# Sensorless Motor Commissioning

***Application Note 133***

This document describes how to run synchronous motors without a feedback of the rotor position. Instead of a position signal a model of the electrical system is used to calculate the rotor angle.

## Table of Contents

Document AN133_SensorlessCommissioning_UserGuide_EP
Version 008, 2023-12-12
Source Q:\Doc\ApplicationNotes\
Destination T:\Doc\ApplicationNotes
Owner dg

# 1 Sensorless Mode

With the TSD servo-drives it is possible to run synchronous motors without a feedback of the rotor position. Instead of a position signal a model is used to calculate the rotor angle based on the applied voltage and the measured current. As the model requires a certain velocity of the rotor to estimate the position of the rotor reliably, the motor is driven in synchronous mode below a certain velocity and in sensorless mode above that velocity.

In synchronous mode a constant current-phasor is rotated to drive the motor. When a certain velocity threshold is reached (SensorlessTransitionStartVelocity) the transition to sensorless mode starts. The position of the motor and the commutation angle are now calculated based on the model, the position controller is activated and the current-phasor is reduced steadily until SensorlessTransitionEndVelocity is reached and the motor is controlled in sensorless mode (see Figure 1).

This document describes a procedure to commission a motor in sensorless mode. Depending on the characteristic of the motor, the sequence has to be adapted to get good results.

Chapter 2 describes the preparation of the TAM configuration. Chapter 3 describes the commissioning process in detail and how the main parameters can be derived.

**Remark** This version of the document prerequisites a TAM System Explorer newer or equal to version 7.24 and a firmware version newer or equal to FW4.10.6.

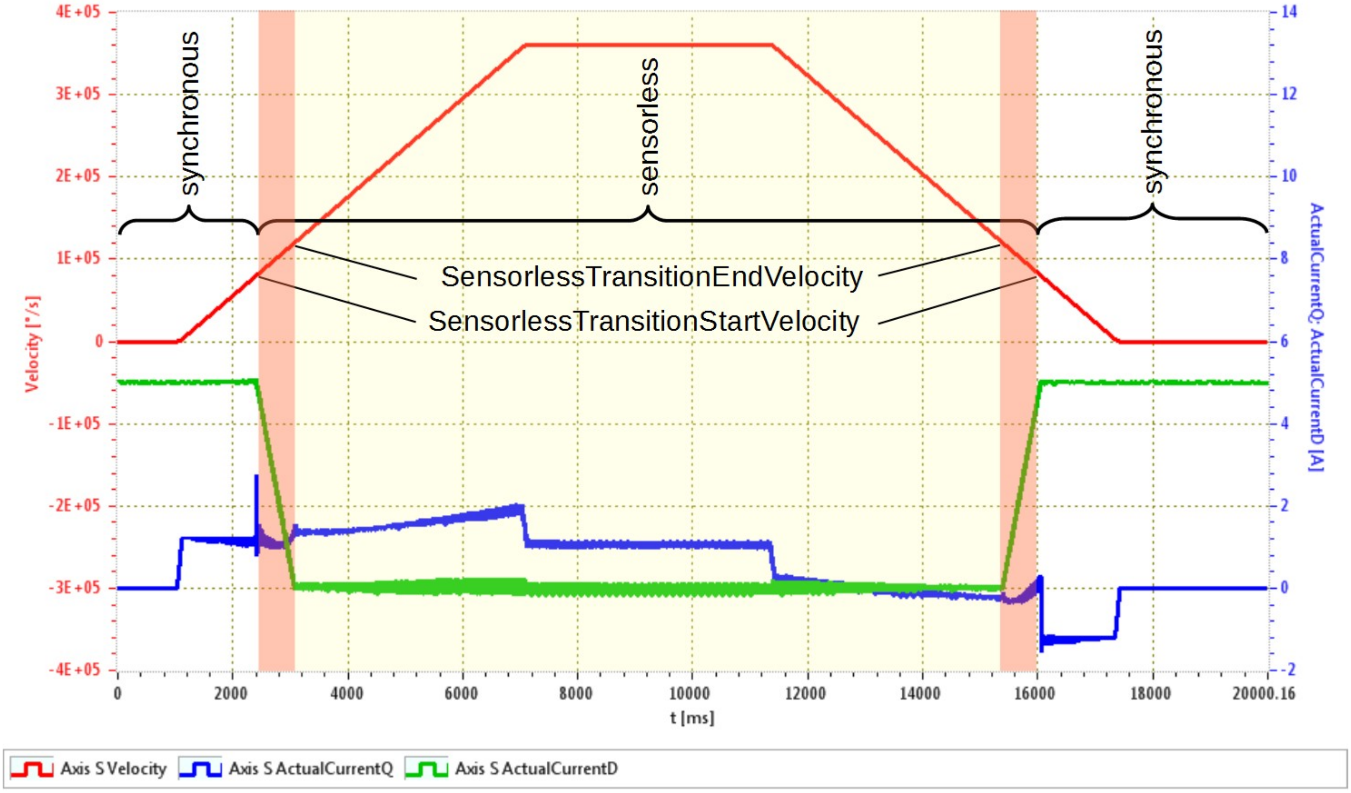


*Figure 1: Full speed move with transition form synchronous operation to sensorless and back to synchronous.*

# 2 TAM Configuration Preparation

This section describes the preparation of the TAM configuration to run a motor in sensorless mode.

