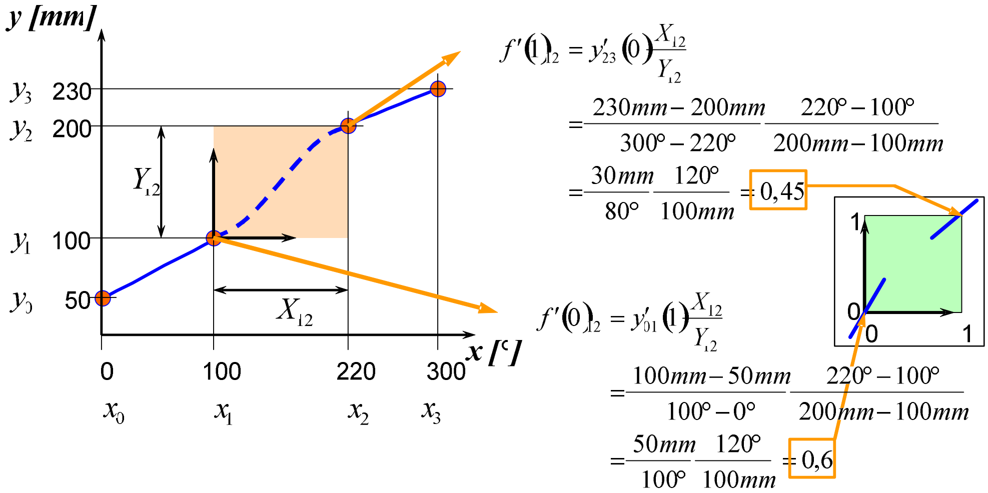
By means of the following relations, you can get the real output motion and its deviation according to the master component X

Table 3-2: Real output motion based upon the standardized laws of motion

Real	Transition function
Position	$y_{ik} = f_{ik} \cdot Y_{ik}$
Velocity	$y'_{ik} = f'_{ik} \cdot \dfrac{Y_{ik}}{X_{ik}}$
Acceleration	$y''_{ik} = f''_{ik} \cdot \dfrac{Y_{ik}}{X^2{}_{ik}}$
Jerk	$y'''_{ik} = f'''_{ik} \cdot \dfrac{Y_{ik}}{X^3{}_{ik}}$

Figure 3-3 shows a numerical example for the standardized transition function of the first deviation (velocity).

Figure 3-3: numerical example of edge points for the transition function



ATTENTION	The examples of the standardization are only meant to make it easy to understand the internal steps of calculation of the module. The user must not execute this standardization during the parameterization, but has to determine the deviations in the edge points in the real range, please refer to example 2.

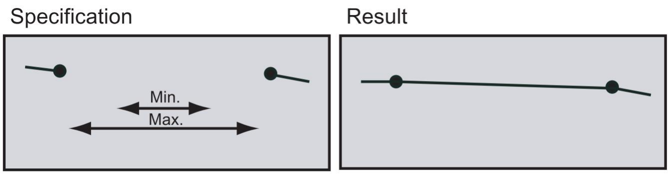
4 Continuity check and correction

A function with assigned parameters can be checked for continuity in the definition range and value range, and possible points of discontinuity can be corrected. During this process, the points of discontinuity are examined separately for the definition range and value range, and are rated for one of the following corrective actions:

- If the absolute value of the spacing between segments exceeds a maximum value, a correction is made by performing an interpolation between the two segments. This results in insertion of a new segment.

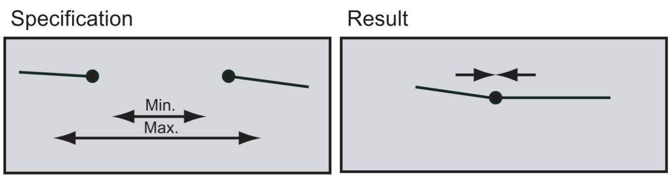
  Is this segment mathematically undefined, e.g. two different positions for the value range but one identical position for the definition range, error „hex 400C“ is thrown.

