Axis Compensation

Application Note 140

The *Axis Compensation* module allows to compensate systematical position deviations of an axis. The axis compensation can significantly improve the absolute accuracy of the machine.

Table of Contents

Document AN140_AxisCompensation_EP
Version 005, 2024-08-21
Source Q:\Doc\ApplicationNotes\AN140_AxisCompensation\
Destination T:\doc\ApplicationNotes\
Owner dg

1 Introduction

1.1 What is Axis Compensation

Axis compensation refers to a method counterbalancing a measurable absolute positioning error based on measurable input parameters. The positioning error is usually measured with an external calibration device and the resulting values can be used to correct the encoder positions. Sometimes this procedure is also called mapping.

Common effects to be compensated are:

- Encoder Alignment & geometric imperfections: The encoder is never perfectly parallel to the moving axis and never perfectly straight, which causes a discrepancy between the encoder feedback position and the actual position. For multiaxis systems, the alignment between different encoders additionally causes discrepancies. Similar effects are observable for geometric imperfections of the positioning system itself.
- Thermal Effects: As materials typically elongate with rising temperatures, thermal effects can play a significant role on the machine's geometry. This is usually not an issue if the machine can be measured and compensated at the same steady-state operating temperature. If the machine has a lot of transient operating points e.g. during start-up or varying operating conditions, a compensation reference measured at a single temperature profile might not be sufficient. In this case, temperature can be used as additional input to the compensation to further improve the absolute accuracy.

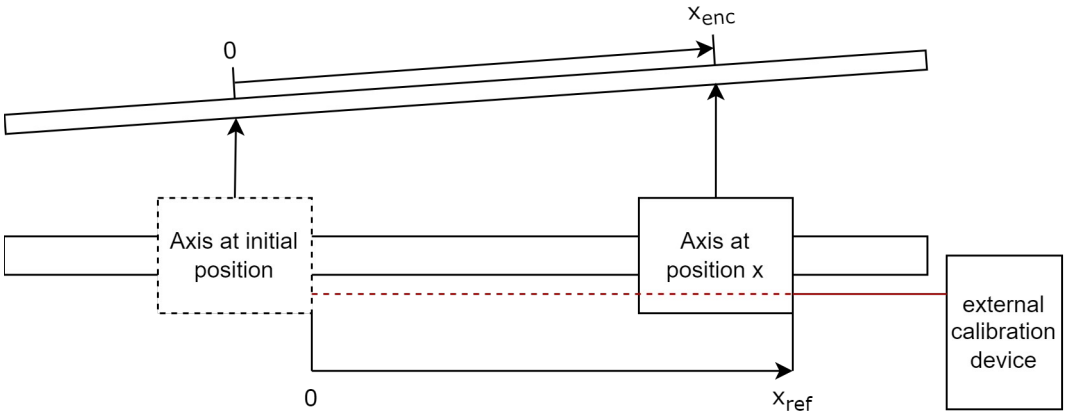

Figure 1: An (exaggerated) encoder misalignment causes the encoder feedback position x_{enc} to be bigger than the actual position x_{ref}, which means the executed absolute move will be shorter than intended

Other effects may as well be compensated, as long as they are time-invariant and a measurable variable(s) can be used as input for the compensation. For the Axis Compensation the correction value is added to the encoder position. In contrast *Cogging Compensation* is applied to the setpoint current. Therefore *Cogging Compensation* is offered as a separate tool by *Triamec Motion AG* and is thus not scope of this documentation or the axis compensation in general.

1.2 Compensation Tables (*TAM Table*)

The actual compensation values are stored in a table called *TAM Table*. *Triamec Motion AG* uses a spe-

cific binary file format *.TAMtbl* for this. It consists of a header and a flat list of numbers. For detailed information, refer to 4.1 and AN124 [1].

1.3 Table lookup inputs (*Dimension Source*, *Dim(1-3)Source*)

In order to look up the correct compensation value in the compensation table described above, between one to three inputs, called *Dimension Source*, have be used, referring to 1D, 2D or 3D compensation. A *Dimension Source* is a *Register Address* of the same drive but not necessarily from the same axis. The corresponding *Registers* on the drive are called *Dim1Source*, *Dim2Source* and *Dim3Source*, which will be referred to as *DimSources* from now on. An unused/unset *DimSource Register* contains the value 0. A used/set *DimSource Register* contains on the firmware level the hex-value of the *Register Address* used as an input. Higher level software, such as the *TAM System Explorer* might display an alias e.g. the name of the *Register* instead of the hex-value.