## 2.1 Required Data

The following data is required for the commissioning of the motor (see data sheet of the motor):

- Peak current *[$A_{PK}$]*[1]
- Nominal current *[$A_{RMS}$]*[1]
- Thermal time constant *[s]*
- Pole pairs *[]* (if number of poles is given in the data-sheet: *PolePairs = NuberOfPoles/2*)
- Torque constant *[Nm/$A_{PK}$]*[1]
- Inertia *[kg $m^2$]*
- Resistance phase-phase *[Ohm]*
- Inductance phase-phase *[H]*
- Type of temperature sensor (if a temperature signal is available)
- Temperature limit (if a temperature signal is available)

To verify if the required dynamic is reachable the following specifications are required:

- Specified max velocity
- Specified max acceleration
- Specified max jerk

## 2.2 TAM Parameters

This section gives a recommendation for the setup of the TAM parameters. Depending on the characteristic of the motor, modifications are required. Default values are used for TAM parameters not mentioned in this section.

First the desired position unit ([rad], [degree] or [turns]) should be configured in Axes[].Parameters.PositionController.PositionUnit. The configured PositionUnit has to be considered when configuring parameters with a relation to position and their value must be converted accordingly.

### General.Parameters

In General.Parameters the voltage limits needs to be adjusted according the to the DC bus voltage.

- DcBusVoltageUpperLimit: This parameter defines the upper voltage limit. It has to be considered, that during deceleration recuperation can cause a rise the DC bus voltage. Therefore the power supply should be equipped with a limitation of the DC bus voltage (e.g. braking resistor). The value of DcBusVoltageUpperLimit should be set slightly above the voltage limitation of the power supply.
- DcBusVoltageLowerLimit: It is recommended to set the lower limit 10 to 20 % below the nominal value of the DC bus voltage.

If the measured DC bus voltage is out of the specified range, a DcBusVoltageOutOfRange error is issued.

[1] Verify if the data sheet expresses the current as a peak value *[$A_{PK}$]* or as effective or root mean square value *[$A_{RMS}$]*. A current value given in $A_{RMS}$ can be transformed to $A_{PK}$ by multiplying it with the square root of two. If $A_{RMS}$ is in the denominator (e.g. torque constant), the value has to be divided by the square root of two.

### Axis[].Parameter.Motor

This parameters are used to specify the properties of the motor.

- Type: Set this parameter to SynchronousAC
- InvertDirection: Set this parameter to True if the motor turns in the wrong direction.
- PolePairs: Set this parameter according to the data-sheet of the motor
- EncoderCountsPerMotorRevolution: Set this parameter to 1000
- PeakCurrent: Set this value according to the data-sheet of the motor in unit *[$A_{PK}$]*.[1]
- NominalCurrent: Set this value according the data-sheet of the motor in unit *[$A_{RMS}$]*.[1]
- CurrentSquareTime: This value corresponds to the thermal time constant of the winding of the motor. Set this value according the data-sheet of the motor.
- TemperatureSensorType: Sensor type if sensor is available – else None
- TemperatureUpperLimit: Set this parameter according the data-sheet of the motor.

**Warning** The limitation of the current (NominalCurrent, PeakCurrent, CurrentSquareTime) must be configured correctly before the drive is enabled the first time. Wrong values could cause damage of the motor.
It is recommended to use a temperature sensor to observe the motor temperature.

### Axis[].Parameters.PathPlanner

Set the following dynamic limits according to the specifications of the axis and consider the unit configured in PositionController.PositionUnit.

- VelocityMaximum
- AccelerationMaximum, DecelerationMaximum, DecelerationEmergency
- JerkMaximum

Consider that the reachable dynamic is also limited by the available DC bus voltage and the restriction of the current (motor or drive) and the physical properties of the motor and the load.

For motors it is common to use modulo. The modulo range is set in axis units. One turn in radians is set in the following example.

- ModuloPositionMaximum = 6.2831853071795862
- ModuloPositionMinimum = 0.0

Make sure, that the modulo range does match the modulo range used by the external control system. Consider that per modulo range at least two position samples must be provided by the control system. Therefore the following condition has to be fulfilled.

$$(\text{ModuloPositionMaximum} - \text{ModuloPositionMinimum}) > v_{max} T_s \tag{1}$$

With $v_{max}$ the maximum speed of the motor and $T_s$ the sampling time of the position set-points provided by the control system. It is maybe required to increase the modulo range by a multiple of one turn to fulfill equation (1).

### Axis[].Parameter.PositionController

The tuning of the position controller and the feed forward is described in chapter 3. This section only describes the initial setting.

- PositionUnit: Select the desired position unit. Consider that the values of all parameters related to position must match with the selection.
- FeedForwardPosition = 0.
- FeedForwardVelocity, FeedForwardCoulombCurrent, FeedForwardCoulombVelocity: Initially set this values to zero. See chapter Feed Forward on page 13 for the final setup.
- FeedForwardAcceleration: As an initial guess, the acceleration feed forward can be calculated based on the inertia *J* and the torque constant $K_t$ of the motor according to equation (2). The selected PositionUnit has to be considered for the calculation. If *J* is given in *[kg m²]* and *Kt* in *[Nm/A]* then the result of equation (2) has to be multiplied with the following factors:
  - [rad]: 1
  - [turn]: 2π
  - [degree]: 2π/360
- See section 'Feed Forward' on page 13 for how to verify and optimize the feed forward setup.

$$\text{FeedForwardAcceleration} = \frac{J}{K_t} \tag{2}$$

- OutputLimit: Set this parameter to 0.95 times value of the Motor.PeakCurrent parameter.
- Encoders[0].Type: Select Sensorless.
- Encoders[0].Pitch: The value for the Pitch depends on the value of Motor.EncoderCountsPerMotorRevolution and on the PositionUnit setting.
  - [rad]: Pitch = 2π/EncoderCountsPerMotorRevolution
  - [turn]: Pitch = 1/EncoderCountsPerMotorRevolution
  - [degree]: Pitch = 360°/EncoderCountsPerMotorRevolution
- Controllers[0]: For the initial setup of Kp, Ki, Kd, T1 and Filter[0] see section 'Position Controller' on page 11.
- Controllers[0].PositonErrorLimit: Set this value high enough to not cause an error in normal operation but also small enough to detect a malfunction of the motor. For an initial setup it is recommended to set the PositionErrorLimit to a value which corresponds to about two turns of the motor. Adjust the value based on a measurement of the PositonError after commissioning.
- Controllers[0].IntegratorOutputLimit = 0.9 * Motor.PeakCurrent.

