



Position Settling Analysis

Application Note 150

The position settling analysis features, provide simple measurement tools, to analyze the repeatability of a motion system. Using it can help to detect system changes. Such system changes can happen by aging effects, a mechanical or electrical damage or simply modifications in software or parameters.

Table of Contents

1	Introduction.....	1	2.4	ISE.....	3
2	Analysis Methods.....	2	2.5	ITSE.....	3
2.1	Standard Deviation.....	2	3	TwinCAT Function Block.....	4
2.2	Settling Time.....	2	3.1	Example.....	5
2.3	Move and Settling Time.....	2		Revision History.....	5

1 Introduction

In firmware version ≥ 4.17 we introduce on-board features for position settling analysis. These features are configured with a set of new *Parameters*, in:

`Axes[].Parameters.PositionController.Analysis`

As soon as a Method is configured, the calculation runs in the background and results are available in:

`Axes[].Signals.PositionController.Analysis`

2 Analysis Methods

Currently, the following analysis methods are implemented. In addition to the Result register, the register `Signals.PositionController.Analysis.Done`, indicates if an analysis is running (False), or the result is ready (True).

The below analyses are triggered as follows, except the *Standard Deviation* method:

- The path planner state changing to *Standstill*,
- and by a value change in the register `Axes[].Commands.CurrentController.InjectedIq`

Accordingly, an axis is analyzed regarding reference tracking or disturbance rejection.

Note Each trigger restarts the analysis. If an application continuously sets `InjectedIq`, i.e. cogging compensation with *Tama*, then this feature cannot be used.

2.1 Standard Deviation

Calculates the standard deviation and averages over Time and updates with Time.

Parameters:

- `PositionController.Analysis.Method = StandardDeviation`
- `PositionController.Analysis.Time = Time` period to average.

Result in `Signals.PositionController.Analysis.Result`:

$$StandardDeviation = \sqrt{\frac{\int_0^{Time} err^2 dt}{Time}}$$

2.2 Settling Time

The *SettlingTime* is the duration from the end of a move until the position error is within the Window for a defined Time. See Figure 1 for the visual definition of the parameters and the result.

Parameters:

- `PositionController.Analysis.Method = SettlingTime`
- `PositionController.Analysis.Time = Minimum` in-window time for a valid measurement
- `PositionController.Analysis.Window = The window size` (error band = 2 * window)

Result:

`Signals.PositionController.Analysis.Result = Settling time` for the last move, measured from the end of the trajectory.

2.3 Move and Settling Time

The result is the duration from the start of a move, until the position error is within the Window for a defined Time.

Parameters:

- PositionController.Analysis.Method = MoveAndSettlingTime
- PositionController.Analysis.Time = Minimum in-window time for a valid measurement
- PositionController.Analysis.Window = The window size (error band = 2 * window)

Result:

Signals.PositionController.Analysis.Result = Settling time for the last move, measured from the start of the trajectory.

The move time while following a trajectory is available at Axes[].Signals.PathPlanner.MoveDuration.

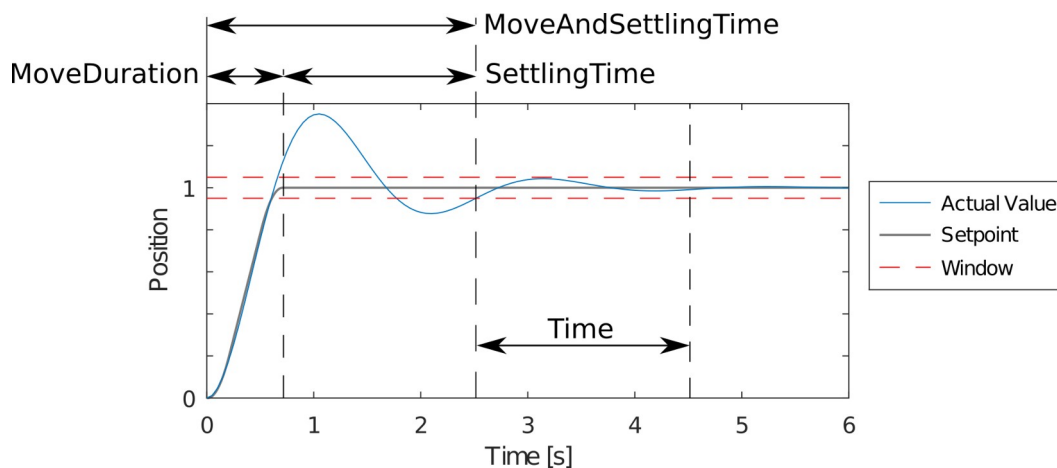


Figure 1: Visualized Move and Settling modes

2.4 ISE

The result of the *ISE* (Integral Square Error) calculation, for the duration Time, after the end of a move (path planner trajectory).

