TRIAMEC

Redefining
Motion Control

Application Note 160

TRAJECTORY STREAMING

This application note describes Trajectory Streaming via an (I)PC. For a Tama-based approach, consider [1].

Table of Contents

1 What is Trajectory Streaming.....1

2 Requirements for Trajectory Streaming.....2

3 API or Tama as Input Source.....2

4 How it works.....3
- Built-in Extensions.....3
- Custom Extensions.....3
- Registers of Interest.....4

5 Trajectory Streaming API Implementations.....4
- 5.1 TraSt File Player (coming soon).....4
- 5.2 TraSt PLC Streaming App (coming soon)...5
- 5.3 Customer-specific API Implementation....6

6 Specifications.....7

References.....7

Revision History.....7

1 What is Trajectory Streaming

Trajectory Streaming (TraSt) is a Triamec-specific software/firmware toolkit that lets you stream synchronized multi-axis motion paths with a sampling frequency of up to 50kHz directly to Triamec's high-speed servo drives from an (I)PC via Ethernet without any additional hardware, like a high-level controller, in between. Such high sampling rates virtually eliminate interpolation errors and enable ultra-precise motion control. Besides position data, also feed-forward current and velocity as well as custom extensions can be sent.

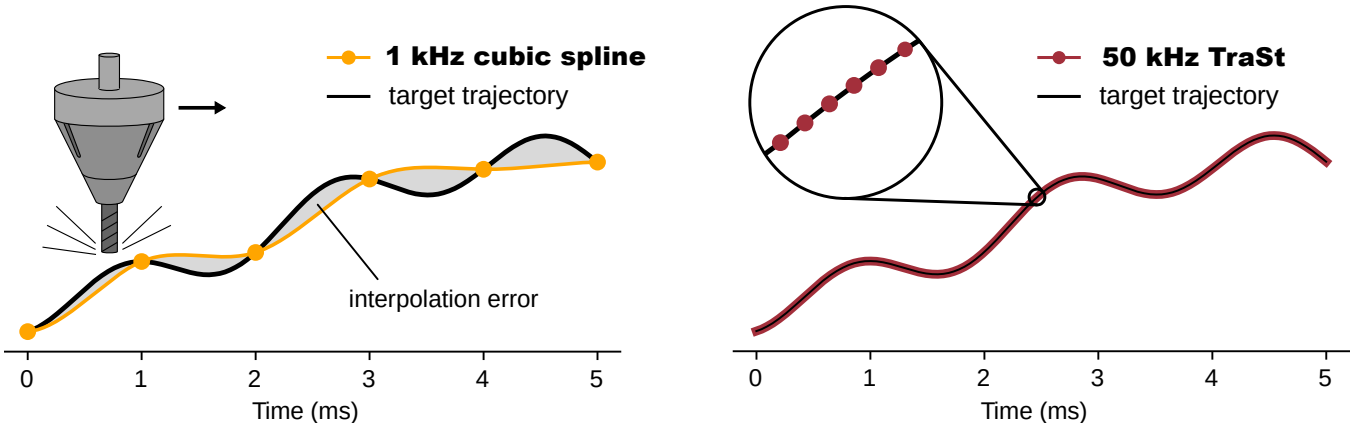

Figure 1: Trajectory Streaming reduces interpolation error significantly compared to traditional methods

2 Requirements for Trajectory Streaming

The following components are needed in order to use Trajectory Streaming:

- At least one Triamec Servo Drive each with:
  - Minimum Firmware Version 4.29.2
  - Trajectory Streaming Software Option “TS”
- When using more than one drive: A real-time bus to synchronize the drives (Tria-Link or EtherCAT)
- A data source (see section 5) to generate/send the trajectory data from:
  - An (I)PC running either:
    - The Trajectory Streaming Listener App
    - The Trajectory Streaming Player App
    - Any other implementation of the Trajectory Streaming API (.NET)

    or
  - A Tama-program running on the drive that generates trajectory data (see [1])
- If using an (I)PC:
  - An Ethernet connection to each drive
  - Trajectory data (with a look ahead, see section 6)