### Axis[].Parameters.Commutation

- PhasingMethod= RotorAlignment
- EnablingMethod= ForcePhasing
- StartTime = 0s
- RampRiseTime = 0.5s
- RampConstTime = 0.5s
- CurrentAmplitude = 0.5...0.75 * Motor.NominalCurrent.
- SineAmplitude = 0.1rad
- SineFrequency = 10Hz

**Note** The current defined in register CurrentAmplitude is also used for the current phasor to run the motor in synchronous mode.

### Axis[].Parameters.CurrentController

- PwmFrequency: For motors with low inductance it is recommended to set this parameter to pwm100kHz. This reduces the current ripple and therefore the heating of the motor. Consider that 100kHz PWM reduces the peak limit and nominal current limit of the drive compared to 50kHz PWM. If high current is required, it might be reasonable to set PwmFrequency to pwm50kHz.
- IntegratorOutputLimit = 0.57 * nominal DC-Bus voltage.
- Kt = 0.
- OutputLimit = 0.57 * nominal DC-Bus voltage.
- SensorlessBandpassDamping: This parameter is used to adjust the filter of the sensorless model. An initial value of 0.5 is recommended.

# 3 Commissioning

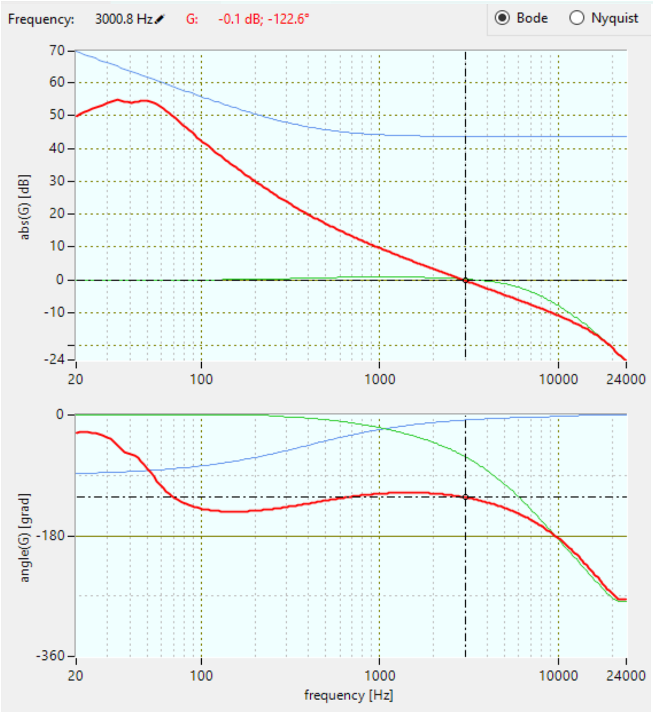
The following section describes one possible procedure to commission a sensorless motor. Depending on the characteristic of the motor, the sequence has to be adapted to get good results. In a first step the setup of the synchronous mode is explained and in a second step the setup of the sensorless mode.

## 3.1 Setup of Synchronous Mode

This section describes the required steps to operate the motor in synchronous mode.

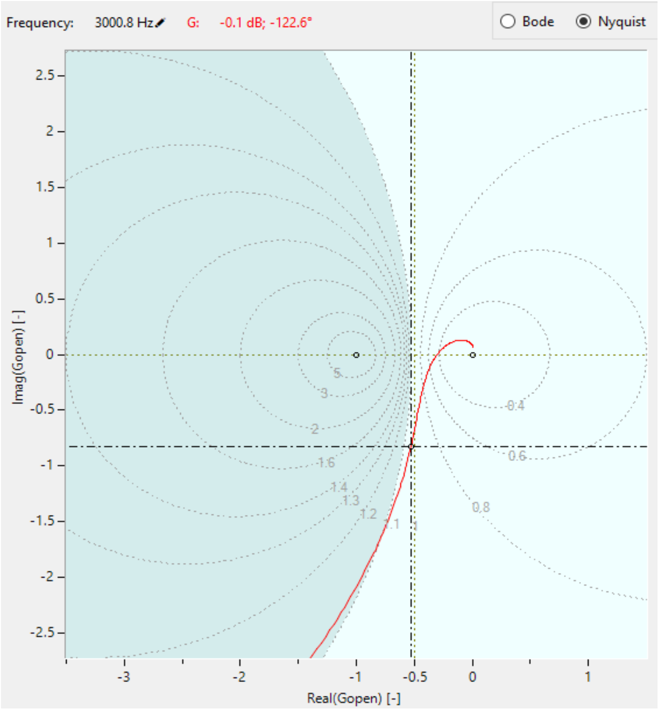
### Controller Parameters $K_R$ and $T_n$

The tuning of the current controller with the parameters $K_R$ and $T_n$ can be done with the frequency response tool based on a frequency response measurement. The current controller should be stiff, with high bandwidth but without overshoot. See figure 2 and 3 for an example of the Bode and Nyquist plot after the tuning of the current controller. See also [1] for how to do the frequency response measurement and the tuning of the current controller.



Frequency: 3000.8 Hz G: -0.1 dB; -122.6° ● Bode ○ Nyquist

*Figure 2: Bode-plot of current controller.*



Frequency: 3000.8 Hz G: -0.1 dB; -122.6° ○ Bode ● Nyquist

*Figure 3: Nyquist-plot of current controller*

### Transition Velocities

These velocities will later be used to define the transition from synchronous mode to sensorless mode (see section 3.2). As the following setup is used for synchronous operation, the transition is disabled by setting:

- CurrentController.SensorlessTransitionStartVelocity = PathPlanner.VelocityMaximum
- CurrentController.SensorlessTransitionEndVelocity = PathPlanner.VelocityMaximum

### Turning the Motor in Synchronous Mode

If all the parameters are set as described above, it should now be possible to enable and to run the axis in synchronous mode by using the Axis Module in TAM System Explorer.

