Impulse Decoupling

Application Note 143

This document describes the support of impulse decoupled axis setups by Triamec Drives.

Table of Contents

1 Summary

Impulse decoupled axes require the evaluation of two encoders, as either the axis position or the commutation position is the difference or sum of the two encoders. To support the required dynamics for

Document AN143_ImpulseDecoupling_EP
Version 003, 2025-02-18
Source Q:\doc\ApplicationNotes\
Destination T:\Doc\ApplicationNotes
Owner sm

correct positioning and commutation, this is implemented directly in the 100kHz control loop of the firmware.

2 Preconditions

Currently the calculation needed for the axis commutation is done directly on the encoder counters. Therefore the following conditions apply.

1. Both encoders must have the same resolution.
2. Both encoders must be of the same type in regard to the configuration at Axes[].Parameters.PositionController.Encoders[].Type.
3. Both encoder signals must be available through an encoder interface on the *Axis* Drive.

Note In special cases where condition 3 cannot be met (i.e. gantry setup), an encoder splitter can be used to route the required signal to the corresponding drive.

3 Definitions

Different terms are used in the context of the impulse decoupling technology. Triamec Motion AG uses the following terms for specific subsystems of such a setup.

Topology The architecture or concept of an axis with impulse decoupling.

Axis The term *Axis* is used as a keyword pointing to the moving part affecting the location of the point of interest.

Stator The term *Stator* is used for the part that compensates the excitation. Although it's a moving part in this context, it usually contains the stator component of the motor.

Base The term *Base* refers to the static reference of the axis.

4 Topologies

The required evaluation of the encoder position depends on the arrangement of the encoders. Triamec drives only support topology A of the visualized ones in Table 1.

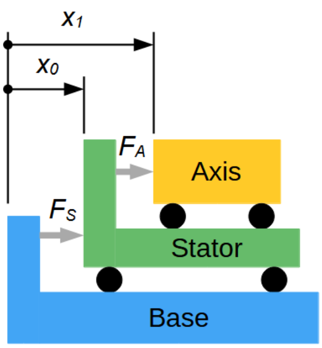
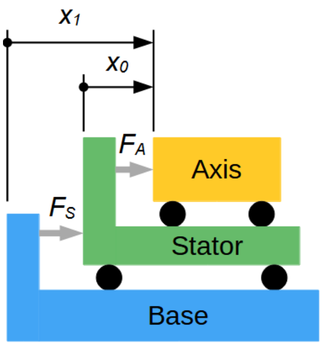
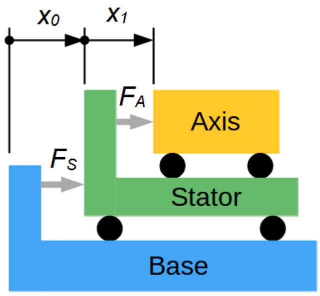
	Topology A 	Topology B 	Topology C 
Axis Controller	x_1	x_1	$x_0 + x_1$
Axis Commutation	$x_1 - x_0$	x_0	x_1
Stator Controller	x_0	$x_1 - x_0$	x_0
Stator Commutation	x_0	$x_1 - x_0$	x_0

Table 1: Topology A supported by Triamec drives

The supported topology A is defined as follows.

- The setup has two encoders X_0 and X_1 positioned to a static *Base*.
- Encoder X_0 measures the position of the *Stator* in reference to the *Base*.
- Encoder X_1 measures the position of the *Axis* in reference to the *Base*.

By this definition the position for commutation angle for the *Axis* motor has to be calculated, based on the position of both encoders.

The encoders can be connected to all encoder interfaces on the drive, also to option modules of type EN or EH. To configure the encoder topology in the drive according to the axis topology shown above, refer to [1].

Note	The supported axis topology enables actuated stator positioning and also passive stator positioning. In case of a passive positioning system the stator controller and stator commutation in Table 1 are obsolete.
Note	The impulse decoupling feature is available from firmware version 4.14 upwards.