1.4 Axis Compensation Commands

To control the activation of the axis compensation, the Register Axes[].Commands.Compensations.Axis.Activation is used. The following commands can be set:

- Disabled: The axis compensation is not active
- Evaluate: The axis compensation *TAM Table* and the *Dimension Source* are evaluated but the compensation will not be applied. The evaluation will throw errors in case of any mismatches.
- AddToEncoders: The axis compensation *TAM Table* and the *Dimension Source* are evaluated and if no error is thrown, the compensation will be activated.

Remark: Homing.State = HomingDone is necessary to apply the axis compensation.

1.5 Axis Compensation Output

In Axes[].Signals.Compensations.Axis, the current compensation output added to the encoder position (if activated) as well as the current Dim(1-3)Index can be observed.

2 Axis Compensation Wizard

2.1 What is the Axis Compensation Wizard

In order to simplify and accelerate the process of setting up an axis compensation, *Triamec Motion AG* provides a specific software tool called Axis Compensation Wizard. It consists of the following base functionalities for each axis:

- Parse compensation data from a *.csv* file to the binary *.TAMtbl* format used for *TAM Tables*
- Get and set the axis compensation *TAM Table*
- Get and set the *Dimension Source* e.g. the input parameter(s) used to look up compensation values
- Activate and deactivate the axis compensation

All business logic used in the Axis Compensation Wizard is accessible via the TAM API. This allows you to integrate all functionalities in your own UI or using them in automation code, see section 4.2.

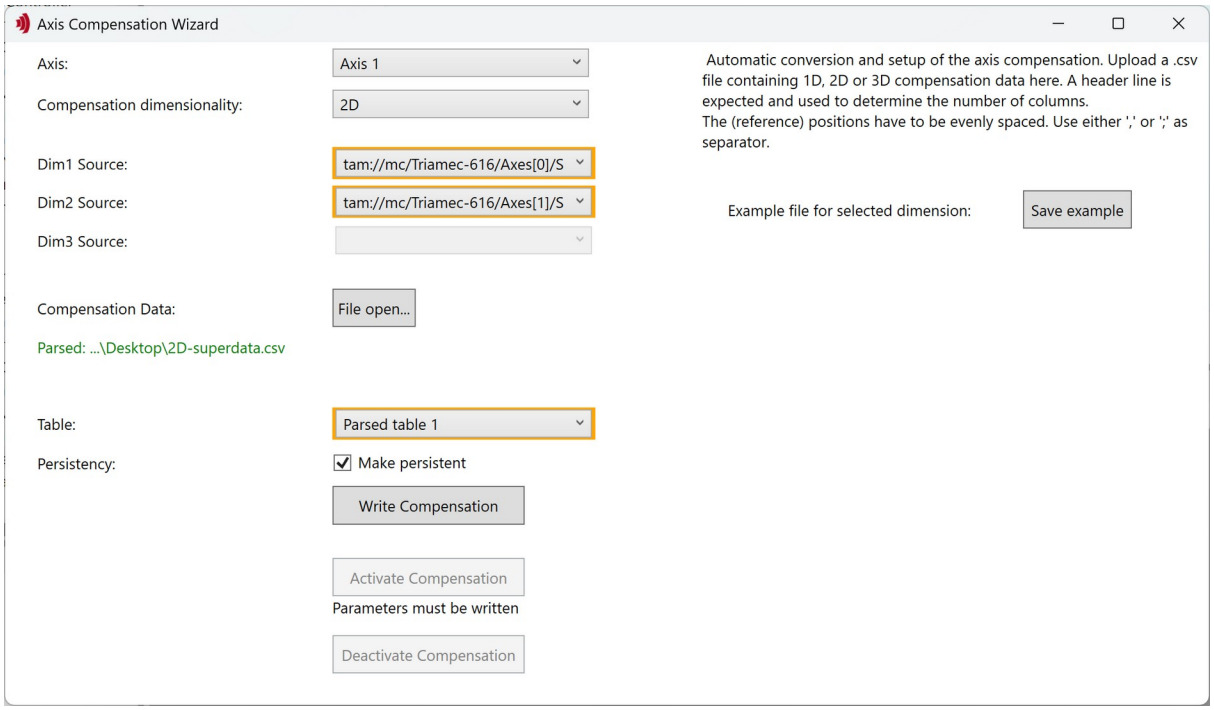

Figure 2: The Axis Compensation Wizard with an unwritten compensation

2.2 Opening the Wizard

The Axis Compensation Wizard can be opened from within the TAM System Explorer in the top menu `Assistant > Axis Compensation Wizard` as shown in Figure 3 or as a Standalone Application by opening the `AxisCompensationWizard.exe`, which can be found in the same folder as the `TamSystemExplorer.exe`.