## 3.2 Setup of Sensorless Mode

In sensorless mode, the position of the motor is estimated based on the induced voltage. This estimation is done based on a model of the motor which requires the parameterization of the resistance and the inductance. Additionally, this section describes the setup of the position controller.

**Remark** To scope signals from register Signals.CurrentController.Sensorless a sampling time of 0.02ms or 0.01ms is required. The values will be displayed as zero in scope if the sampling time is 0.1ms or bigger.

### Resistance R

As an initial value the phase resistance (0.5 x phase to phase resistance) can be measured with a multi meter. Also the resistance of the cabling (and of the filter if used) needs to be considered. A more precise option to measure and verify the resistance R is by running the motor at very low velocity in synchronous mode and measure the ratio between voltage and current:

1. Prepare the scope by loading the *Scope Configuration* named **Sensorless Motor-R Analysis**.
2. Run the motor with low velocity in synchronous mode and measure the signals for at least one electrical turn. The velocity should be low enough so that the voltage and current are in phase (see Figure 4).
3. Measure the ratio between VoltageAlpha and CurrentAlpha or VoltageBeta and CurrentBeta to determine the resistance (Figure 4). And write the result to register Parameters.CurrentController.R.

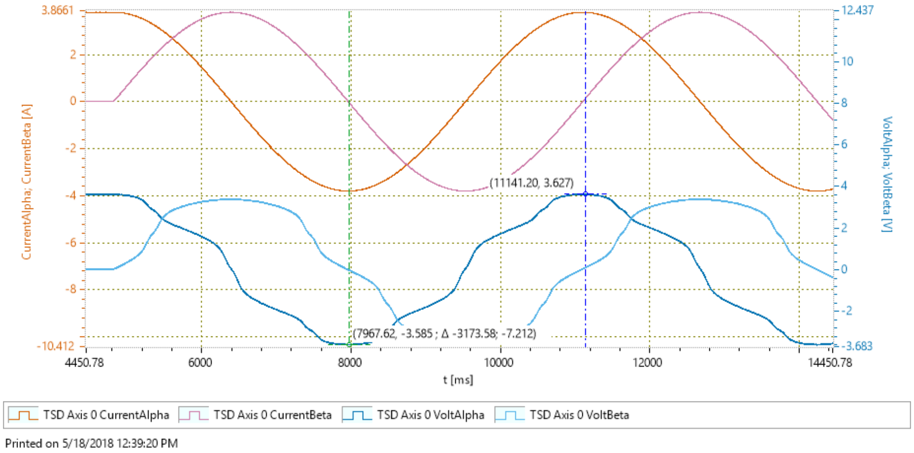


*Figure 4: Measurement used to evaluate the resistance. R= 3.6V/3.8A = 0.85Ohm.*

### Inductance L

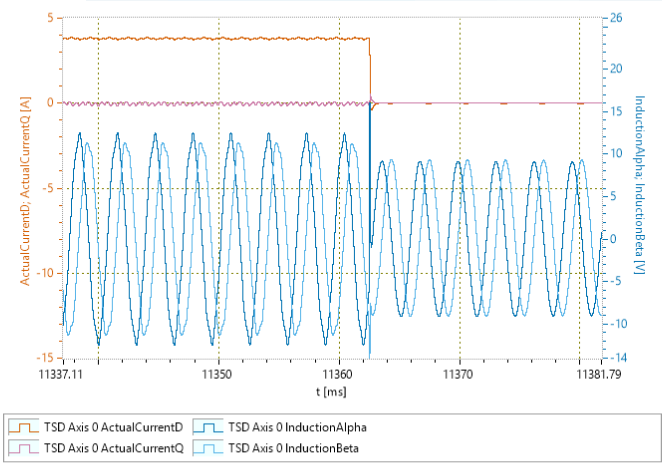
As an initial value, the phase inductance value (0.5 x phase to phase inductance) from the data sheet can be used for the parameter Parameters.CurrentController.L. If an additional filter is used, also the inductance of the filter has to be considered. It is recommended to verify and adjust the parameter with the following measurement sequence:

1. Prepare the scope by loading the *Scope Configuration* named **Sensorless Tuning**.
2. For this measurement set the following parameters temporarily to zero:
    - PositionController.FeedForwardVelocity
    - PositionController.FeedForwardCoulombCurrent
3. Run the motor at a certain speed (e.g. *0.5 vMax*) in synchronous mode. Maybe this requires to temporary set the following parameters to a higher value:
    - CurrentController.SensorlessTransitionStartVelocity
    - CurrentController.SensorlessTransitionEndVelocity
4. Start the scope.
5. Now the parameters are changed in such a way that the motor is in free-run. To achieve this, set the value of Commutation.CurrentAmplitude to zero temporarily. This causes the controller to regulate the current to zero.
6. Now the values of InductionAlpha and InductionBeta can be compared at the transition from synchronous mode to free-run. If the parameter L is set correctly, the amplitudes and phases of both signals should be continuous at the transition (without a step). If the amplitude gets smaller abruptly at the transition to free-run, L has to be increased and vice versa (see Figure 5 and 6).
7. Repeat from step 3 until the condition at 6 is fulfilled and the optimal value $L_{opt}$ is found. Don't forget to stop the running move and to reset Commutation.CurrentAmplitude to the initial value before starting the next one*.*
8. Write the resulting inductance to register *Parameters.CurrentController.L*

**Note** In most cases most robust results were achieved, by setting the inductance slightly smaller than the found $L_{opt}$*: L = 0.75...0.95 ** $L_{opt}$

