# Information
# for special configurations
# and control functions

Short Manual for proper analogue or digital torque control in combination with interesting ride, safety and other features for general and battery powered applications.

Version 1.03

Erstellt von: Dipl.-Ing. (FH) Edward Käding
E-Mail: ek@unitek.eu
Stand: 01.02.2023 – Version 1.03

# 1 Analog Torque Control

## 1.1 Analog Torque Control in general

In traction applications it is preferred to use the electrical drive with torque control. This is a similar technic to the use of throttle position in combustion engines.

The setting inside “Command Mode” is used to configure the input command logic:

Analog Torque: Command via Analog Input → Torque-Control

⇨ Set Command Mode to “Analog Torque”

The internal ADC measurement has got a 12 Bit resolution for the ±10 V input range.
The signal is then left-aligned creating a 16 Bit input range raw value of ±32767.

The final usage of the analogue input is the calculated (Ain x scaled) value which is defined by the settings of ‘Offset’, ‘Cutoff’ and ‘Scale’. The ‘Cutoff’ defines a range (dead zone) where Ain x scaled is set to 0.
The calculation is defined by the formula: **Ain x scaled = (Ain x in + Offset) * Scale**

⇨ The Ain x scaled value is then the torque command for the inverter.
This is of course a current command which is proportional to the maximum inverter current (±32767 == ±I max pk at 100 %).

The ‘Format’ defines the command logic of the analogue input. While (+Cmd) will command a positive torque (speed) value in case of a positive Ain x scaled value, the (-Cmd) Format will command a negative torque (speed) value.

The ‘Mode’ selection will define which raw bipolar voltage input will be used. While a Mode select of [-10..+10V] will use even a negative voltage input, the Mode select of [0..+10 V] will only use the positive voltage input.

In case you only have a positive potentiometer supply, but require a bidirectional speed command (e.g., boat application), it is possible to modify the analogue input by setting a large offset, a cutoff in between, and adjusting of scale to cover a full command range. Bit 12 in ID-Address: 0xDC (0x10xx), can serve as enable (RUN) only possible with Analog Command < Cutoff (torque = 0).

Important to understand:
The parameter Nmax100% (ID: 0xC8) represents the resolution of the 16 Bit (±32767) actual speed value in relationship to the physical speed calculation. Always set this parameter to a rotation speed value above the motor nominal rotation speed.

By adjusting the setting of parameter “N-Lim” (ID: 0x34) an activation for speed limitation in torque control mode can be activated. → “Torque-cruse-control”

N-Lim = 100 %, no speed limitation
N-Lim < 100 %, speed is limited to Nmax100% * N-Lim (Torque-cruse-control activated)

With the usage of limited speed, the speed controller is active within the range of limitation. This means the parameters of the speed controller and the noise of speed actual signal are important in this operation range.

## 1.2 Analog Torque Control with automatic recuperation logic (Brake Car)

The theory of recuperative braking is actually very simple. As soon as the motor phase current (I_actual) and the motor phase voltage (SpeedActual) are of opposite sign, then recuperative energy will go back to the battery. This means that all you must do is to send a negative torque command for generator torque, and you will be recuperating current back to the battery ($P_{DC} \leftrightarrow P_{AC}$).

Using digital torque command, the user has complete control whether the demanded torque will be a driving or a generator torque (recuperative braking) just by sending the desired torque value (M set (dig.)) using one of the digital communication interfaces (CAN, RS232).

Using analogue torque control the command logic for recuperation is more complicated.
Our software provides a function to let the inverter calculate a recuperative braking logic internally just by triggering the activation depending on the settings.

An analogue input is normally derived from an analogue throttle (potentiometer). The angle of the throttle is usually limited, the supply is normally unipolar and the zero position may be out of working range.

Diagram 1 gives an example of at which the working range of a throttle is between +1 V and 5 V.

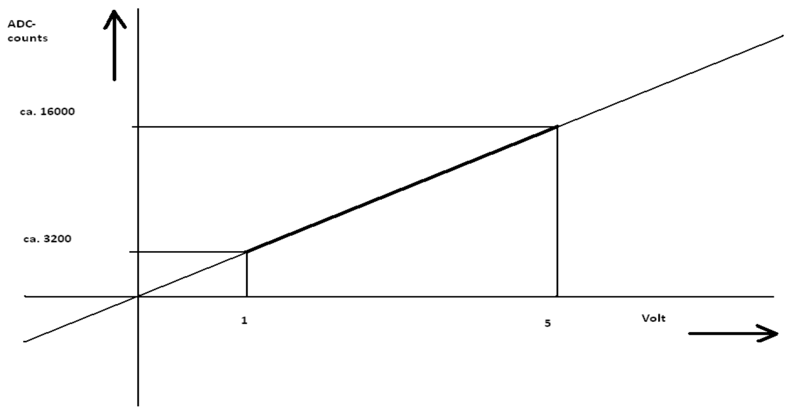



Diagram 1: Raw input signal → Ain x

Diagram 2 shows the modified analogue input by applying an offset of -4200, a cutoff of 400, and a scale factor of 2.500.

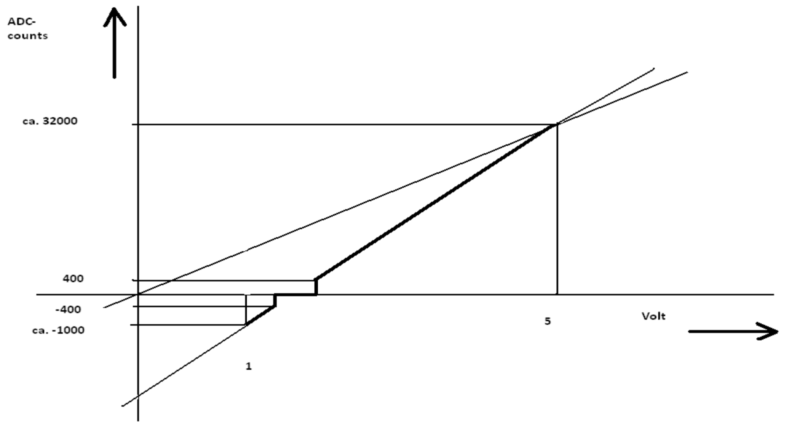



Diagram 2: Modified input signal → Ain x scaled

The positive values represent the range for driving torque. The scale factor is configured to reach the maximum possible output value. The offset creates a negative value range which can be interpreted as the range where the internal recuperation logic will be triggered.
The result will be a vehicle with only one direction for moving. Backward movement has either to be performed manually or with the usage of a gear box. A more recent Firmware (≥ 471) makes it possible to change the sign of torque-command by triggering the input configuration “N cmd Reverse” (page Logic in NDrive).

**Enable** special function “recuperative braking logic” version 1:
→ set one digital input to the function “Brake Car”
→ **and** set Mode select to [-10..+10V] of the analogue input of the used throttle