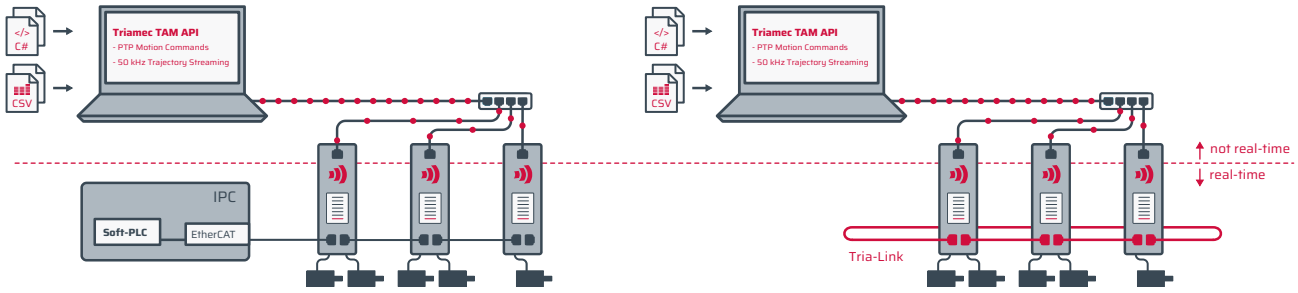

Figure 2: Trajectory Streaming setup with an (I)PC, an Ethernet switch and either an EtherCAT or Tria-Link bus. The trajectory itself can be supplied in the csv format provided by Triamec or via a custom implementation of the API

3 API or Tama as Input Source

Commonly, an implementation of the Trajectory Streaming API will be used, as it allows more flexibility and complexity in regard to the trajectory data and changes to it. In this case, the (I)PC will run an implementation of the Trajectory Streaming API, which controls the process from the computer’s side and serves as the interface for handing over trajectory data segment by segment, as the process continues. Whether you use an implementation of the Trajectory Streaming API provided by Triamec or your own implementation depends on your use case, as described below.

Using a Tama-program allows a standalone use of Trajectory Streaming without a (permanently) connected (I)PC. Instead of having to continuously send segments of trajectory data from the (I)PC, the Tama-program running on the drive generates the trajectory data. On the other hand, this means that the trajectory used for streaming is more or less static and can only be changed either by additional inputs used in the Tama-program or by changing the Tama-program itself, which requires a connection to an (I)PC. This approach is thus quite limited when compared to the approach described above but might be worth considering if a permanent (I)PC connection is not possible or if the trajectory desired is short, simple and static. For further information on this approach, consider [1].

4 How it works

The Trajectory Streaming API on the (I)PC allows establishing an (Ethernet) connection to all drives involved in the streaming process. Once all the parameters for the streaming are configured, the Trajectory Streaming can be started, meaning that the trajectory can be sent to the drives segment by segment. The drive stores the received trajectory data in a table, which acts as a circular buffer, and starts applying those values. The fuller this buffer is, the more stable the trajectory streaming process becomes in regards of jitter occurring on the (I)PC, the data transmission or other parts of the process.
Additionally, the process speed can be controlled by the feed override, which serves the same purpose as the Feedrate Override commonly found on CNC machines.
All drives involved need to be connected with the same real time bus (EtherCAT, TriaLink) to be able to synchronize the execution of the trajectory.

Built-in Extensions

Additionally to the position data, a current feed-forward as well as expected velocities can be sent with each position data point, which then applied by the firmware synchronously together. Please note that the unit of the current feed-forward is not [A] but **[A/(row/s)^2]** in order for the current to scale with different stream rates as well as changes to the feed override value.
The declaration of velocities is only needed in specific high speed spindle cases, where the spindle rotates so fast that less than 2 positions are streamed during one position modulo (which often corresponds to one full rotation of the spindle). This is necessary because the path planner would otherwise not be able to resolve the correct number of rotations between position data correctly. If possible, another solution is to further increase the streaming rate.

Custom Extensions

Besides the built-in feed-forward extensions, custom extensions can be sent too. These values have to be retrieved, used and/or applied via a custom Tama-program. Be aware that writing/reading register values and running Tama-programms run at a fixed rate of 10kHz, which might not match your streaming rate. However, it is possible to retrieve multiple (extension) values from the table in one Tama cycle. A detailed explanation can be found in [1].