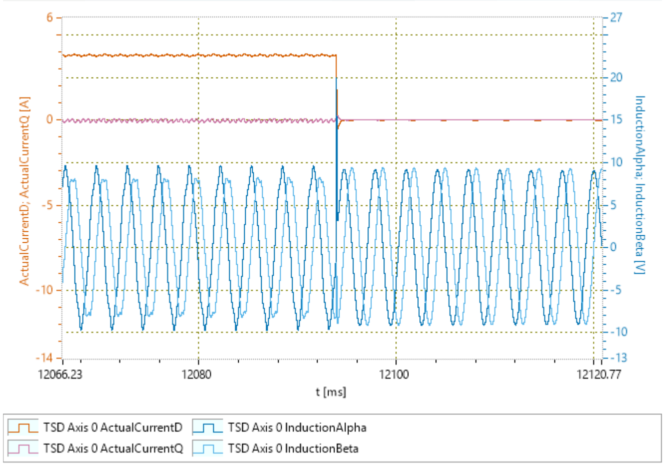
Don't forget to reset the following parameters after the optimization of resistance R and Induction L:

- CurrentController.SensorlessTransitionStartVelocity
- CurrentController.SensorlessTransitionEndVelocity
- Commutation.CurrentAmplitude
- PositionController.FeedForwardVelocity
- PositionController.FeedForwardCoulombCurrent



Printed on 5/18/2018 2:00:37 PM

*Figure 5: Estimated induced voltage before optimization of induction L (L=0).*



Printed on 5/18/2018 2:04:01 PM

*Figure 6: Estimated induced voltage after optimization of induction L*

## Transition Velocities

The transition velocities define the transition from synchronous to sensorless mode (see Figure 1). The transition velocities are defined with the following parameters:

- CurrentController.SensorlessTransitionStartVelocity
- CurrentController.SensorlessTransitionEndVelocity

The bandwidth of the output filter (see section 'Controller Output Filter' on page 11) should be lower than the first harmonic oscillation of the induction model. With the following rule of thump the first harmonic oscillation at the transition velocity is twice the bandwidth of the output filter:

$$v_{SL\,Start} \geq \frac{4\pi f_{0\,PF}}{p}; \qquad v_{SL\,End} = 1.1\,v_{SL\,Start} \tag{3}$$

whereas

- $v_{SL\,Start}$ is SensorlessTransitionStartVelocity [rad/s],
- $v_{SL\,End}$ is SensorlessTransitionEndVelocity [rad/s],
- $p$ is the number of pole pairs (Parameters.Motor.PolePairs),
- $f_{0\,PF}$ is the bandwidth of the position controller output filter (recommended: 100Hz).

The selected PositionUnit has to be considered for the calculation. Depending on the selected PositionUnit the result of equation (3) has to be multiplied with the following factors:

| | |
|---|---|
| [rad]: | 1 |
| [turn]: | 1/2π |
| [degree]: | 360/2π |

For example with $f_{0\,PF}$ = 100Hz and $p = 1$ the velocities are set to

- SensorlessTransitionStartVelocity = 1250rad/s
- SensorlessTransitionEndVelocity = 1380rad/s
- Parameters.CurrentController.SensorlessTransitionStartVelocity = 1250rad/s

- Parameters.CurrentController.SensorlessTransitionEndVelocity = 1380rad/s

This parameters may need to be adjusted in case the bandwidth of the output filter has changed.

### Position Controller

As an initial guess the controller parameters are derived based on a physical model (see 5.1). For this, the inertia of the rotor *J* with unit *[kg m²]* and the torque constant $K_t$ with unit *[Nm/A]* must be known.

$$K_P=(2\pi f_{0PC})^2\frac{J}{K_t}; \qquad K_I=\frac{(2\pi f_{0PC})^3}{10}\frac{J}{K_t}; \qquad K_D=4D\pi f_{0PC}\frac{J}{K_t}; \tag{4}$$

Depending on the selected PositionUnit the result of equation (4) has to be multiplied with the following factors:

- [rad]: 1
- [turn]: 2π
- [degree]: 2π/360

As a first estimate a bandwidth of $f_{0\,PC}$ = *10Hz* and a damping of *D* = *0.7* are recommended. With this and depending on the position unit the initial parameters can be calculated as follows:

| | | | | |
|---|---|---|---|---|
| *[rad]* | $K_P\approx 4000\frac{1}{s^2}\frac{J}{K_t}$; | $K_I\approx 25000\frac{1}{s^3}\frac{J}{K_t}$; | $K_D\approx 90\frac{1}{s}\frac{J}{K_t}$; | $T_1\approx 0.0001\,s$ |
| *[turn]* | $K_P\approx 25000\frac{1}{s^2}\frac{J}{K_t}$; | $K_I\approx 155000\frac{1}{s^3}\frac{J}{K_t}$; | $K_D\approx 550\frac{1}{s}\frac{J}{K_t}$; | $T_1\approx 0.0001\,s$ |
| *[degree]* | $K_P\approx 70\frac{1}{s^2}\frac{J}{K_t}$; | $K_I\approx 400\frac{1}{s^3}\frac{J}{K_t}$; | $K_D\approx 1.6\frac{1}{s}\frac{J}{K_t}$; | $T_1\approx 0.0001\,s$ |

### Controller Output Filter

To avoid that the position controller disturbs the sensorless model, a low pass filter of second order is applied to the controller output with following bandwidth and damping:

$$f_{0PF}\approx 10\cdot f_{0PC}\approx 100\,Hz; \qquad D=0.7; \tag{5}$$

Consider also the dependency with the SensorlessTransitionStartVelocity (see equation 3).

