MCS2 Trajectory Streaming

Abstract
The MCS2 is a multi channel controller. Each channel is typically commanded independently from other channels when using point-to-point movements. The **trajectory streaming** offers a possibility to move several channels synchronously along a defined trajectory. This user guide explains the general concept of the trajectory streaming and gives implementation hints for the usage.

INTRODUCTION

When trying to move several channels synchronously communication delays induce a time offset of the resulting movements. For most applications this usually does not represent a limitation.
If higher requirements regarding synchronization exist, command groups may be used to define an atomic group of commands that is executed synchronously. This allows to reduce communication delays, but consecutive targets still must be commanded by a controlling application. For this, the application must instruct all channels to move to a new target and then must wait for feedback from all channels (in the form of channel state polling or listening to events) before commanding the next targets. This usually requires a maintaining state machine or sequencer in the control application.
The **trajectory streaming** follows a different approach. The trajectory for all channels is pre-calculated by the application and is then streamed to the controller which takes care of executing a synchronized movement. Obtaining regular feedback about the current state of each channel is not needed due to the fact, that the application calculates the trajectory and not the controller. I.e., there is no need to know when a movement has finished to start the next movement. Several trigger modes allow to synchronize the stream to external devices.

GENERAL STREAMING CONCEPT

A trajectory stream is a defined sequence of support points (frames). Each frame contains all target positions for all channels that participate in the trajectory stream.
Note that the trajectory stream is not a command buffer which holds a sequence of point-to-point movements. Instead, the application must calculate the target positions of all channels with a fixed timing interval. The timing with which the frames are executed is defined by the stream rate that is configurable by the user. This rate is constant for the duration of the stream. Optionally, the timing can be synchronized or even fully controlled by using an external trigger signal.
When using the *trajectory streaming* the controlling application takes over the control about the velocity and acceleration of the movements. If a positioner is supposed to perform an accelerated movement along the trajectory then the support points must be calculated accordingly. Which channels participate in the trajectory stream must be defined when starting the stream. Participating channels which should not move at a specific time while streaming must still be supplied with target positions. (Keeping the streamed absolute position constant results in holding the position.)
Note that stream targets are always absolute positions. To prevent sudden jumps or unexpected behavior of the positioners, the calculated stream targets must either start at the current positions of the channels or all channels must be moved to the position of the first stream target via point-to-point movement commands prior to starting the stream.
Other channels which are not streamed may be commanded via regular point-to-point commands while the stream is executed. When commanding *point-to-point movements*, in contrast to the trajectory streaming, the user simply defines velocity and acceleration by setting the corresponding properties and instructs the channel to move to a specific target position. The controller then internally generates the desired trajectory to move the positioner based on these settings. The application must request feedback from the channel to determine the end of a movement and to know when to command the next movement.
Figure 1 shows the command structure for three channels. Two channels are moved by a trajectory stream while a third channel receives point-to-point movement commands.

Stream Buffer

When the host transmits stream frames to the controller they are stored in a (FIFO) stream buffer in the controller. While sending the stream frames a flow control mechanism of the library prevents buffer overflows. The rate at which the controller receives frames from the host may vary. Typically the rate is considerably higher or frames arrive in bursts with intermis-