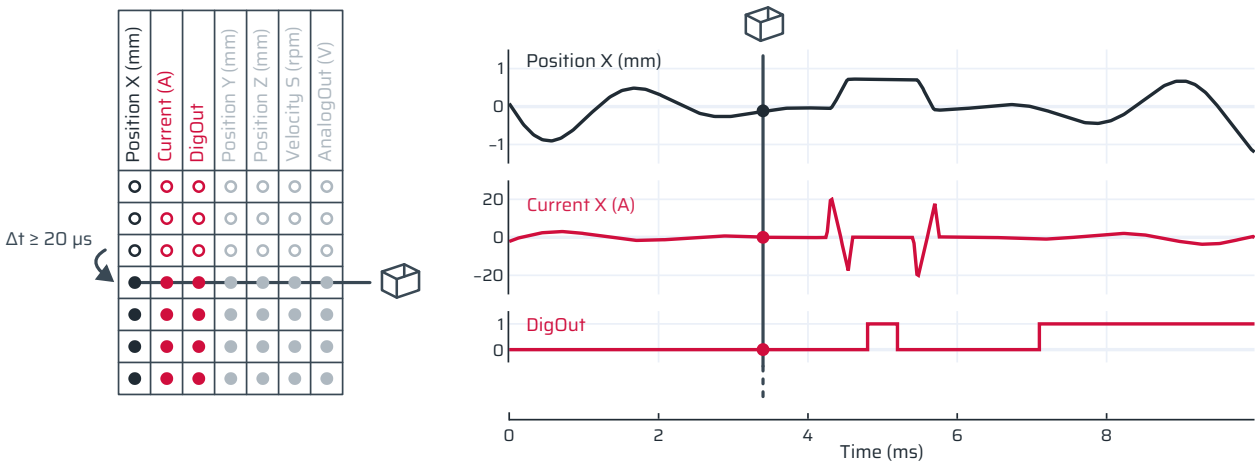

Figure 3: A maximum of 6 extensions can be sent together with each position value in Trajectory Streaming

Registers of Interest

The registers are all automatically set when using the Trajectory Streaming API but looking at the register values might help debugging.

- Axis.Commands.Pathplanner.Stream contains information about the setup of the stream
- Axis.Commands.General.Emergency.TrajectoryStreamError allows to set a specific reaction on the drive if there is an error while streaming (e.g. out of data)

Further explanations of the registers can be found in [1].

5 Trajectory Streaming API Implementations

5.1 TraSt File Player (coming soon)

This app is meant to provide an easy starting point for trying out Trajectory Streaming. You can find the source code under <LINK>. Furthermore it can serve as a template for your own, custom implementation, see our github page. The trajectory data itself as well as the metadata for the streaming (e.g. involved axes, stream rate etc.) have to be supplied in a specific .csv file format. A template file can be retrieved from the app itself. It does not allow a fully automated execution of your program and thus might not be feasible for your production environment out of the box. We recommend using a Customer-specific API Implementation to fully integrate Trajectory Streaming into your production software.

You can access the app via the Tam System Explorer > Assistants.

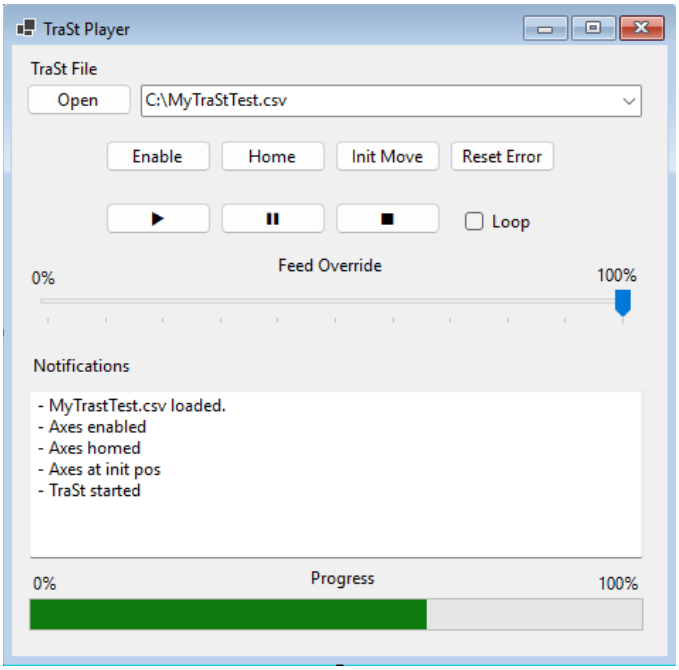

Figure 4: TraSt Player App

5.2 TraSt PLC Streaming App (coming soon)

This app allows you to start Trajectory Streaming from your PLC with a simple G-code command. The app "listens" to the PLC and starts executing Trajectory Streaming once receiving the command with the information on what to stream. On the PLC side, a custom function block provided by Triamec is needed as well as an active communication via either ADS or OPC UA between the App and the PLC. The code can be found on our github page and can be reused to seamlessly integrate in your existing framework.