The configuration of the filter is done by setting the following parameters:

- PositionController.Controllers[0].Filters[0].Type = Lowpass2
- PositionController.Controllers[0].Filters[0].EdgeFrequencyDenominator = 100 Hz
- PositionController.Controllers[0].Filters[0].DampingDenominator = 0.7

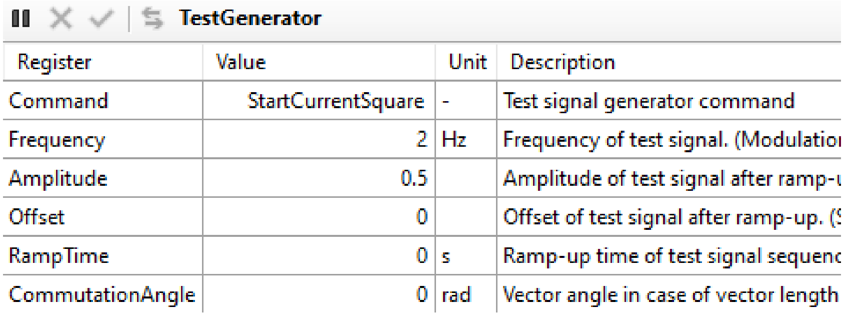
### Tuning of the Position Controller

To verify the stability of the position controller and to optimize the parameters, there are two procedures.

1. Do a frequency response measurement, following the instructions in chapter 3.3, and apply the usual tuning procedures as described in [1].
2. Tune the axis in time domain using a test signal while the motor is running fast enough to reach the sensorless mode.

Use the following procedure to tune in time domain:

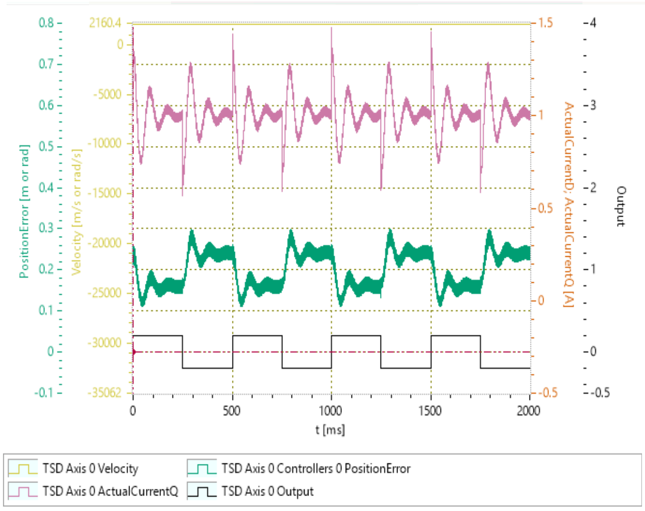
1. Prepare the scope by loading the *Scope Configuration* named **Position Controller Disturbance Tuning**.
2. Run the motor fast enough to reach the sensorless mode (velocity > CurrentController.SensorlessTransitionEndVelocity).
3. Apply a test-signal. Figure 7 shows a possible setup of the test-signal generator used for the optimization of the position controller (Commands.TestGenerator). The square signal is added to the current set-point as soon as the command is set to StartCurrentSquare. In case the axis fails when the test signal is applied, the amplitude of the test signal has to be reduced.
4. Now the transient response of the position error can be optimized by adjusting parameters Kp, Ki, Kd, T1. Figures 8 and 9 show the plots before and after the optimization.



TestGenerator

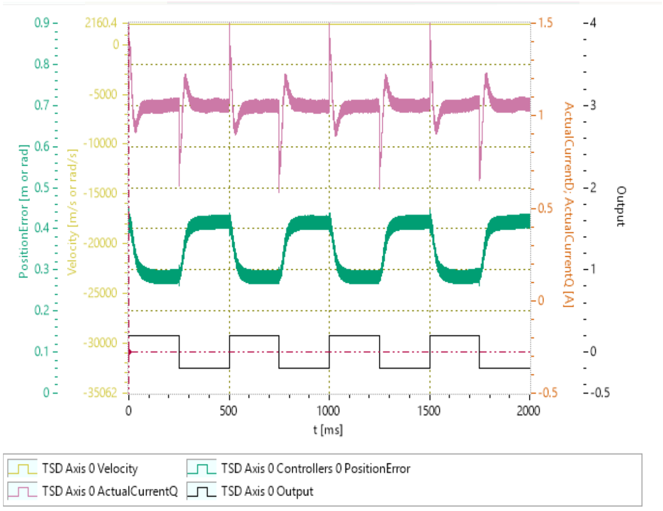
| Register | Value | Unit | Description |
|---|---|---|---|
| Command | StartCurrentSquare | - | Test signal generator command |
| Frequency | 2 | Hz | Frequency of test signal. (Modulatio |
| Amplitude | 0.5 | | Amplitude of test signal after ramp-u |
| Offset | 0 | | Offset of test signal after ramp-up. (S |
| RampTime | 0 | s | Ramp-up time of test signal sequenc |
| CommutationAngle | 0 | rad | Vector angle in case of vector length |

*Figure 7: Test-signal generator setup.*



Printed on 5/18/2018 2:40:29 PM

*Figure 8: Before optimization of the position controller.*



Printed on 5/18/2018 2:39:36 PM

*Figure 9: After the optimization of the position controller.*

Now it should be possible to operate the motor in sensorless mode.