Remark: Due to the flash wear error described in 2.4, it is generally recommended to use the *Axis Compensation Wizard* within the *TAM System Explorer*

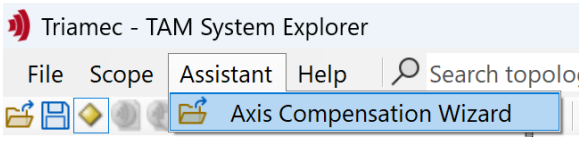

Figure 3: Opening the Axis Compensation Wizard from within the TAM System Explorer

2.3 Input file format

Currently the following file formats can be parsed:

- Comma separated values (.csv):
  - Lines that start with `//` e.g. comments are ignored by the parser (can be used for metadata)
  - A header line is mandatory and has to be separated with the below listed separator as it will be used to identify the columns and to determine the number of columns
  - Accepted separators are ',' and ';'

- The first columns specify the reference values – one column per dimensionality, followed by columns with the compensation values
- One or more compensation columns which share the same reference values are possible. The Assistant will allow to select which compensation column to apply to the selected axis
- Each line (except the header line) must contain a reference value for each compensation dimension and a compensation value (one for each table). Therefore *#columns = dimensionality + #tables*

Example files can be generated and saved in the Wizard itself, see below.

2.4 UI Elements

Axis

Select the axis you want to apply the compensation to. You can still use *DimSources* from other axes as long as they are on the same drive. All axes found during the start up of the application are shown. Later connected/disconnected drives are not added/removed.

Compensation dimensionality

Select the dimensionality for your compensation. Necessary to parse files correctly and activate the necessary *DimSources*.

DimSource

Select which register(s) to use as an input to the compensation. All enabled inputs must contain a valid register address. Also external signals can be used as *DimSource* by mapping these signals to one of the available Application.Float registers.

Remark: Currently there is only a shortlist of possible Registers available to be used. Adding a custom register by address will be possible in future versions.

Compensation Data

Select a file to parse tables from. Not necessary if the table to be used is already set on the axis.

Table

Select which table to use for the compensation. The list consists of the already uploaded table (if available) and/or parsed tables generated from the uploaded Compensation Data.

Persistency

Select if the table shall be persistent or volatile. A volatile table will be lost after a reboot of the drive. Writing or deleting a persistent table causes flash wear. You can change the persistency state of an already uploaded table.

Remark: Writing many persistent tables in a short amount of time can lead to a flash wear error. In such a case, wait some minutes before writing a new table. Use the *Axis Compensation Wizard* within the *TAM System Explorer* to be aware if this error occurs.

Write Compensation

Write all (changed) parameters to the drive. Changed parameters are indicated by an orange border around the parameter. If no parameter changed or the chosen parameters are incompatible, writing will not be possible.

Activate Compensation

Activate the compensation. This is not possible if the compensation is changed (unwritten), the selected settings are incompatible or if the compensation already is activated.

Deactivate Compensation

Only possible if the compensation is active.

3 Implementation

3.1 Compensation Sign

It is assumed that the value for the axis compensation Δx is provided as the difference between the external reference position x_{Ref} and the encoder position x_{Enc}.

$$\Delta x = x_{Ref} - x_{Enc} \tag{1}$$

Accordingly the axis compensation Δx is added to the encoder position x_{Enc} by the drive to reconstruct the reference position x_{Ref}.

3.2 Enabling of the Compensation

When enabled, the compensation is smoothed with a ramp to avoid a jump. The duration of the ramp is 10ms.
The compensation is only applied, if the axis is homed.

3.3 Interpolation Type

For the compensation within the set-points linear interpolation is used for one dimensional interpolation bilinear interpolation for two dimensional interpolation and trilinear for three dimensional interpolation.