5 Commissioning

To successfully commission an impulse decoupled axis the following procedure is recommended.

5.1 Coordinate System

The impulse decoupling feature requires that both encoders count in the same direction. Therefore the very first step is to define the direction of the *Axis* according to specs.

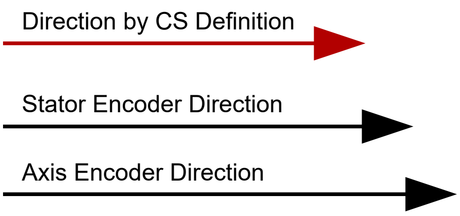


Figure 1: Coordinate System and Encoder Directions

Ensure the directions match to the definition during the commissioning steps in the next chapters. If your setup has a passive *Stator* positioning system setup the *Stator* encoder now and skip chapter 5.2.

5.2 Stator

We first set up the *Stator*. Therefore follow the *Setup Guide* [2]. Ensure at least the following steps before proceeding to the *Axis.*

- Set all parameters to reach a stable and stiff control setup.
- Verify the above step by checking the commutation and simple step movements.
- Setup the homing procedure for the *Stator* and verify it by running the homing sequence manually. See also [3] for more information about homing.
- Check the position limits for the *Stator*, so that an error is thrown when excited too much.
  Axes[].Parameters.PathPlanner.PositionMaximum
  Axes[].Parameters.PathPlanner.PositionMinimum

5.3 Axis

To commission the *Axis*, enable the *Stator* or fix it mechanically.

Caution Never enable the *Stator* when mechanically fixed!

Now setup the *Axis* as if it was a standard axis, following the Setup Guide [2]. The Bode Measurement can be done as well as if the *Stator* was rigidly connected to the *Base*.

- Set all parameters to reach a stable and stiff control setup.
- Verify the above step by checking the commutation and simple step movements.
- Setup the homing procedure for the *Axis* and verify it by running the homing sequence manually. See also [3] for more information about homing.
- Check the position limits for the *Axis*.
  Axes[].Parameters.PathPlanner.PositionMaximum
  Axes[].Parameters.PathPlanner.PositionMinimum

Commutation Source

At this point we set up the commutation source for the *Axis*. Therefore choose the correct setting in the following register.

Axes[].Parameters.Commutation.Source

The setting depends on the commutation direction which changes with the setting Axes[].Parameters.-Motor.InvertDirection. Therefore revert to this register on both axes and follow Table 2 for the commutation source calculation. As the encoder index depends on the setup, we use *Axis* and *Stator* in place.

Motor.InvertDirection		Commutation.Source
Axis	*Stator*	
False	False	Encoder[Axis] - Encoder[Stator]
True	False	- Encoder[Axis] - Encoder[Stator]
False	True	Encoder[Axis] + Encoder[Stator]
True	True	- Encoder[Axis] + Encoder[Stator]

Table 2: Axis Commutation.Source Calculation vs. Motor.InvertDirection Settings

6 Modes of Operation

Different modes of operation are possible in an impulse decoupled axis setup. A mechanical stator positioning is usually designed as a spring-damping-system. If the axis setup experiences external forces (i.e. from cable trains), active *Stator* positioning can lead to better results.

With an actively positioned stator the following behaviors can be realized.

6.1 Stiff Stator

This mode is convenient to simulate a stator which is rigidly mounted. While this has nothing to do with impulse decoupling, it is mainly used for initialization routines (i.e. homing) and commissioning purposes. This mode of operation is realized by configuring a stiff controller.

Note The applied forces must not exceed mechanical limitations and current limitations of the *Stator* motor.

6.2 Active spring-damping-system

With this mode of operation the position controller of the stator stimulates a spring-damping-system. The integrator part of the controller can be used to compensate static forces caused by cables, friction or gravity, etc.

6.3 Virtually decoupled

In this mode, setpoints for the *Stator* are continuously calculated, based on the mass ratio between *Axis* and *Stator*. A weak position controller is used to follow these setpoints. The *Stator* behaves like a free movable mass and the controller compensates external forces. This mode of operation has to be realized with a *Tama* program.