## Feed Forward

This section describes how to configure and verify the feed forward parameters. To configure the velocity and coulomb feed forward the actual current has to be measured at two different velocities:

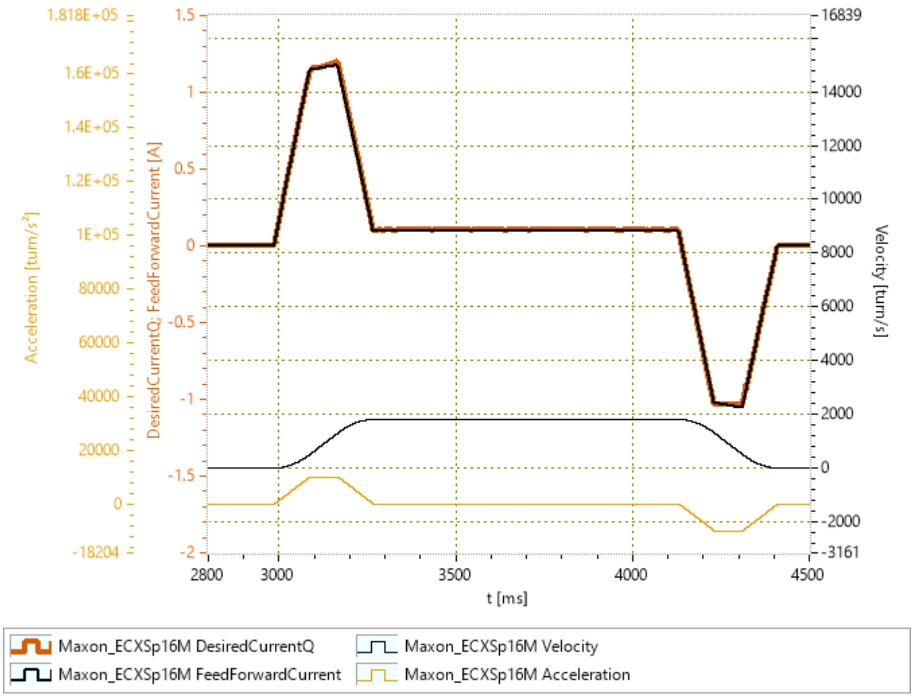
1. Prepare the scope by loading the *Scope Configuration* named **Sensorless Tuning**.
2. Run the motor at a velocity ($v_1$) in the lower end of the sensorless range and measure the average of the ActualCurrentQ ($i_{Q1}$).
3. Run the motor at the maximum velocity ($v_2$) and measure the average of the ActualCurrentQ ($i_{Q2}$).
4. With this measurement the velocity feed forward and the coulomb feed forward can be calculated:

$$v_{ff} = \frac{i_{Q2} - i_{Q1}}{v_2 - v_1}; \qquad c_{ff} = \frac{i_{Q1} v_2 - i_{Q2} v_1}{v_2 - v_1}; \tag{6}$$

- PositionController.FeedForwardVelocity = $v_{ff}$.
- PositionController.FeedForwardCoulombCurrent = $c_{ff}$.
- PositionController.FeedForwardCoulombVelocity = 0.1 · SensorlessTransitionStartVelocity.

To configure and verify the acceleration feed forward, the feed forward current is compared with the actual current:

1. Prepare the scope by loading the *Scope Configuration* named **Feed Forward Analysis**.
2. Run the scope and accelerate the motor to the maximum speed and decelerate the motor to zero afterwards. Adjust the parameter FeedForwardAcceleration to match FeedForwardCurrent as good as possible with ActualCurrentQ during acceleration and deceleration (Figure 10).



Printed on 12.10.2020 17:35:35

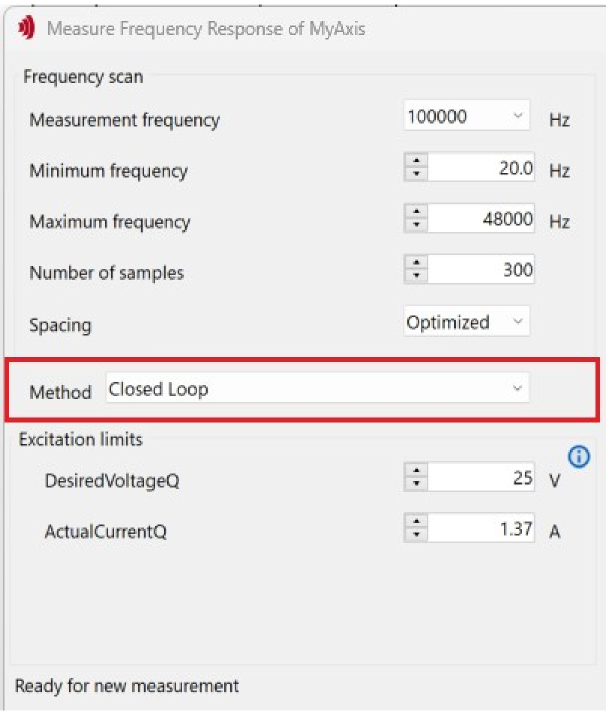
*Figure 10: Optimized FeedForwardCurrent to match ActualCurrentQ.*

## 3.3 Frequency Response Measurement

With TAM System Explorer 7.14 it becomes possible to do a Frequency Response measurement of the position loop based on the estimated position of the sensorless model. This can be useful to verify or further optimize the setup of the controller. This section describes how to execute the Frequency Response measurement.

The estimation of the position is only possible, if the motor is running at a velocity above the SensorlessTransitionEndVelocity in sensorless mode. Therefore the motor must be configured as described in the previous chapters. To do the measurement the following steps are required:

1. Enable the motor and run the motor with a velocity above SensorlessTransitionEndVelocity.
2. Start the Frequency Response Measurement Tool (see [1] for a description of the tool).
3. Set the Method to **Closed Loop** or **Closed Loop – Integrator off** (Figure 11) then configure the measurement settings.
4. If the settings are set to reasonable values, the measurement can be started. As the movement of the motor already requires some voltage, one has to make sure, that the desired voltage does not exceed the output limit of the current controller or the DC bus voltage. In case this is a problem, reduce the **DesiredVoltageQ** or the velocity of the motor.