You can access the app via the Tam System Explorer > Assistants.

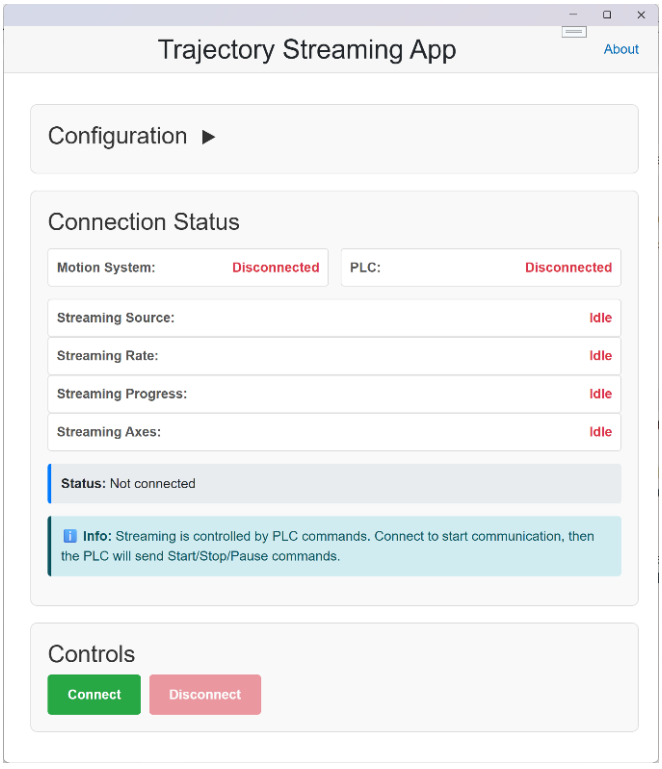

Figure 5: TraSt PLC Listener App

5.3 Customer-specific API Implementation

In order to incorporate Trajectory Streaming in your specific workflow/machine software and tailor it to your specific needs, the Trajectory Streaming API can be used directly. The Trajectory Streaming API uses the Motion System API, which is a convenience wrapper for the existing TAM API with a focus on motion control tasks. Please refer to our Trajectory Streaming code samples on github for further information.

6 Specifications

Specification	Value
Max. number of streaming axes	8 (tested, recommended), 16 (tested, possible), technical limit not yet known
Max. stream rate	50kHz (tested, recommended), 99kHz (tested, technical limit)
Max. number of extensions	6 per drive (built-in and custom extensions used for all axes combined)
Max. duration of trajectory	infinite (limited by the data handling on the (I)PC side)

Contraints	Explanation
Trajectory look ahead required	Since some time (0.5-1s) is required from reading the trajectory data segment from the source on the (I)PC to having it available on the drive and because this time is not constant, a look ahead is necessary. During streaming, the drive-side buffer contains future motion data. This look ahead depends on the sampling rate and can reach multiple seconds. When using continuously generated trajectories (in contrast to offline-generated files), this look ahead needs to be considered.
Trajectory data generation	Triamec may help generating trajectory data for testing purposes but strongly advises to build this expertise on customer site, as this process requires in-dept knowledge and tuning dependent on the machine and process.

References

[1] “Trajectory Streaming using Tama”, AN158_TrajectoryStreamingTama_EP001.pdf, Triamec Motion AG, 2026
[2] “MAxi API Reference”, "MAxi API Reference.chm, Triamec Motion AG, 2026
→ Temporarily available on https://triamec.com/en/support/documents-manuals
→ To be integrated in the common “TAM Software API Reference”

Revision History

Version	Date	Editor	Comment
001	2026-07-28	ns	Initial edit

Copyright © 2026
Triamec Motion AG
All rights reserved.

Triamec Motion AG
Lindenstrasse 16
6340 Baar / Switzerland

Phone +41 41 747 4040
Email info@triamec.com
Web www.triamec.com

Disclaimer

This document is delivered subject to the following conditions and restrictions:

- This document contains proprietary information belonging to Triamec Motion AG. Such information is supplied solely for the purpose of assisting users of Triamec products.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice.

Source Q:\doc\ApplicationNotes\
Destination T:\doc\Org\Templates
Owner ns