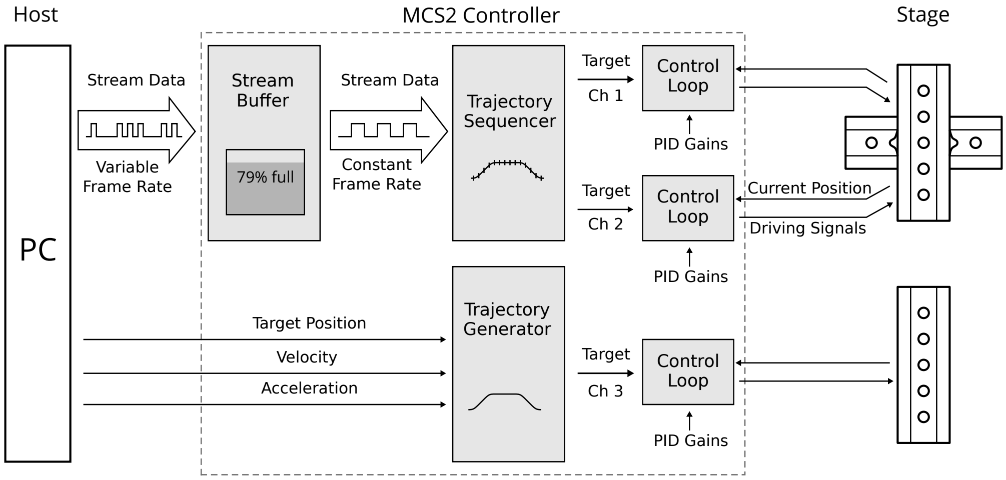

Figure 1. *Trajectory Streaming and Point-to-Point Movement Command Structure*

sions (or both), e.g. due to USB / ethernet latency or application interruption by the operating system. Note that while the stream buffer allows the device to generate a constant stream rate, it also induces a delay to the incoming frames. This delay depends on the controller's buffer size, the number of channels that participate in the stream as well as the configured stream rate. Once the stream buffer is filled the controller starts to play back the trajectory with the configured stream rate.

Control Loop

The stream then defines the continuously moving targets for all channels and the control-loop of each channel tries to follow this target as best as it can. For this, the tuning of the control-loop has a major influence on how well the physical position of a channel follows the defined trajectory. Note that there is no guarantee that a channel follows the trajectory *exactly*. A certain amount of following error must always be taken into account. This lays in the nature of a control-loop and cannot be avoided completely. But the tuning of the control-loop may be adjusted to reduce the following error to its minimum. Furthermore, a lower velocity causes a lower following error. Reducing the velocity, if applicable, also allows to reduce the following error. If the resulting velocity or acceleration of the streamed trajectory exceeds the physical limits of the positioners the controller does not recognizes this directly, it simply continues to play back the stream targets. If a positioner is not able to follow the trajectory, the endstop detection, the following error detection (if enabled and configured) or the overload detection will quite likely trigger sooner or later and subsequently will abort the stream with a corresponding error.
Note that open-loop movements are not supported by the trajectory streaming.
Please refer to the "Trajectory Streaming" chapter of the *MCS2 Programmer's Guide* for a full list of preconditions and general streaming rules. Furthermore, the supplied programming examples show the usage of the streaming API functions and explain how the stream data must be formatted to be sent to the controller.

Stream Interpolation

The controller performs a linear interpolation between the streamed target positions to guarantee a smooth movement. In some applications this behavior might be unwanted. E.g. for a scanning application with movements on a sub micrometer scale a staircase-like motion may be required.
For this, each target should be reached as fast as possible and should then be held e.g. to perform a measurement, before continuing to move to the next target. For this kind of application, the interpolation can be disabled with the stream options property.

EXTERNAL TRIGGERS

Input Triggers may be used to synchronize the stream clock to external devices. For this, different trigger modes are available. The stream may be started with an external trigger signal or the internal stream clock may be synchronized to an external clock signal. Figure 2 shows the relation of the trajectory stream rate resp. trajectory update and the external trigger signal depending on the different trigger modes.
The "external once" trigger mode allows to start the synchronized movement with an external trigger signal. The "external sync" trigger mode is used to synchronize the internal stream rate with an external clock signal to prevent the clocks from drifting apart. For this, the configured stream rate must be a whole-number multiple of the external clock rate. With the "external" trigger mode the external clock defines the stream rate and fully controls the trajectory's start and further updates of the trajectory. With each incoming trigger, the next support point of the given trajectory is targeted.
Output trigger on the other side allow to generate e.g. trigger signals based on the *current* position of