4 Advanced Topics

4.1 Compensation Tables detailed (*TAM Table*)

The actual compensation values are stored in a table called *TAM Table*. *Triamec Motion AG* uses a specific binary file format *.TAMtbl* for this. It consists of a header and a flat list of numbers. For detailed information, refer to AN124 [1]. In the case of an axis compensation, float values are used. In order to find the correct return value in the flat list of numbers, the corresponding index of the list element has to be determined. A variable used for this index lookup (meaning it is an input to the table) is called

Dimension Source. The firmware calculates this index based on the *Dimension Source* value and the following 3 parameters:

- *StartValue:* The *Dimension Source* value of the first data point in this dimension
- *Distance*: The *Dimension Source* value between data points in this dimension
- *Size*: The number of data points in this dimension

Depending on its dimensionality, a *TAM Table* is associated with 1 to 3 such *Dimension Sources*. The flat list approach means that the reference positions have to be equidistant in each dimension used, such that they can be described with the 3 parameters listed above. This lookup is executed at a frequency of 10kHz. All tables can be accessed via the file system described in AN124 [1].

4.2 TAM API

All functionalities of the *Axis Compensation Wizard* are accessible in the *TAM API* (.NET) via the Triamec.Tam.UI NuGet. All business logic used in the *Axis Compensation Wizard* is located in the top-level class `AxisCompensator`. This allows you for example to program your own automated axis compensation set up. Further information about the TAM API can be found in the TAM API Developer Manual [2].

5 Tama Approach

The axis compensation can also be done with a Tama program. This approach is more complicated and is generally only recommended if your desired axis compensation cannot be realized with the firmware-based axis compensation, which is used by the wizard (e.g. you need a non-linear interpolation).
The following chapter describes this approach and how to use it.

5.1 Introduction

Usually the compensation data is based on calibration data, recorded with an external measurement system. This data has to be converted to the binary *TAM Table* format, to deploy it to the drive. On the drive side, the compensation is evaluated and applied by a *Tama* program. This *Tama* program is provided by *Triamec Motion AG* on request. Also the source code is available and allows to implement user specific solutions.

This chapter first describes the required steps to setup the axis compensation. Afterwards, some remarks about the implementation are provided.

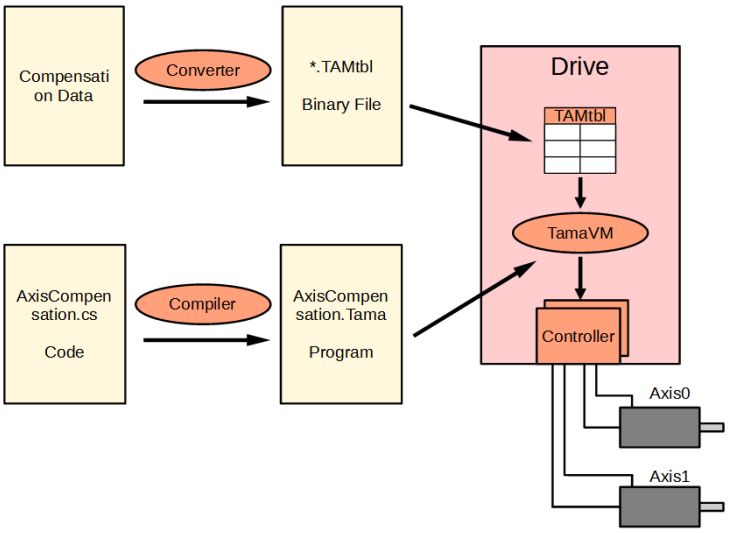

Figure 4: Overview of Setup with TAM Table and Tama program.

5.2 Setup the Axis Compensation

Transfer the Compensation Data

To transfer the compensation data either the *Axis Compensation Wizzard* can be used (see chapter 2) or the data can be converted and loaded manually. The following section describes the requred steps to to manually convert and load the data:

1. The compensation data need to be provided as a binary file in the *TAM Table* format. A detailed description of the table format can be found in [1]. Depending on the format of the compensation data, a converter has to be programmed. *Triamec* provides converters for some data formats. Please contact *Triamec Motion AG* for more information.
2. The transfer of the compensation data to the drive is done via the web server of the drive. See [1] for how to access the web server with a browser. The table can be transferred to one of the

predefined **.TAMtbl* files listed on the **Directory** page (Figure 5). It is recommended to copy the desired path from the list into the clipboard.
Change to the *transfer to drive* page and paste the file path to the **Filepath in drive** text box. Click **Browse...** to select the binary file containing the compensation data (Figure 6). Press Start to initiate the transfer.