Figure 4-1: Interpolation by insertion a new segment



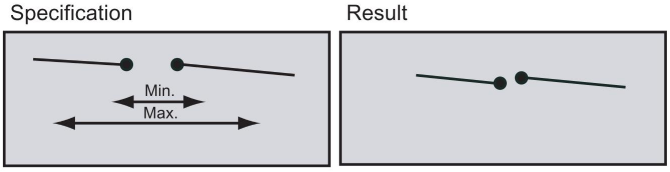
- If the absolute value of the spacing between segments is greater than the minimum value and less than the maximum value, correction is made by joining the segment end points. The mean value of the spacing of the function is used for the correction. The shape of the segments is affected as a result.

Figure 4-2: Correction by joining segment end points



- If the absolute value of the spacing between segments or interpolation points is less than the minimum value, a correction is not made. The discontinuity point is retained. When this discontinuity point is accessed, the right boundary point is output.

Figure 4-3: Allowing the discontinuity to remain unchanged



- Minimum values:

  sCamProfileInternal.sContinuityCheck.r64MinimumDeviationLeadingRange

  sCamProfileInternal.sContinuityCheck.r64MinimumDeviationFollowingRange

- Maximum values:

  sCamProfileInternal.sContinuityCheck.r64MaximumDeviationLeadingRange

  sCamProfileInternal.sContinuityCheck.r64MaximumDeviationFollowingRange

Depending on the combined evaluation in the definition range and value range, the point of discontinuity is corrected according to the following scheme:

Table 4-1: Combined evaluation for definition range and value range of point of discontinuity

		Correction for definition range (X-axis)		
		Discontinuity retained	Join segment end points	Interpolation
Correction for value range (Y-axis)	Retain discontinuity	Retain discontinuity	Join segment end points	New segment
	Join segment end points	Join segment end points	Join segment end points	New segment
	Interpolation	Retain discontinuity	Join segment end points	New segment

- Function continuity can be achieved with linear interpolation.
- With spline interpolation, continuity is possible in the derivatives.

If the continuity condition cannot be adhered to because of the selected interpolation method or the programmed geometry, a message is provided to that effect.

5 Literature index

Table 5-1: Literature

No.	Literature
1.	Siemens Industry Online Support https://support.industry.siemens.com
2.	Download page of this entry https://support.industry.siemens.com/cs/ww/en/view/105644659
3.	SIMOTION ProjectGenerator easyProject https://support.industry.siemens.com/cs/ww/en/view/51339107
4.	VDI 2143, Bl. 1: Laws of motion for cam disks - theoretical basics

6 Alterations / Author

Table 6-1: Alterations / Author

Version	Date / Author	Alteration
V0.9	2011-11-05 / FB	First generation
V1.0	2012-02-13 / AW	Extensions
V1.0.1	2013-02-22 / AW	eCamMode added (_interpolateCam)
V1.1.0	2013-05-24 / AW	Continuity check and correction
V1.1.1	2014-01-21 / AW	FBLCamHdlCreateCam – error code added FBLCamHdlCalcCamMinMaxValues – camPositionMode added
V1.2.0	2014-12-03 / AW	FBLCamHdlCalcAxisDynamics added eCamModify added No support of SIMOTION V4.1 anymore
V1.3.0	2015-01-30 / AW	FBLCamHdlCreateCam - error codes added, errorSegNo added, behavior of errorInfo changed (see chapter 2.3.6 Error numbers), input performanceMode added
V1.3.1	2015-09-14 / AW	POLY_7_JERK_ZERO – parameter description revised
V1.3.2	2016-12-06 / FB	FBLCamHdlCalcAxisDynamics – additional information at meaning of error numbers 4050 and 4051 added
V1.3.3	2017-01-26 / AW	FBLCamHdlCalcAxisDynamics – plausibility CaMin / CaMax removed, now CaMin = 0 and CaMax = 0 allowed, errorID 4051 removed

7 Contacts

Application center

SIEMENS

Siemens AG
Digital Factory
Factory Automation
Production Machines
DF FA PMA APC
Frauenauracher Str. 80
91056 Erlangen, Germany
Fax: +49-9131-98-1297
mailto: tech.team.motioncontrol@siemens.com