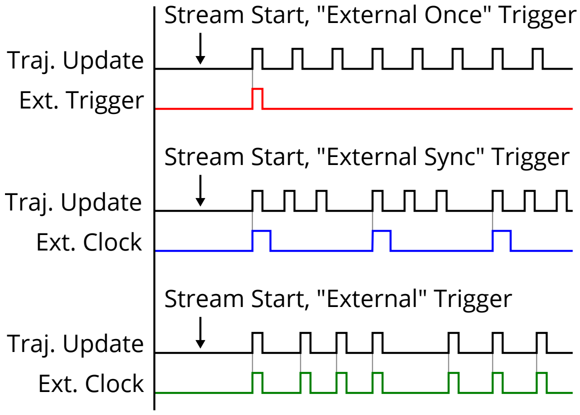

Figure 2. *External Trigger Modes*

each channel. As described before, the generated stream trajectory always defines the *desired* positions. Since a control-loop is involved, the current position may deviate from this position. The output trigger component allows to generate trigger signals based on the *real* physical position of a channel with almost no latency and therefore may be used to trigger external devices like detectors or cameras. Please refer to the ”Input Trigger” and ”Output Trigger” chapters of the *MCS2 Programmer's Guide* for more information.
Note that the device must be equipped with an additional I/O module in order to use external trigger signals.

CALCULATING STREAM TARGETS

As pointed out before, the application on the host PC must pre-calculate the trajectory for all participating channels. The calculations may be done before starting the stream or on-the-fly.
Depending on the application, different approaches may be followed to calculate the stream targets. The easiest method would be to simply increment the target position between two frames by a specific value. In combination with the stream rate this results in a movement of the positioner with a specific and constant velocity. For channels which should hold the current position the target position from the last frame is simply used again for the next frame.
However, when calculating the trajectory the physical limits of all positioners regarding maximum possible acceleration and velocity should be observed to avoid vibrations or jumps and to generate a smooth movement. Therefore, acceleration and deceleration phases should be added to the movements. To achieve this the very common formulas of accelerated movements may be used. To calculate the stream target s at any point of time from a given velocity v and acceleration a, the following basic formula must be used:

$$s(t) = \frac{a * t^2}{2} + v_0 * t \qquad (1)$$

First, a movement must be split into different sections, like:

1. accelerating (s_1),
2. moving with a constant velocity (s_2),
3. decelerating (s_3) and
4. holding the current position (s_4).

For each section the desired stream targets must then be calculated using formula 1. By passing the incrementing time parameter along with the max. velocity and acceleration to the formula in a loop all targets may be generated and appended to a data buffer which is then used for the streaming.

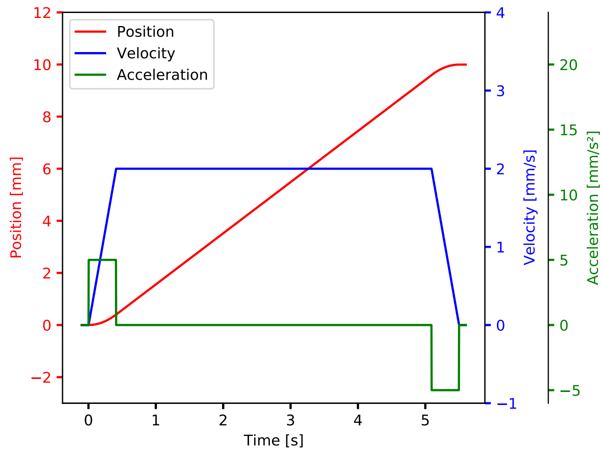

Figure 3. *Movement with Acceleration, constant Velocity and Deceleration*

Figure 3 shows the position, velocity and acceleration over time of an example movement from position zero to s_{end} = 10 mm with a maximum velocity of v = 2 mm/s and a maximum acceleration of a = 5 mm/s^2. For this movement the calculation of the stream targets is shown in the following.
Assuming a stream rate of 1000 Hz should be used, a target position must be calculated for each t_{diff}:

$$t_{diff} = \frac{1}{\text{streamrate}} = \frac{1}{1000\,\text{Hz}} = 1\,\text{ms} \qquad (2)$$

Acceleration Phase

The acceleration movement s_1 starts from zero velocity v_0 = 0. Its distance s_{acc} and the required time t_{acc} can be calculated with v_1 = 2 mm/s and a_1 = 5 mm/s^2.