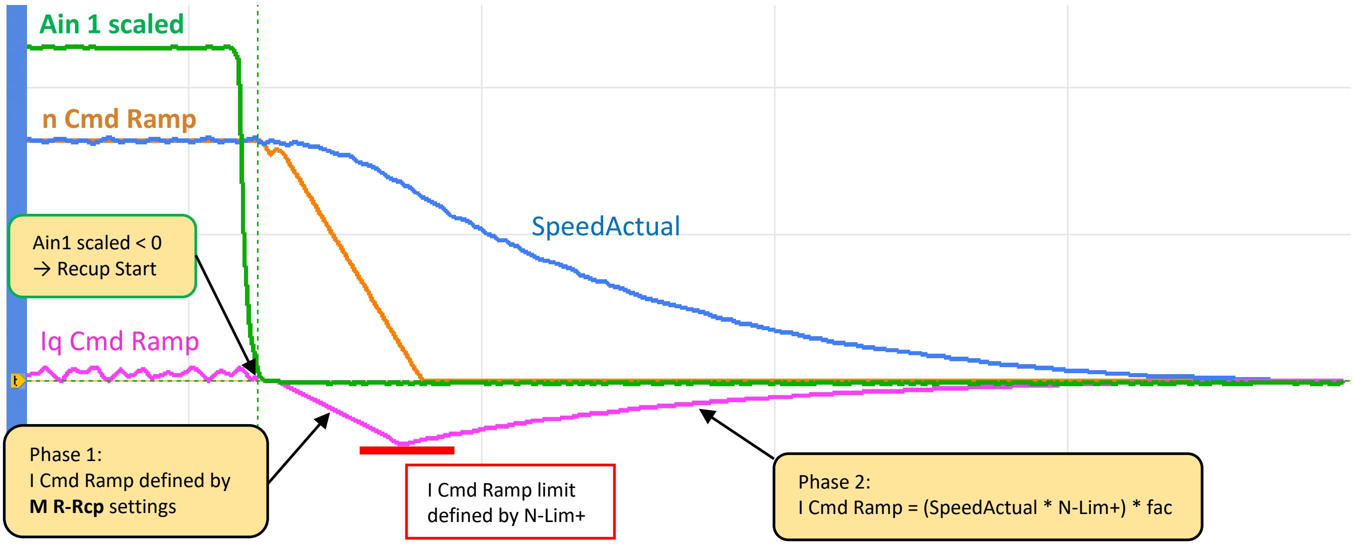
**Trigger** of the special function “recuperative braking logic” (2 logics):
→ Logic 1: The Ain x scaled value is < 0
→ Logic 2: The configured input (e.g. Din1 = “Brake Car [Ax]” or “Brake Car #2 [Ax]”) is triggered

**Calculation** logic for the braking torque command:
→ Logic 1: Torque = (-SpeedActual * N-Lim+) * fac (using parameter “N-Lim+” (ID: 0x3F))
→ Logic 2: Torque = ( SpeedActual * N-Lim- ) * fac (using parameter “N-Lim-“ (ID: 0x3E))

Note: ‘N-Lim+’ and ‘N-Lim-’ represent the maximum possible current in % of I max peak = 100 %.
In case of a long “N R-Dec” Ramp, different responses will apply.
In case the parameter fac > 1, this will enhance the maximum possible current.

The idea of Logic 1 is that if the throttle pedal is not pressed, the internal recuperation braking command will be activated.
The idea of Logic 2 is to connect the brake pedal to a switch (Digital Input) which will then activate the internal recuperation braking command. Meaning that different from Logic 1 the vehicle will be sailing if the throttle pedal is not pressed.

**Ramp generator** for all analogue torque command values:
- M R-Acc → rate of increase of driving torque.
- M R-Dec → rate of decrease of driving torque.
- M R-Rcp → rate of increase of recuperative braking torque if either Logic 1 or 2 is activated.




**Disable** the “recuperative braking logic”: (without settings change)
Set one digital input to “recu-disab [Ax]”. If this input is triggered, it will force any recuperation torque to zero (e.g., in case of a full battery).

## 1.3 Analog Torque Control with automatic recuperation logic (Brake Car #2)

This calculation of the automatic recuperation logic is practically the same as in using the “Brake Car” setting except with one exception. This calculation logic allows the driver to modify the maximum possible torque command using the throttle position.

As shown in chapter 1.2 by modifying the analogue input value of the potentiometer with the offset settings it is possible to create a negative analogue input value. While for the “Brake Car” logic only a negative value is relevant, the “Brake Car #2” logic uses the percentage of the negative value in relationship to its maximum possible negative value to define the maximum possible recuperative braking torque. Meaning that if your throttle is only at 50 % of its possible negative position, the braking torque will also only be 50 % of the possible “N-Lim+” definition. If the negative throttle position is at 100 %, the calculation will be the same as if the “Brake Car” logic would be used.

Diagram 3 shows the same modified input signal from Diagram 2 but also showing the possible negative throttle positions relevant for the following braking calculations.

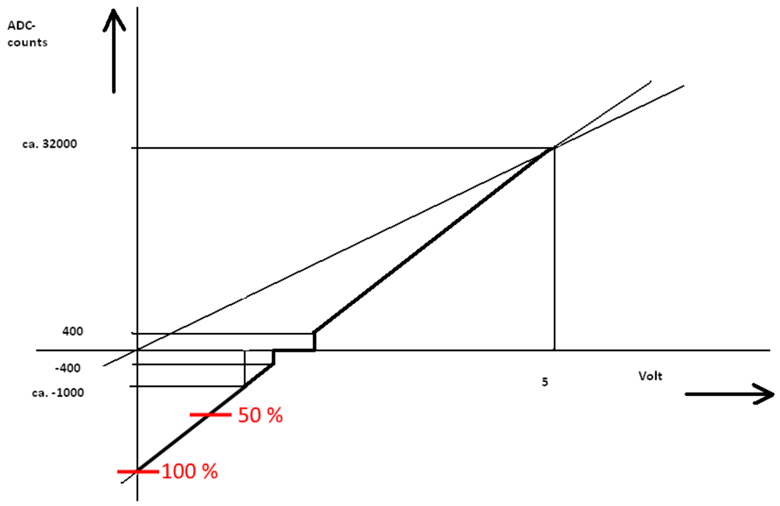



Diagram 3: modified input signal
→ Ain x scaled for „Brake Car #2”

**Enable** special function “recuperative braking logic” Version 2:
→ set one digital input to the function “Brake Car #2”
→ **and** set Mode select to [-10..+10V] of the analogue input of the used throttle.

**Trigger** of the special function “recuperative braking logic” (2 logics):
→ Logic 1: The Ain x scaled value is < 0
→ Logic 2: The configured input (e.g., Din1 = “Brake Car [Ax]” or “Brake Car #2 [Ax]”) is triggered

**Calculation** logic for the braking torque command:
→ Logic 1: Torque = -SpeedActual * N-Lim+ * (percentage of negative Ain x scaled value)
→ Logic 2: Torque = SpeedActual * N-Lim-

Note: The ramp generator calculation and the disable logic are the same as in chapter 1.2.

# 2 Digital Torque Control

## 2.1 Digital torque control in general

This chapter will only contain the digital torque control logics which can be relevant for car or battery powered applications.

Switching between Speed-Control and Torque-Control is done automatically inside the drive when “Command Mode” is set to “Dig. Commands”.

| | | | |
|---|---|---|---|
| Dig. Commands | Send request via CAN or RS232 to N set(dig.) ID: 0x31 | → | Speed-Control |
| | Send request via CAN or RS232 to M set(dig.) ID: 0x90 | → | Torque-Control |

The general Torque control M set(dig.) (ID: 0x90) is of course not a torque command. It is a current command which has been scaled to be able to send a defined current value to set a desired torque value defined by the motor specifications.

Note: Please read the “BAMOCR FAQ.pdf” manual on how to properly control a motors torque request.

# 3 ID Register 0xCD – Special Configurations and Control Functions Part 1

The following configurations and special functions of the ID Register 0xCD are valid for FW ≥ 480.