Parameters:

- PositionController.Analysis.Method = ISE
- PositionController.Analysis.Time = Time period for calculation.

Result in Signals.PositionController.Analysis.Result:

$$ISE = \sqrt{\frac{\int_0^{Time} err^2 dt}{Time}}$$

2.5 ITSE

The result of the *ITSE* (Integral time Squared Error) calculation, over the duration Time, after the end of the move (path planner trajectory).

Parameters:

- PositionController.Analysis.Method = ITSE
- PositionController.Analysis.Time = Time period for calculation.

Result in Signals.PositionController.Analysis.Result:

$$ITSE = \sqrt{\frac{2 \int_0^{Time} t \cdot err^2 dt}{Time^2}}$$

3 TwinCAT Function Block

A *Function Block (FB)* is available for ease of use of the analysis feature from *TwinCAT*.

- *Tria-Link*: TL_PositionSettlingAnalysis
- *EtherCAT*: TE_PositionSettlingAnalysis

The *FB* must be triggered with a positive edge on *Execute*. Output *Active* indicates (TRUE) that the analysis parameters are active. The *Done* output is TRUE, when a new *Result* is available. *Error* becomes TRUE if a timeout occurs. The timeout duration is at least the *DEFAULT_ADS_TIMEOUT* (global constant, set to 5 seconds) plus the configured *AnalysisTime*. For external triggers (*InjectedIq* = 0) 2x the *DEFAULT_ADS_TIMEOUT* applies. A new measurement result is expected within this time.

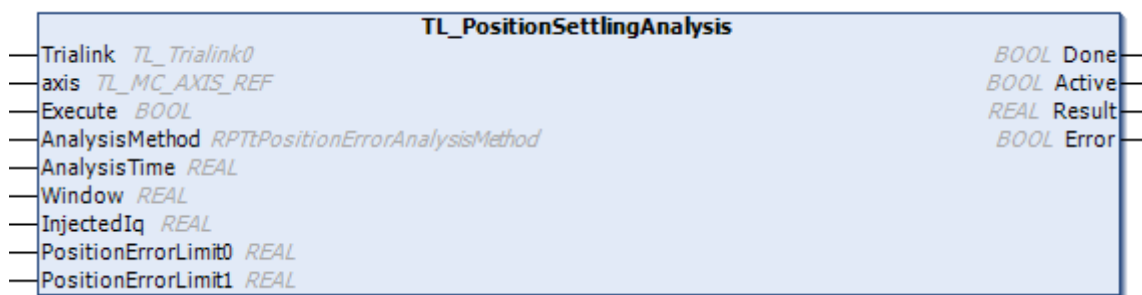


Figure 2: FB for position settling analysis (Tria-Link)

The analysis parameters must be set according to the analysis method used and considering the axis/application limits:

- *AnalysisMethod* to choose the method to be used, see also chapter 2.
- *AnalysisTime* to set the characteristic time for analysis in [s].
- *Window* to set the size (error band = 2 * window) in [drive units]
- *InjectedIq* to set the current in [A] added to the position controller output during measuring (will be set if value ≠ 0 and set back to 0 after measuring)
- *PositionErrorLimit0* to set the position error limit of Controller 0 in [drive units] (will be set if value ≠ 0 and set back to the previous value)
- *PositionErrorLimit1* to set the position error limit of Controller 1 in [drive units] (will be set if value ≠ 0 and set back to the previous value)

Note The values `InjectedIq` and `PositionErrorLimit` are changed during the measurement if their values $\neq 0$. If `InjectedIq` = 0, the FB waits for an external trigger, e.g. Standstill or change of `InjectedIq` (except the *Standard Deviation* method). Both `InjectedIq` and `PositionErrorLimit` will be set back at the end of measuring.

3.1 Example

An example can be found on:

- www.triamec.com > Products > Software > Beckhoff-TwinCAT Integration > Example with Tria-Link and NC

Revision History

Version	Date	Editor	Comment
001	2023-03-14	sm	Document Creation
002	2023-06-14	sm	Add trigger variant <code>InjectedIq</code> .
003	2023-06-16	sm	Explicit trigger declaration, excluding Standard Deviation since fw 4.19
004	2023-08-10	rb	Function Block <code>PositionSettlingAnalysis</code>

Triamec 中国-销售/技术服务中心

TEL: (0) 13776269231 Email: sales@tyt-motion.com URL: www.triamec.com.cn

Copyright © 2023
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.