Measure Frequency Response of MyAxis

Frequency scan

Measurement frequency 100000 Hz

Minimum frequency 20.0 Hz

Maximum frequency 48000 Hz

Number of samples 300

Spacing Optimized

Method Closed Loop

Excitation limits

DesiredVoltageQ 25 V

ActualCurrentQ 1.37 A

Ready for new measurement

*Figure 11: For sensorless the measurement method needs to be set to "Closed Loop".*

# 4 Trouble-Shooting

MotorPeakCurrentError during acceleration or deceleration:

- Reduce jerk and/or acceleration.
- Check stability of the position controller (see section 3.2).
- Check setting of acceleration feed forward (see section 3.2).
- In some cases more robust results where achieved with a reduced L.

MotorPeakCurrentError or PositionErrorLimit when high current is applied:

- In some cases it helps to slightly reduce SensorlessBandpassDamping.

DC bus voltage out of range during deceleration:

- During deceleration the energy of the motor is transferred back to the DC-bus which causes an increase of the DC bus voltage. Check if the power-supply is able to handle this increase of the voltage.
- Increase DcBusVoltageUpperLimit slightly above the voltage limitation of the power supply.
- Reduce the deceleration parameter.

Power supply shutdown during deceleration:

- Recuperation causes an increase of the DC bus voltage which causes a shutdown of the power supply.
- Use a power supply with limitation of the voltage (e.g. with brake resistor).
- Reduce parameter PathPlanner.DecelerationMaximum.

Motor runs in the wrong direction:

- Change value of Parameter.Motor.InvertDirection.

Motor temperature is too high*:*

- If the motor has a low inductance, the ripple-current caused by the PWM could lead to additional losses in the motor. The ripple could be analyzed by attaching a current probe to a motor-phase and comparing the signal-to-ripple ratio with an oscilloscope.
- Check Parameters.CurrentController.PwmFrequency is set to *pwm100kHz.*
- Add additional inductance with a pre-filter (see [https://www.triamec.com/de/sinus-filter-TF.html](https://www.triamec.com/de/sinus-filter-TF.html)).
- Check the stability of position and current controller.
- Check the cooling of the motor.
- Check the data sheet of to verify that the motor is operated within the specification.

Motor does not turn at start of move (synchronous mode):

- Check if Parameters.Commutation.CurrentAmplitude is configured correctly.
- Check the parameterization of the current controller.
- Check SensorlessTransitionStartVelocity and SensorlessTransitionEndVelocity.

Value of signals is displayed as zero in scope:

- Signals.CurrentController.Sensorless require a sampling rate of 50 kHz or 100kHz.
- Check if plot setting *Sampling Time* is set accordingly.

# 5 Appendix

## 5.1 Position Controller

With a PD-controller we get the following closed-loop transfer-function:

$$G_c = \frac{posAct}{posCmd} = \frac{K_t}{J} \frac{K_D s + K_P}{s^2 + \frac{K_t}{J} K_D s + \frac{K_t}{J} K_P}. \tag{7}$$

By comparing the coefficients of the denominator with a second order system with natural frequency $\omega_0$ and damping $D$ we get:

$$K_P = \omega_0^2 \frac{J}{K_t}; \qquad K_D = 2 D \omega_0 \frac{J}{K_t} \tag{8}$$

or with time constant representation of the controller:

$$K_R = \omega_0^2 \frac{J}{K_t}; \qquad T_v = 2 \frac{D}{\omega_0} \tag{9}$$

Additionally we set bandwidth limitation of the differentiation to

$$T_1 = \frac{T_v}{10} = \frac{D}{5 \omega_0}. \tag{10}$$

## 5.2 Sensorless Filter

The sensorless model uses a filter. This filter can be adjusted with the parameter CurrentController.SensorlessBandpassDamping. Figure 12 shows the effect of this parameter to the filter characteristic.

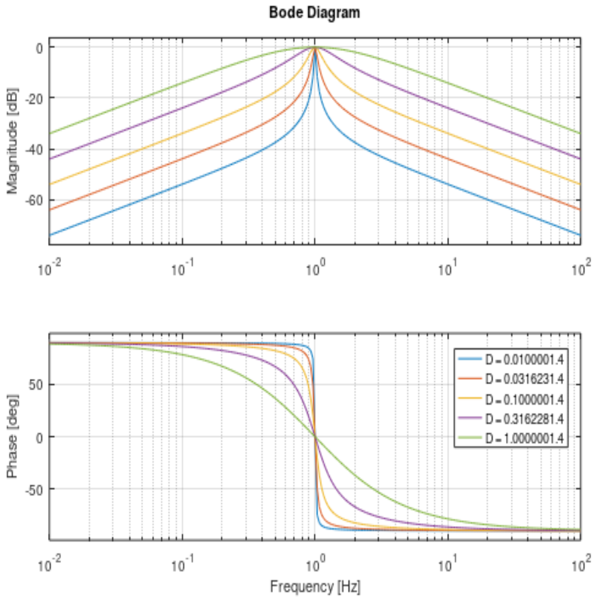


*Figure 12: Sensorless Filter Bode*

## References

[1] "Servo Drive Setup Guide, TSD and TSP Series",
ServoDrive-SetupGuide_EP023.pdf, Triamec Motion AG, 2023.

## Revision History

| Version | Date | Editor | Comment |
|---|---|---|---|
| 001 | 2018-05-09 | dg | initial draft |
| 002 | 2019-08-06 | dg | released version |
| 003 | 2020-08-21 | dg | Description of IntegratorOutputLimit modified and equation for controller parameters added if position unit is degree. |
| 004 | 2020-10-14 | dg | Revised and adapted to FW4.9.7 |
| 005 | 2021-01-05 | dg | Bode section adjusted. |
| 006 | 2023-06-28 | sm | update template and cleanup, use scope templates (TAM Software 7.24) |
| 007 | 2023-11-24 | ab | Review of entire document, made some corrections and adaptions |
| 008 | 2023-12-11 | ns | Replace Bode measurement/tuning with Frequency Response measurement/tuning |

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.