## 3.1 Configuration Bitmap – ID: 0xCD

| Symbol: | Function: | ID-Address: |
|---|---|---|
| | | **0xCD** |
| DC Current Sensor (Internal) | Enable the internal DC current sensor<br>(Only specific inverters) | Bit 0 |
| DC Current Limiting | Enable auto. DC Bus current limiting (Motor / Generator)<br>→ Requires DC Current Sensor | Bit 1 |
| DC Power Limiting | Enable auto. DC Bus power limitation (Motor / Generator)<br>→ Requires DC Current Sensor | Bit 2 |
| Enable with Torque Command | Always start inverter using torque command<br>→ Enable with free running torque | Bit 3 |
| Torque recuperation ramp | Activate the 'M R-Rcp' ramps settings option,<br>to control recuperation Acc torque ramps | Bit 4 |
| Reset SpeedRamp at n=0rpm* | Reset Speed Ramps if the actual Speed has a sign change | Bit 5 |
| Torque TimeOut after 100ms | Automatically sets Torque command to 0 if no new external torque command input within 100 ms is registered | Bit 6 |
| Auto. N-Lim Saturation | Auto. Saturation of the max allowed rpm speed depending on the measured Vdc-Bus and a fix constant factor value | Bit 7 |
| DC Current External Usage | Usage of the received external DC Current value for the auto. DC Bus current/power limitation (Motor / Generator) | Bit 8 |
| *rsvd_12..8* | … | Bit 12..9 |
| Auto. Brake Multipl. Factor | A Multiplication factor can be defined to the automatic recuperation logic for additional braking force | Bit 13..12 |
| Dig E Brake At0NmCmd | Activates the automatic recuperative braking logic, if the received torque command equals 0 (Mset(dig.) = 0) | Bit 14 |
| Dig. E-Brake Activation | Activates automatic recuperative braking logic | Bit 15 |
| | | |
| Enable Traction Control (TC)* | Enable internal Traction Control Logic (TC) | Bit 16 |
| TC- DeltaTime configuration* | TC delta time configuration. Time steps on when actual speed difference should be checked | Bit 18..17 |
| TC- Min. Speed configuration* | TC minimum speed configuration. Defines at what motor speed the TC should start operation | Bit 20..19 |
| TC- n Command Reset Option* | Reset Option for n command in case TC was triggered | Bit 21 |
| *rsvd_31..22* | … | Bit 31..22 |
| | | |
| * In development…<br><br>Note: Some functions require certain inverter types | | |

### 3.2 DC Current Sensor (Internal) – 0xCD Bit 0

The DC current sensor is only installed in specific battery inverters:

- Bamocar-D3-PG-700-160 Device Type: 0xC0
- Bamocar-D3-PG-700-100 Device Type: 0xC1
- Bamocar-D3-PG-700-900 Device Type: 0xC2

Activation condition:

| | | |
|---|---|---|
| 0xCD Bit 0 = 0 | → | DC Sensor disabled |
| 0xCD Bit 0 = 1 | → | DC Sensor enabled |

The measured value of the DC current sensor can be read via the ID-Address: 0xFA (32 Bit, RO, H/L, signed)

| | | | |
|---|---|---|---|
| 0xFA – L: | Idc actual filt | Unit: [dA] | (recommended) |
| 0xFA – H: | Idc actual | Unit: [Num] | |

Each High (H) and Low (L) 16 Bit signal is signed (+/-) with a min-, max scaling range depending on the device type.

## 3.3 DC Current Limiting – 0xCD Bit 1

The internal DC current limiting logic is designed as means of a battery protection logic.
The logic will try to limit the measured DC current to either one of the two specified current limits in motor or generator operation direction by reducing the allowed AC motor current.

Meaning it is not a DC current controller logic trying to control (hold) to a desired DC current value, but a safety supervision logic in case the commanded AC current will result in a higher DC current than the specified operation limit.

Since the limits can be updated at any time, various applications can still be used with this logic.
Especially in case of generator applications defining the max. allowed recuperation DC currents.

A condition for this logic to operate properly is the requirement of a DC current feedback value. This can either be the internal DC current sensor value or an external value send via the ID-Address 0xFB – L (Unit: [dA]).

Activation condition 1:

| | | |
|---|---|---|
| 0xCD Bit 1 = 0 | → | DC current limiting disabled |
| 0xCD Bit 1 = 1 | → | DC current limiting enabled |

DC current feedback signal selection:

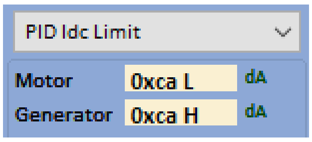
| | |
|---|---|
| Internal DC current sensor value: | 0xCD Bit 0 = 1 |
| External DC current value usage: | 0xCD Bit 8 = 1 |

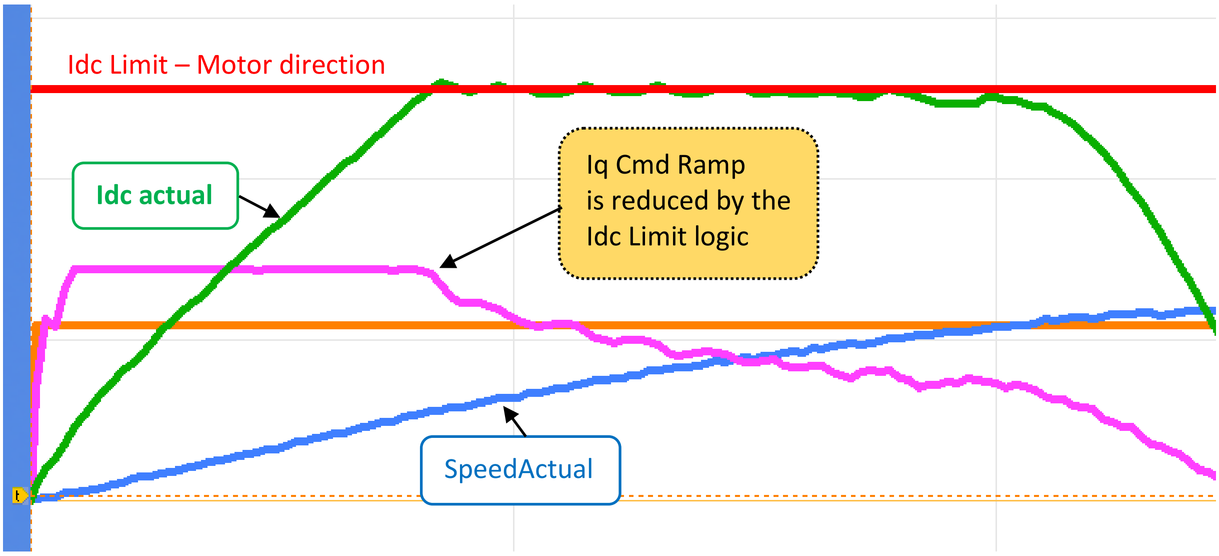
Note:
If the external DC current value usage is selected (0xCD Bit 8 = 1) the DC current limiting logic will only use the external value as the valid feedback value (→ ID-Address: 0xFB – L Unit: [dA] (16-Bit signed))

Limits Configuration:

| | | | |
|---|---|---|---|
| 0xCA – L: | Motor direction | Unit: [dA] | (only positive value entry) |
| 0xCA – H: | Generator direction | Unit: [dA] | (only positive value entry) |

Setting the limits using NDrive (*still under development*):
NDrive → Oscilloscope → drop down option ‘PID Idc Limit’.




Example during motor acceleration

### 3.4 DC Power Limiting – 0xCD Bit 2

The DC power limiting logic is only possible with specific inverter types which measure the analogue Vdc-Bus voltage.

The DC power limiting logic is basically an extended version of the DC current limiting logic.
The only difference is that the two specified power limits in motor or generator operation direction are used with the actual measured Vdc-Bus voltage to calculate a new Idc limit reference value.

Idc Limit (mot, gen) = Pdc Limit (mot, gen) / Vdc-Bus voltage

The DC current limiting logic then uses this updated value as the new reference value.

A condition for this logic to operate properly is the same as for the general DC current limiting logic.
A DC current feedback value must be present. This can either be the internal DC current sensor value or an external value send via the ID-Address 0xFB – L (Unit: [dA]).

Activation condition 1: (Enable general limiting logic)

| | | |
|---|---|---|
| 0xCD Bit 1 = 0 | → | DC current limiting disabled |
| 0xCD Bit 1 = 1 | → | DC current limiting enabled |

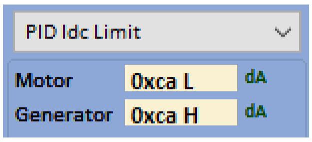
Activation condition 2: (Set it to power limiting logic)

| | | |
|---|---|---|
| 0xCD Bit 2 = 0 | → | DC power limiting disabled |
| 0xCD Bit 2 = 1 | → | DC power limiting enabled |

Limits Configuration:

| | | | |
|---|---|---|---|
| 0xCA – L: | Motor direction | Unit: [0.1 kW] | (only positive value entry) |
| 0xCA – H: | Generator direction | Unit: [0.1 kW] | (only positive value entry) |

Setting the limits using NDrive (*still under development*):
NDrive → Oscilloscope → drop down option ‘PID Idc Limit’.