7 Tama Program Sample

The Tama program sample implements the enabling and homing of the stator and the main axis. After the homing is done, a weak controller is applied to the stator axis and the main axis is ready to be operated in decoupled mode.

Load and run the Tama program:

1. The source code of the *Tama* program used for the demonstration of the impulse decoupling 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:
    1. In the sample code the main axis is assigned to Axis[0] and the stator axis to Axis[0]. This can be adjusted by modifying the following lines in the ImpulseDecoupling.cs file:

    ```
    AxisHandler mainAxis = new AxisHandler(AxisHandler.AxisIndex.Axis_0);
    AxisHandler statorAxis = new AxisHandler(AxisHandler.AxisIndex.Axis_1);
    ```
3. To build the *Tama program* Visual Studio (Express) 2017 is recommended.
4. See [1] for how to download the Tama program to the drive, enable the isochronous TamaVM and save it persistent on the drive.
5. Set the register Axes[].Parameters.PathPlanner.Mode of the stator axis to Transformation.

7.2 Parametrization and Control

The following parameters are used to configure the impulse decoupling. These parameters need to be configured and permanently stored before the decoupling is activated.

Tama Parameters

Application.Parameters.Floats[0]...Floats[255]

Index	Address	Description
0	0x00238101	Moving mass of main axis [kg]
1	0x00238102	Moving mass of stator axis [kg]
2	0x00238103	Proportional controller gain Kr of the stiff controller
3	0x00238104	Integrator gain Ki of the stiff controller
4	0x00238105	Differentiator gain Kd of the stiff controller
5	0x00238106	Lowpass time constant T1 of the stiff controller
6	0x00238107	Bandwidth of the weak controller used when the impulse decoupling is active.
7	0x00238108	Damping of the weak controller used when the impulse decoupling is active
8	0x00238109	Slow velocity used for repositioning of the axis during activation

Tama Variables

In contrast to Tama parameters, Tama variables are used for values which are not stored permanently. The following variables are all set by the Tama program.

Application.Variables.Integers[0]...Integers [31]

Index	Address	Description
0	0x00238701	Tama version
1	0x00238702	Bitfield for Warnings and Errors
2	0x00238703	Displays the sub-state of the currently active mode

Tama Command

The Tama Command is used to controll the state machine of the Tama program.
Application.TamaControl.IsochronousMainCommand
Register Address: 0x00237D00

Value	Command	Description
0	NoCommand	No action. Register is reset to NoCommand after successful command dispatch.
1	Activate	Activate impulse decoupling
2	Disable	Disable impulse decoupling
3	ResetWarnings	Clear the warning bitfield (Variables.Integers[0]).
4	ResetError	ResetError

Tama States

The Tama state display the current state of the Tama state machine.

Application.TamaControl.IsochronousMainState

Register Address: 0x00237E00

Value	State	Description
0	Disabled	Stator axis is disabled.
1	Activation	Running activation routine.
2	Active	Impulse decoupling is active
3	Disabling	Disable impulse decoupling

Warnings and Errors

Bitfield in Application.Variables.Integers[1].
Register Address: 0x00238702

Bit	Warning	Description
0	DeviceOrAxisError	Drive or axis is in error state. Try to reset with ResetError or ResetWarning command.