$$\begin{aligned} s_{acc} &= \frac{{v_1}^2}{2 * a_1} = 0.4\,\text{mm} \\ t_{acc} &= \frac{v_1}{a_1} = 0.4\,\text{s} \end{aligned} \qquad (3)$$

The acceleration phase s_1 can be calculated for t = 0 to t = 400 ms with v_0 = 0 and a_1 = 5 mm/s^2.

$$s_1(t) = \frac{a_1 * t^2}{2} + v_0 * t \tag{4}$$

Constant Velocity Phase
Assuming that the deceleration phase takes the same time and distance as the acceleration phase, the distance s_{const} and the required time t_{const} can be calculated with s_{end} = 10 mm and v_2 = v_0 = 2 mm/s:

$$\begin{aligned} s_{const} &= s_{end} - 2 * s_{acc} = 9.2\,\text{mm} \\ t_{const} &= \frac{2 * s_{const}}{v_2 + v_0} = 4.6\,\text{s} \end{aligned} \tag{5}$$

Note that the previous final target of the last section must be added to each calculated target. Since the acceleration is zero while moving with constant velocity, the following simplified formula can be used to calculate the constant velocity phase s_2 for t = 0 to t = 4.6 s with v_0 = 2 mm/s:

$$s_2(t) = s_1 + v_0 * t \tag{6}$$

Deceleration Phase
By using the acceleration a_3 = -a_1, the deceleration phase s_3 can be calculated for t = 0 to t = 400 ms with v_0 = 2 mm/s and a_3 = -5 mm/s^2:

$$s_3(t) = s_2 + \frac{a_3 * t^2}{2} + v0 * t \tag{7}$$

Holding Phase
To add a holding phase s_4 to the end, the final target of the last section is repeated for t = 0 to t = 100 ms:

$$s_4(t) = s_3 \tag{8}$$

CONCLUSION

The general concepts of the MCS2 trajectory streaming as well as an example how to calculate stream targets for accelerated movements has been described in this user guide.
For more information on the actual programming and the used API functions and properties please refer to the *MCS2 Programmer's Guide* and the provided programming examples.

Sales partner / Contacts

Germany

SmarAct GmbH

Schuette-Lanz-Strasse 9
26135 Oldenburg
Germany

T: +49 441 - 800 879 0
Email: info-de@smaract.com
www.smaract.com

France

SmarAct GmbH

Schuette-Lanz-Strasse 9
26135 Oldenburg
Germany

T: +49 441 - 800 879 956
Email: info-fr@smaract.com
www.smaract.com

USA

SmarAct Inc.

2140 Shattuck Ave. Suite 302
Berkeley, CA 94704
United States of America

T: +1 415 - 766 9006
Email: info-us@smaract.com
www.smaract.com

China

Dynasense Photonics

6 Taiping Street
Xi Cheng District,
Beijing, China

T: +86 10 - 835 038 53
Email: info@dyna-sense.com
www.dyna-sense.com

Natsu Precision Tech

Room 515, Floor 5, Building 7,
No.18 East Qinghe Anning
Zhuang Road,
Haidian District
Beijing, China

T: +86 18 - 616 715 058
Email: chenye@nano-stage.com
www.nano-stage.com

Shanghai Kingway Optech Co.Ltd

Room 1212, T1 Building
Zhonggeng Global Creative Center
Lane 166, Yuhong Road
Minhang District
Shanghai, China

Tel: +86 21 - 548 469 66
Email: sales@kingway-optech.com
www.kingway-optech.com

Japan

Physix Technology Inc.

Ichikawa-Business-Plaza
4-2-5 Minami-yawata,
Ichikawa-shi
272-0023 Chiba
Japan

T/F: +81 47 - 370 86 00
Email: info-jp@smaract.com
www.physix-tech.com

South Korea

SEUM Tronics

1109, 1, Gasan digital 1-ro
Geumcheon-gu
Seoul, 08594,
Korea

T: +82 2 - 868 10 02
Email: info-kr@smaract.com
www.seumtronics.com

Israel

Trico Israel Ltd.

P.O.Box 6172
46150 Herzeliya
Israel

T: +972 9 - 950 60 74
Email: info-il@smaract.com
www.trico.co.il