Important: The entry value unit is then as [0.1 kW] and not as shown as [dA]

## 3.5 Enable with Torque Command – 0xCD Bit 3

While using digital control this logic allows free running enable control with a motor free of torque (FW ≥ 476).

If the “Command Mode” is set to “Dig. Commands” and the inverter is enabled (Ena = 1), the first automatic command request will always be a Speed-Control command with N set(dig.) = 0. Meaning that if the inverter is enabled while the motor is rotating, the inverter will try to ramp the motor to 0 rpm after enable. This can create high negative current spikes.
Only after the first torque command request (M set(dig.)) is received the inverter will change to Torque-Control.

With 0xCD Bit 3 it is possible to define whether the enable mode should be Speed-Control or Torque-Control.

| | | |
|---|---|---|
| 0xCD Bit 3 = 0 | → | Speed-Control after enable |
| 0xCD Bit 3 = 1 | → | Torque-Control after enable |

This means that with 0xCD Bit 3 = 1, it is possible to enable the inverter to a running motor in Torque-Control mode without any braking or driving torque (M set(dig.) = 0).

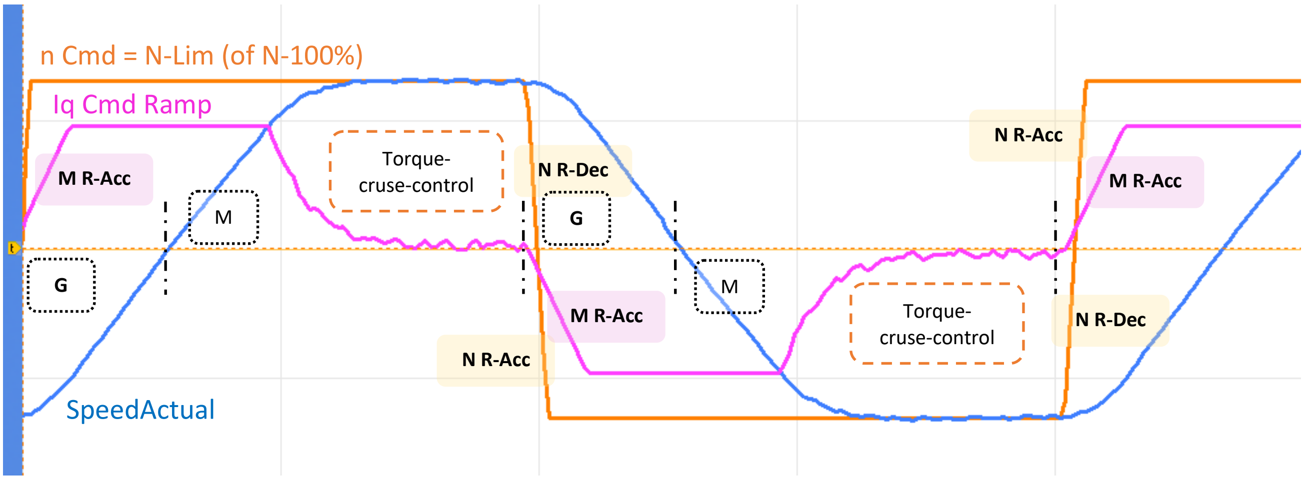
## 3.6 Torque recuperation ramp (M R-Rcp) – 0xCD Bit 4

Activate the ‘M R-Rcp’ ramps setting option, to control recuperation Acc torque ramps (FW ≥ 476).
A separate configuration of the recuperative current acceleration ramp is now possible by defining the recuperation ramps (M R-Rcp) independent of the motor acceleration (M R-Acc) ramp setting.

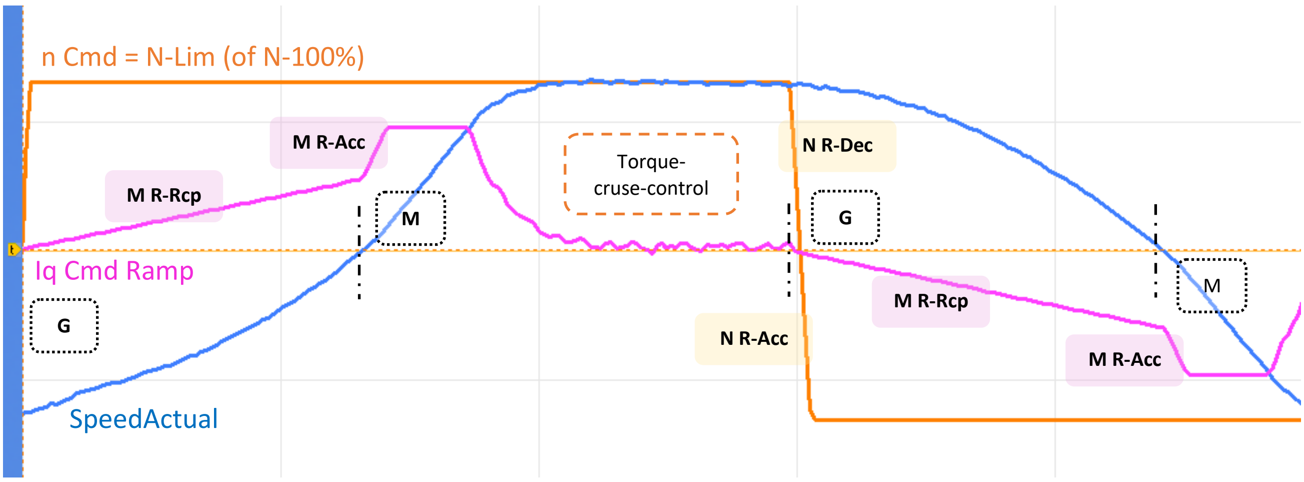
0xCD Bit 4 = 0 → Use “M R-Acc” (ID: 0x35-H)
0xCD Bit 4 = 1 → Use “M R-Rcp” (ID: 0xC7-H)

| | | |
|---|---|---|
| N R-Acc | 10 | ms |
| N R-Dec | 10 | ms |
| R-Lim | 1000 | ms |
| M R-Acc | 50 | ms |
| M R-Dec | 10 | ms |
| M R-Rcp | 500 | ms |

Example: 0xCD Bit 4 = 0 M R-Rcp Deactivated




Example: 0xCD Bit 4 = 1 M R-Rcp Activated




*G – Generator Operation*
*M – Motor Operation*

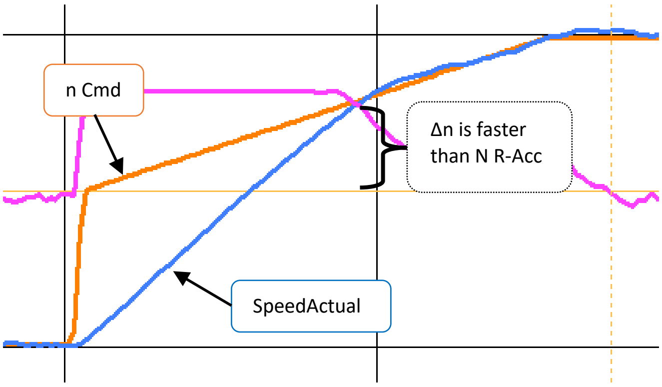
### 3.7 Reset SpeedRamp at n=0rpm – 0xCD Bit 5

This logic will reset the speed ramp to zero the moment the actual speed has got a sign change.
The idea is that the speed ramp will start new if a change of operation quadrant is present (e.g., motor direction change). Speed ramps can be much faster than the actual speed due to load conditions. With a reset a fresh ramp command can be generated.

Note:
This logic is only in operation if digital or analogue torque control is used, and the torque recuperation ramp is activated (0xCD Bit 4 = 1).