8 Flow Charts

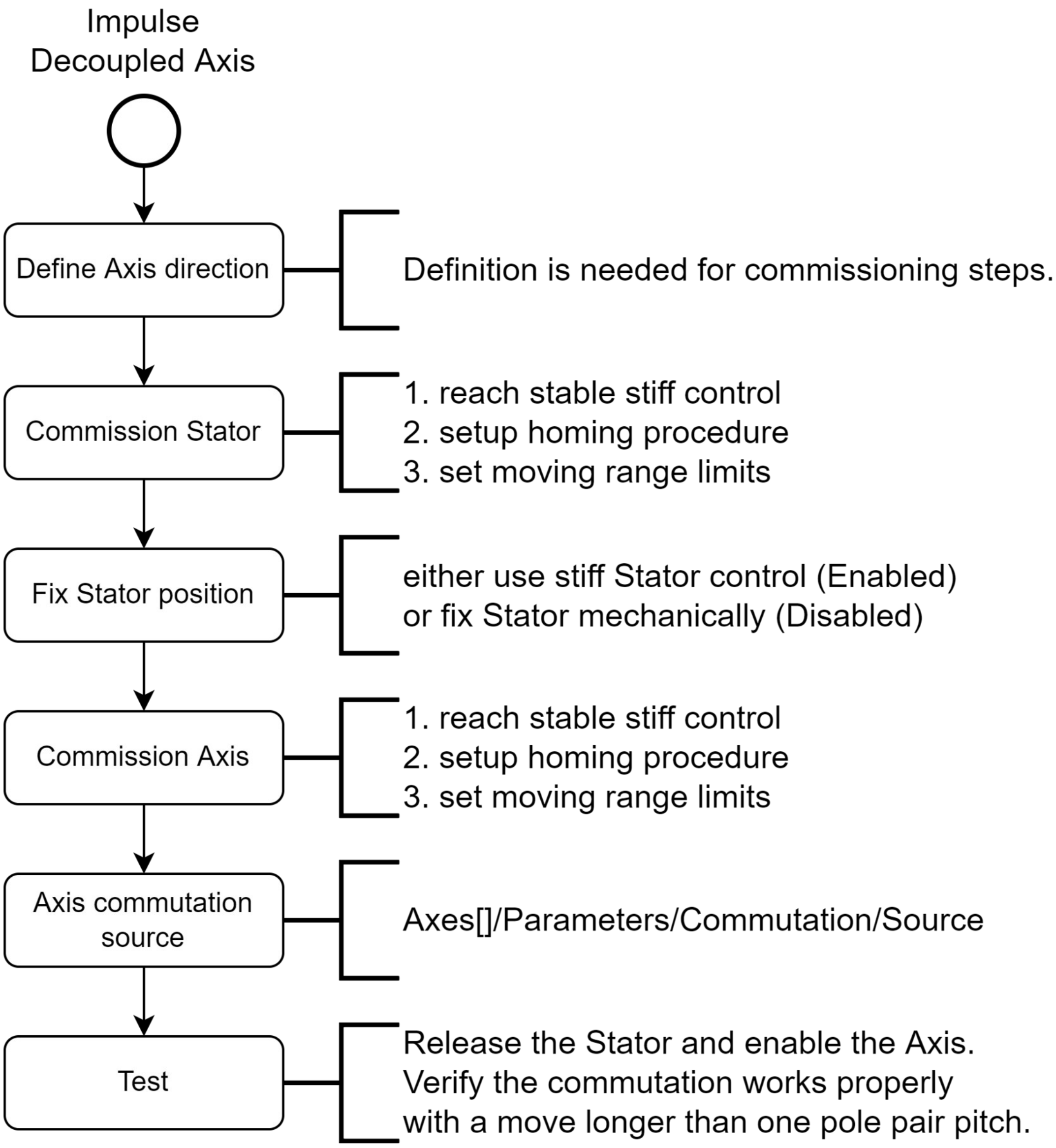


Figure 2: Commissioning flow chart

References

[1] "Encoder Configuration", AN107_Encoder_EP017.pdf, Triamec Motion AG, 2022

[2] "Servo Drive Setup Guide", ServoDrive-SetupGuide_EP016.pdf, Triamec Motion AG, 2022.

[3] "Homing Procedures and Setup", AN141_HomingProceduresAndSetup_EP002.pdf,

Triamec Motion AG, 2022

Revision History

Version	Date	Editor	Comment
001	2022-03-04	sm	Mini-release to communicate setup constraints
002	2022-05-17	sm	Commissioning description, Preconditions, Mode of Operation proposals
003	2022-10-28	dg	Tama sample code

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.