3. In case the Persistent register in the table header is set to true, the table is stored permanently to the drive. Alternatively the table can be stored persistently by setting the following registers:
   Application.Tables.Small1.Header.Persistent = True
   Application.Tables.Small1.Command = Commit

Directory

File	actual size bytes	maximum size bytes
tables/small1.TAMtbl	304	65360
tables/small2.TAMtbl	0	65360
tables/small3.TAMtbl	0	65360
tables/small4.TAMtbl	0	65360
tables/small5.TAMtbl	0	65360
tables/small6.TAMtbl	0	65360
tables/small7.TAMtbl	0	65360
tables/small8.TAMtbl	0	65360
tables/large1.TAMtbl	0	2096976
tables/large2.TAMtbl	0	2096976

Figure 5: Directory page.

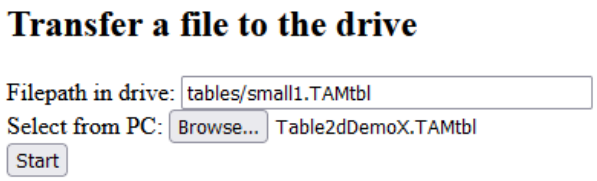

Figure 6: File transfer page.

Load and Run the Tama Program

1. The source code of the *Tama program* `AxisCompensation.cs` can be requested from *Triamec Motion AG*.
2. The *Tama program* might be adjusted depending on the application. The following implementation needs to be verified and might be adjusted:
   a) For one or two dimensional compensation, uncomment the corresponding preprocessor directive `#define compensation1D` or `#define compensation2D`.
   b) The compensation is implemented for both Axis[0] and Axis[1]. If only one axis has to be compensated, the code for the other axis has to be commented out.
   c) In the sample code table *small1.TAMtbl* is used for the compensation of Axis[0] and *small2.TAMtbl* for the compensation of Axis[1]. This needs to be adjusted according to the table the compensation data was transferred to.
   d) For one dimensional compensation the setpoint position of Axis[0] is used as the variable for the interpolation. For the two dimensional interpolation the setpoint for Axis[0] is used as the first variable and the setpoint of Axis[1] as the second variable for the bilinear interpolation.
   e) The interpolated compensation value is applied to Commands.PositionController.Encoders[].InjectedPosition, which is then added to the encoder value by the firmware.
   f) The compensation is activated as soon as Homing.State = HomingDone and the axis is enabled.
   g) Initially, the connection of the compensation is smoothed via a ramp to avoid a jump. The duration of the ramp is defined by the constant `cRampTime`.

h) It is assumed that the value of the axis compensation Δx provided as the difference between the external reference position x_{Ref} and the encoder position x_{Enc} ($\Delta x = x_{Ref} - x_{Enc}$). Therefore in the *Tama* program the evaluated axis compensation Δx is added to the current encoder position x_{Enc} program to reconstruct the reference position x_{Ref}.

i) For the compensation within the set-points linear interpolation is used for one dimensional interpolation and bilinear interpolation for two dimensional interpolation. The linear interpolation has the effect, that the velocity and its derivatives are discontinuous. This discontinuity can affect the surface at the location of the setpoint in some cases. If such an effect is detected, may an extension of the interpolator to a higher order may improve the situation, e.g. Qubic-B-Splines.

3. We recommend Visual Studio to build the *Tama* program.
4. See [3] on how to download the *Tama* program, enable the *isochronous TamaVM* and save it persistent on the drive.

References

[1] “Triamec Drive File System”, AN124_Filesystem_EP005.pdf, Triamec Motion AG, 2023

[2] “TAM API Developer Manual”, SWNET_TamApiDeveloperManual_EP048.pdf, Triamec Motion AG, 2024

[3] “Tama Real-Time Drive Programming – User Guide”, SWTAMA_UserGuide_EP001.pdf, Triamec Motion AG, 2024

Revision History

Version	Date	Editor	Comment
001	2021-09-14	dg	How to setup axis compensation
002	2023-04-26	dg, sm	General update, doc template, adapt to updated Tama program
003	2024-06-14	ns	Add Wizard, remove Tama program
004	2024-06-26	ns	Add TAM API remark, update UI element section
005	2024-08-19	ns	Merge with Tama approach document

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.