Activation condition:
(0xCD Bit 4 = 1) && (0xCD Bit 5 = 1) → Reset SpeedRamp at n=0rpm enabled

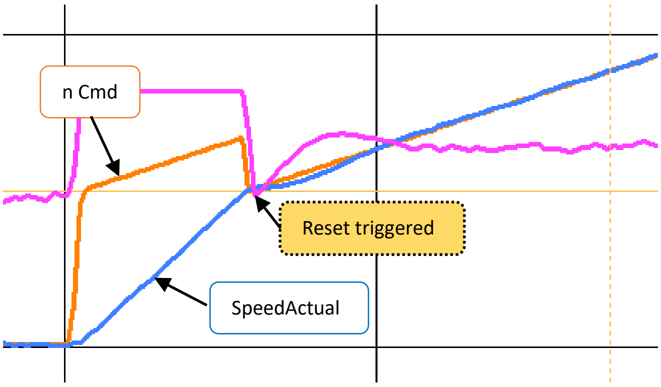
Reset Logic disabled:




| | | |
|---|---|---|
| N R-Acc | 500 | ms |
| N R-Dec | 10 | ms |
| R-Lim | 1000 | ms |
| M R-Acc | 10 | ms |
| M R-Dec | 10 | ms |
| M R-Rcp | 10 | ms |

→ After the motor rotation speed has a sign change, the motor actual speeds acceleration is faster than should be allowed by the definition of the N R-Acc.

Reset Logic enabled:




→ After the motor rotation speed has a sign change, the speed command ramp is reset.
It is as if the control logic starts a new phase.

**Important:**
This logic can have its flaws. Especially with high loads at very low rpm speeds causing unwanted reset triggers.

## 3.8 Torque TimeOut after 100 ms – 0xCD Bit 6

This logic is a special safety feature for digital torque control especially designed in case of a signal communication problem by an external acceleration pedal or lever.

It automatically sets the internal torque command to 0 if no new external torque command input within 100 ms is registered. Every time a new digital torque command is received, the 100 ms timer starts again.

This logic only sets the digital torque command to 0 Nm but does not disable the inverter output control.

## 3.9 Auto. N-Lim Saturation – 0xCD Bit 7

Automatic saturation of the max allowed rpm speed (N-Lim) depending on the measured Vdc-Bus and a fix constant factor value.

This logic is a simple safety feature to adjust the max. rotation speed depending on a battery's operation cycle (max voltage → min voltage) preventing that the inverter will try to reach points of operation not possible at a too low battery voltage.

Activation conditions:

| | | |
|---|---|---|
| 0xCD Bit 7 = 1 | → | Auto. N-Lim saturation enabled |
| N-Lim (ID: 0x34) < 100 % | → | Torque-cruse-control activated |

Configuration of the fix constant factor value:

0x87 – L: Kt Unit: [Vrms / rpm] (Note: 'Kt' is used for this logic as a workaround)

Explanation and Example:
A battery has got a maximum and minimum load voltage (e.g., max 400 V, min 300 V).
A motor has got a specific BackEmf constant for Idle operation (Vrms / rpm), but also a voltage drop when AC current is applied for creating torque. This voltage drop can be very high which also needs to be considered.

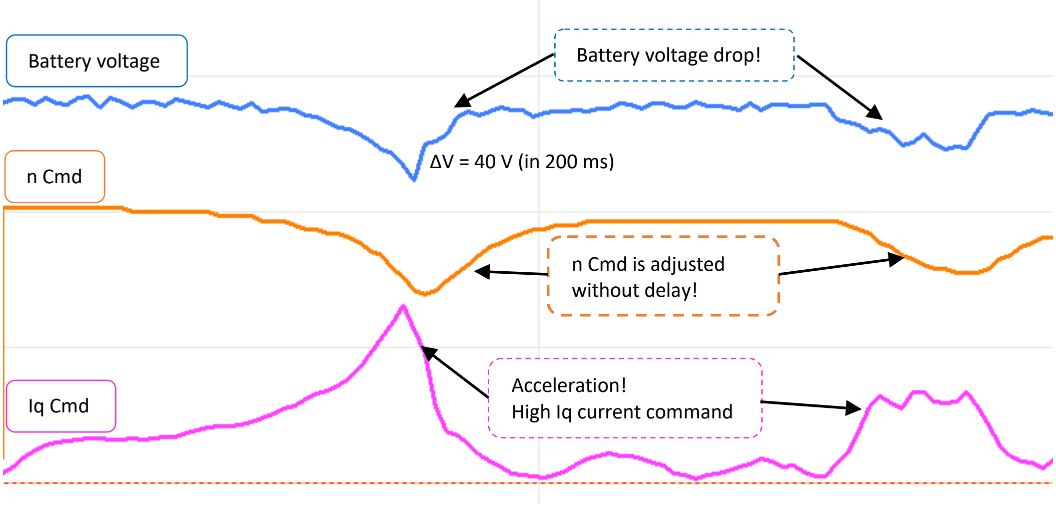
For example, an Emrax 208 HV motor has got a BackEmf constant of 0,04714 Vrms/rpm.
This means that at idle operation, during the max. to min. voltage state of our battery, the max allowed rpm speed is from 6000 rpm (at 400 V) down to 4500 rpm (at 300 V). This is already a very big difference!

- ⇨ This means that the max allowed rpm speeds must be adjusted to this battery change!
- ⇨ And the battery voltage can also drop very quickly if currents are required from it (see image)!

Additionally for example the Emrax manual states that the 208 HV motor, the BackEmf constant at 200 Arms is 11 rpm/V (0,06428 Vrms/rpm). The difference is really a lot and must be taken into consideration.

**Now** using a Iq current command of 200 Arms the max allowed rpm is from 4400 rpm (at 400 V) down to 3300 rpm (at 300 V). This means that at 400V there is a difference of 1600 rpm between Idle and Load operation!

**At the parameter Kt**, you can enter the BackEmf (with voltage drop) value, and this logic will constantly update the allowed commanded speed, ensuring always to stay inside valid operation points.




Operation example of the automatic saturation logic
in case of voltage drop because of a high current command.

### 3.10 Digital automatic recuperation logic (Brake Car)

#### 3.10.1 Digital Torque Control with automatic recuperation logic in general

As explained in chapter 1.2, the theory of recuperative braking is actually very simple. As soon as the motor phase current (I_actual) and the motor phase voltage (SpeedActual) are of opposite sign, then recuperative energy will go back to the battery.

This means that all you must do is to send a negative torque command for generator torque, and you will be recuperating current back to the battery ($P_{DC} \leftrightarrow P_{AC}$).

Using digital torque command, the user has complete control whether the demanded torque will be a driving or a generator torque (recuperative braking) just by sending the desired torque value (M set (dig.)) using one of the digital communication interfaces (CAN, RS232).

The downside of this technique is that the refresh rate for the calculation is limited to the same refresh rate as the communications interface (e.g., every 10 ms).

The automatic recuperation logic is a function which will calculate a recuperative braking logic internally just by triggering the activation depending on the settings. The logic is the same as with the analogue logic except that the trigger command must also be a digital trigger.

**Enable** special function "recuperative braking logic":
→ set one digital input to the function "Brake Car".

**Trigger** of the special function "recuperative braking logic" (3 logics)
→ Logic 1: Enable the logic "Dig. E Brake At0Nm Cmd" (0xCD Bit 14 = 1).
The trigger will be when Mset(dig.) = 0.
→ Logic 2: Setting the "Dig. E-Brake Activation" Bit to 1 (0xCD Bit 15 = 1).
→ Logic 3: The configured input (e.g., Din1 = "Brake Car [Ax]") is triggered.

**Calculation** logic for the braking torque command:
→ Logic 1: Torque = (-SpeedActual * N-Lim+) * fac (using parameter "N-Lim+" (ID: 0x3F))
→ Logic 2: Torque = (-SpeedActual * N-Lim+) * fac (using parameter "N-Lim+" (ID: 0x3F))
→ Logic 3: Torque = ( SpeedActual * N-Lim- ) * fac (using parameter "N-Lim-" (ID: 0x3E))

Note:
- 'N-Lim+' and 'N-Lim-' represent the maximum possible current in % of I max peak = 100 %.
  In case of long "N R-Dec" Ramps, different responses will apply.
- The multiplication factor 'fac' can be configured via 0xCD Bit 13..12.
  The default value is fac = 1.

**Ramp generator** for all analogue torque command values:
- M R-Acc → rate of increase of driving torque.
- M R-Dec → rate of decrease of driving torque.
- M R-Rcp → rate of increase of recuperative braking torque if either Logic 1 or 2 is activated.

**Disable** the "recuperative braking logic": (without settings change)
- Set one digital input to "recu-disab [Ax]". If this input is triggered, it will force any recuperation torque to zero (e.g., in case of a full battery). This is then without settings change but with an extra input.

### 3.10.2 Auto. Brake Multipl. Factor – 0xCD Bit 13..12

These 2 Bits are for an additional configuration option for the logic of the digital and analogue recuperation logic. They define the multiplication factor (=> fac) for calculating the braking currents. The idea is to give the operator a possibility to configure more torque than the standard recuperation formula would command.

| 0xCD – Bit 13..12 | Multiplication factor |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

### 3.10.3 Dig. E Brake At0NmCmd – 0xCD Bit 14

This is an additional option for digital torque control to trigger the recuperation logic (Brake Car).
It activates the trigger option that will activate the automatic recuperative braking logic, if the received torque command equals 0 (Mset(dig.) = 0). For example, if the acceleration pedal would go to 0 torque command.

Activation conditions:

0xCD Bit 7 = 14 → Auto. Recuperation at 0 cmd enabled
and
Set a digital Input to Brake Car → Enable the Auto. Recuperation Logic

### 3.10.4 Dig. E-Brake Activation – 0xCD Bit 15

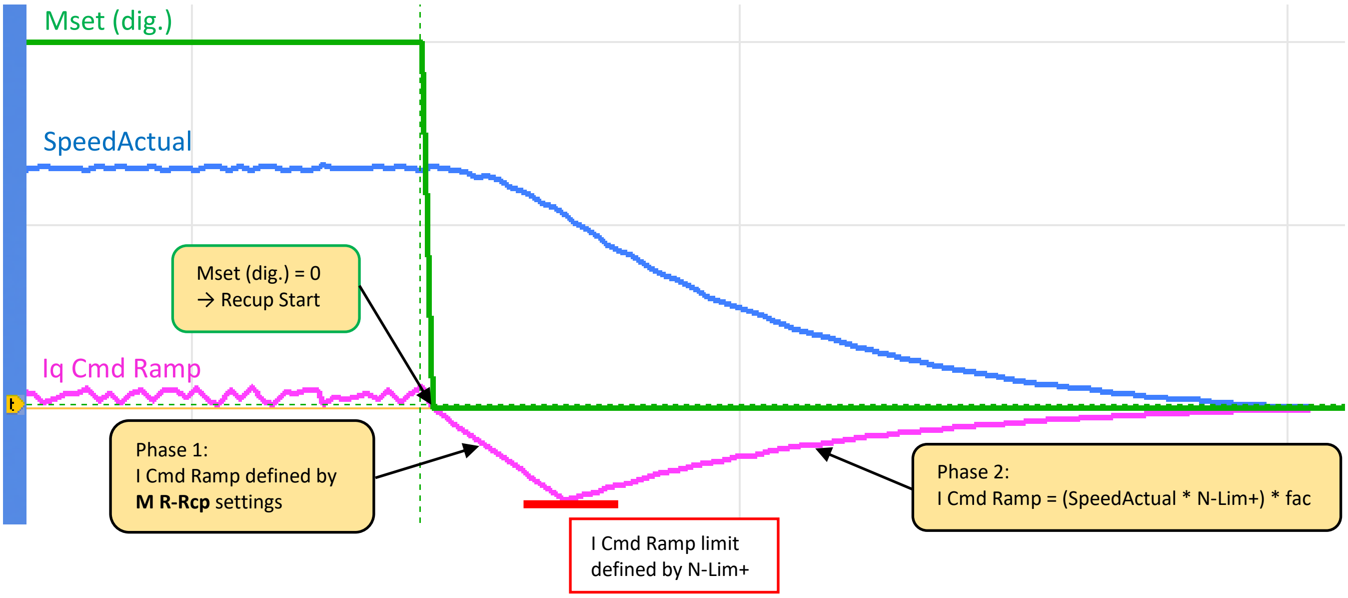
This is basically a trigger Bit for the digital automatic recuperation logic.
If this Bit is set, the inverter will activate the automatic recuperation logic and start sending a braking torque command.

Activation conditions:

Set a digital Input to Brake Car → Enable the Auto. Recuperation Logic

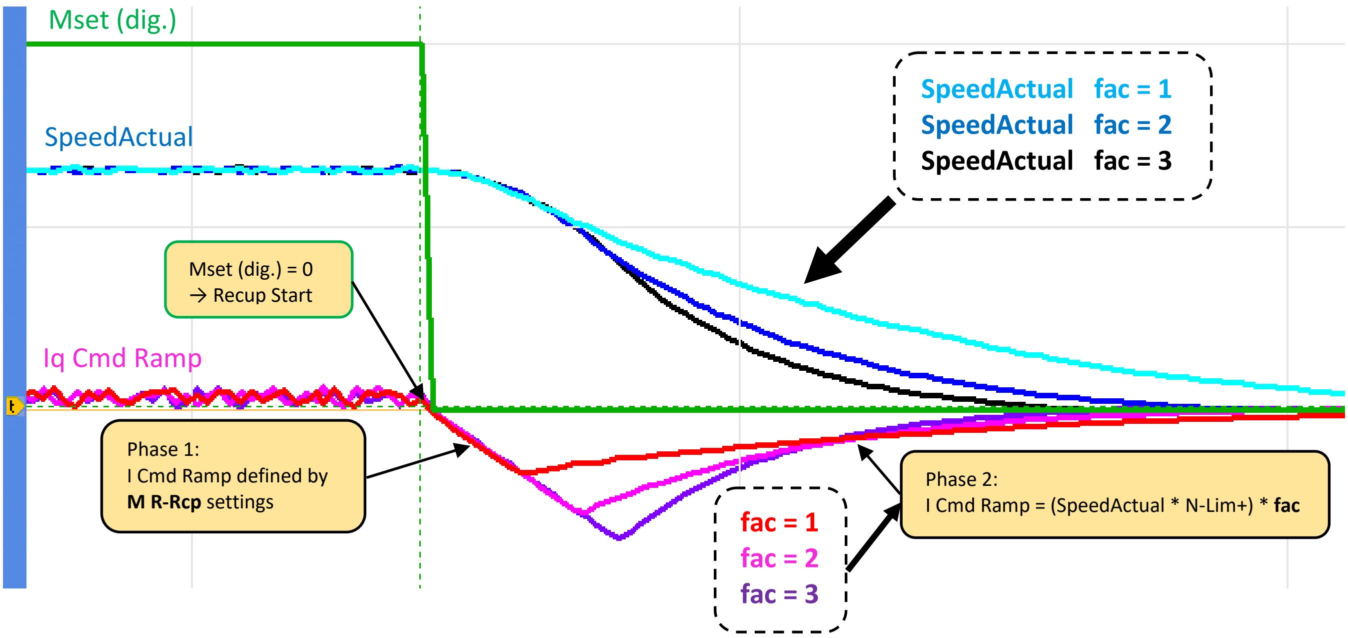
Example 1: General digital automatic recuperation logic

Trigger logic 2 (Mset (dig.) = 0)), 'M R-Rcp' activated and a standard 'fac = 1' multiplication value.




Example 2: Using different multiplication factors 'fac'

Trigger logic 2 (Mset (dig.) = 0)), 'M R-Rcp' activated and different 'fac = 1, 2, 3 ' multiplication value.



### 3.11 Traction Control (TC)

#### 3.11.1 Enable Traction Control (TC) – 0xCD Bit 16

*Note: Still under development. Unwanted behaviour could happen in certain situation. Testing is required.*

This traction control logic is designed for the operation mode of general torque control with activation of torque-cruse-control (N-Lim < 100%).

It must be clear that this is not a traction control logic like an ESP system in a car, where the information of all 4 wheels is used to stabilise the car or to stop one wheel to accelerate faster than the other.

This logic will constantly check if the actual speed of the motor is faster than the allowed settings of N R-Acc depending on defined time intervals.

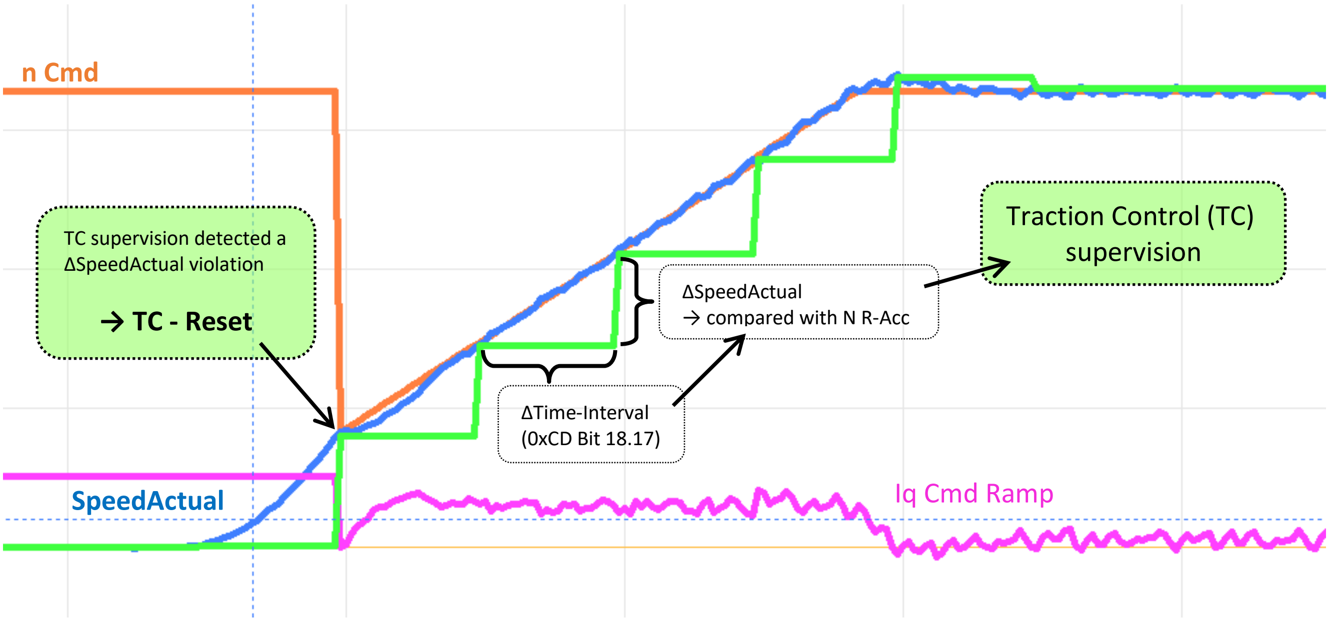
Using torque control, this can happen in a lot of situations.
E.g., your wheel is at a pathway curb and the driver needs a lot of torque to get over this pathway curb. After that the applied torque will cause the vehicle to accelerate at a very high rate since the high torque is still applied.

Activation conditions:

| | | |
|---|---|---|
| 0xCD Bit 16 = 1 | → | Auto. N-Lim saturation enabled |
| N-Lim (ID: 0x34) < 100 % | → | Torque-cruse-control activated |

Configuration:

| | | |
|---|---|---|
| N R-Acc (ID: 0x35 – L): | → | Max allowed acceleration speed |
| 0xCD Bit 18..17 | → | Define the delta times |
| 0xCD Bit 20..19 | → | Define the min. speed for TC activation |
| 0xCD Bit 21 | → | n Command reset option |



### 3.11.2 TC- DeltaTime configuration – 0xCD Bit 18..17

This configuration option defines the delta time interval of the traction control logic.
It defines the time steps to calculate the delta SpeedActual change and then compare it with the allowed delta change.

| 0xCD – Bit 18..17 | Delta Time Interval [ms] |
|---|---|
| 0 | 20 ms |
| 1 | 50 ms |
| 2 | 100 ms |
| 3 | 250 ms |

### 3.11.3 TC- Min. Speed configuration – 0xCD Bit 20..19

With this configuration option, the rpm speed starting point for the traction control logic can be defined.
The 4 percentage options, defining the used rpm speed as starting points, are calculated according to the reference inside the parameter N-100%.

| 0xCD – Bit 20..19 | Min. rpm speed for TC activation [% of N-100%] |
|---|---|
| 0 | 0 % |
| 1 | 10 % |
| 2 | 20 % |
| 3 | 30 % |

### 3.11.4 TC- n Command Reset Option – 0xCD Bit 21

Define 2 different options on how the speed command should be reset in case the traction control logic triggers an acceleration violation.

Configuration:

0xCD Bit 21 = 0

→ n Command reset to the SpeedActual value

0xCD Bit 21 = 1

→ n Command reset between the SpeedActual value, and the traction controls last checked SpeedActual value. The reset point is therefore lower than the standard logic.

# 4 ID Register 0xDC – Special Configurations and Control Functions Part 2

The following configurations and special functions of the ID Register 0xDC are valid for FW ≥ 480.
These are also visualised and can be configured in NDrive at page 'Device'.

## 4.1 Configuration Bitmap – ID: 0xDC

| Symbol: | Function: | ID-Address: |
|---|---|---|
| | | **0xDC** |
| DAC Output Channel | Configuration of the DAC output selection | Bit 7..0 |
| | | |
| CAN Extended | CAN messages will always send a data length of 8 Bytes | Bit 8 |
| Generator Mode V1 | Generator logic by controlling the Vdc-Bus level for DC-loading<br>→ This is a very customer specific logic and will not be explained | Bit 9 |
| *rsvd_10* | … | Bit 10 |
| Vdc-Bus Unit | Change the unit on how the Vdc-Bus voltage value (ID: 0xEB) is transmitted [0 = Num ; 1 = dV] | Bit 11 |
| ENA only at cmd = 0 | Safety logic, where the inverter only enables if speed and torque command is 0 | Bit 12 |
| Generator Mode V2 | Generator logic where the max. allowed current will automatically be limited according to the Vdc-Bus voltage value → Full Battery = No recuperation current allowed | Bit 13 |
| MTPA Logic* | Maximum Torque Per Ampere (MTPA) calculation for motors if different L sigma (d,q) constructions | Bit 14 |
| CanOpen Acv | Defines the used CAN protocol [0 = CAN ; 1 = CANopen]<br>*Application User Protected* | Bit 15 |
| | | |
| Bamobil 3.2 I`Offset** | Current offset correction for Bamobil 3.2 inverter type<br>(Only specific inverters) *Application User Protected* | Bit 16 |
| ECode Re-Define** | Re-Definition on how the ECode message is processed<br>(Only specific inverters) *Application User Protected* | Bit 19..17 |
| HAL Estimator | Interpolation of the HAL feedback signal to achieve linearization of the electrical angle | Bit 20 |
| SW-Lock Init | During Initialisation the Inverter will automatically set the SW-Lock (ID: 0x51 Bit 2) → SW disable active | Bit 21 |
| Auto. CAN Msg | 8 Pre-Defined Variables are configured to be send on the CAN Bus at inverter initialisation | Bit 22 |
| SC Compensator | Special compensation logic to improve the Sine and Cosine feedback signals | Bit 23 |
| | | |
| PID I – DIC* | Dynamic Integral Clamping for current PID controller<br>*Application User Protected* | Bit 24 |
| *rsvd_25* | … | Bit 25 |
| Auto. 16kHz PWM switch* | Auto. Switch to 16kHz PWM frequency if motor speed is above a defined rotation speed<br>*Application User Protected* | Bit 26 |

| | | |
|---|---|---|
| PWM Bootstrap** | Disable the Bootstrap logic during PWM Initialisation<br>(Only specific inverters) *Application User Protected* | Bit 27 |
| 10Hz Derating | Disable the 10Hz derating logic<br>*Application User Protected* | Bit 28 |
| Check I peak configuration | Configuration for the Critical Currents Shut-Off Limits<br>*Application User Protected* | Bit 31..29 |
| | | |
| * In development…<br>** Special Inverter type settings → Do not change!<br><br>Note: Some functions require certain inverter types | | |

### 4.2 CAN Extended – 0xDC Bit 8

This option defines the length of a transmitted CAN message. Usually, the length of a transmitted CAN message is determined by the length of the variable (16 Bit or 32 Bit).
Some receiving CAN modules require a fix length for incoming messages to process them. With this option the transmitted CAN messages will always send a message length of 6 Bytes.

0xDC Bit 8 = 0
→ length of CAN Tx message depending on variable length
16 Bit variable = 4 Bytes
32 Bit variable = 6 Bytes

0xDC Bit 8 = 1
→ length of CAN Tx message always fix
16 Bit or 32 Bit variable = 6 Bytes

### 4.3 Vdc-Bus Unit – 0xDC Bit 10

This option defines the unit of the Vdc-Bus value transmitted on the ID-Addresses 0x66 and 0xEB.
ID: 0x66 – VdcBus
ID: 0xEB – VdcBus filtered

The standard unit is [Num] which then needs to be converted by using the device conversion table or factor inside the inverter manual. If this Bit is selected, the software will use the internal device specific conversion factor to transmit the Vdc-Bus values as the unit [dV].

Configuration:
0xDC Bit 10 = 0 → Vdc-Bus as [Num]
0xDC Bit 10 = 1 → Vdc-Bus as [dV]

Note:
In NDrive page “Device”.
NDrive will always convert the received value to a physical value [V], using its own conversion factors.
NDrive does not know that 0xDC Bit 10 is set and that the received value is already converted to a [dV] value.

## 4.4 ENA only at cmd = 0 – 0xDC Bit 12

This option is a safety logic specifically designed for analogue control mode (speed, torque).
Only when the resulting analogue command equals 0, will the inverter software logic start the enable process.

This logic is designed for analogue control applications such as e.g., a boot with an analogue control lever. Only when the control lever is set to the zero-command position the electrical engine can be started.

Activation conditions:

| | | |
|---|---|---|
| Command Mode | → | Analog Speed or Analog Torque |
| 0xDC Bit 12 = 1 | → | ENA only at cmd = 0 |

If the Command Mode is set to dig. Commands, this logic will not be active.
Using digital control mode and the inverter is disable, any incoming commands are set to 0 anyway.

## 4.5 Generator Mode V2 – 0xDC Bit 13

This is a special generator control logic for any applications using an electric motor to load a battery.

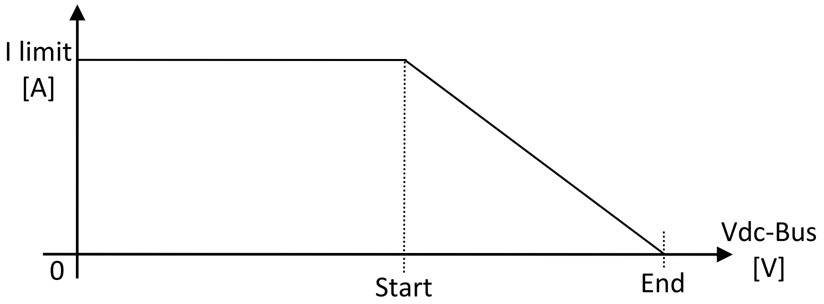
It controls the max. allowed AC current limit and will automatically reduce it according to the Vdc-Bus voltage value. The goal is that when the measured battery voltage reaches a defined voltage value, the recuperation current will be reduced until the battery is full, and the logic will not allow any generator AC current commands.

The reference value for this logic is always the measured Vdc-Bus voltage.

In total, 2 break points can be defined:

| | | | |
|---|---|---|---|
| Start Point = Var 4 | Unit: [dV] | ID: 0xD4 | (In NDrive page "Logic" → Var 4) |
| End Point = Device Mains | Unit: [V] | ID: 0x64 | |

Between the Start and the End Point a linearisation will reduce the allowed regenerative currents.
If the Vdc-Bus voltage reaches the End Point voltage, any recuperation current should stop automatically.




It is only active as long a generator operation is recognised.
Switching back to motor operation, the reduction of the current limits is reset to the standard settings.

### 4.6 MTPA logic – 0xDC Bit 14

This option logic is primarily designed for the usage of PMSM motors with a reluctance property in its design.

The logic will automatically calculate and command the required Id-current to have maximum torque per ampere operation depending on the motor construction.

This logic requires the motor specific data of Cos Phi, L sigma-d, L sigma-q and R stator for its MTPA calculations.

Activation conditions:

| | | |
|---|---|---|
| 0xDC Bit 14 = 1 | → | MTPA calculation and Id cmd enabled |
| L sigma-d ≠ L sigma-q | → | Motor with reluctance properties |

### 4.7 CANopen Acv – 0xDC Bit 15

This option will let you choose whether to use the standard UniTek propriety CAN protocol or a CANopen protocol. The default option is the UniTek propriety CAN protocol since it allows the access to all signals and parameters, while CANopen is limited to a small pre-defined selection of signals and parameters.

Configuration:

| | | |
|---|---|---|
| 0xDC Bit 15 = 0 | → | Propriety CAN |
| 0xDC Bit 15 = 1 | → | CANopen |

### 4.8 HAL Estimator – 0xDC Bit 20

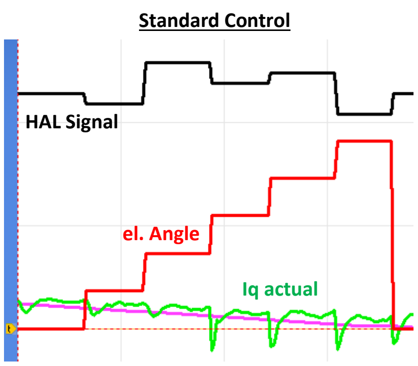
This option tries to improve the feedback quality of 3 digital HAL sensors. It does this by the method of Interpolation of the HAL feedback signal to achieve linearization of the electrical angle.

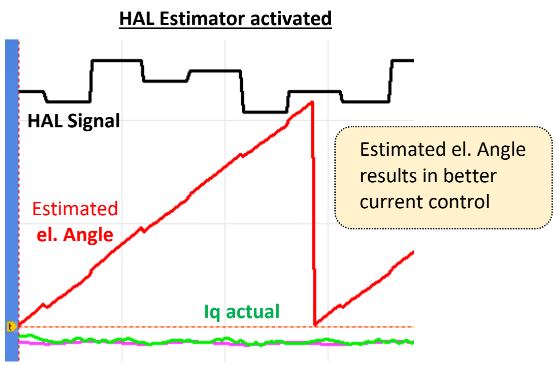
**Important:** It is important to understand that the quality of field orientated control (FOC) of a PMSM motor with sinusoidal BackEmf curve in combination with 3 digital HAL sensors as a feedback system, especially at higher rpm speeds, **is generally not recommended**!

The HAL Estimator will start its operation above a rpm speed of 5000 Num and stop when it is below again.

Activation condition:

| | | |
|---|---|---|
| 0xDC Bit 20 = 1 | → | HAL Estimator activated |




**Note:**
In case a motor with trapezoidal BackEmf curve with 3 digital HAL sensors as feedback system is controlled using the Block-Commutation PWM operation logic, the HAL Estimator logic should not be used.

### 4.9 SW-Lock Init – 0xDC Bit 21

This option will have the effect that during the startup initialisation process the Inverter will automatically set the SW-Lock (ID: 0x51 Bit 2). This can be useful for hard wired RFE and RUN(FRG) connections and an enable process via CAN, that a first SW-Lock command does need to be send.

Activation condition:

0xDC Bit 21 = 1 → SW-Lock is set during the startup initialisation process

### 4.10 Auto. CAN Msg – 0xDC Bit 22

This option will configure 8 pre-defined variables to be send on the CAN Bus periodically using a pre-defined time interval for each variable. It basically uses the cyclic transmission function (example 9 from the CAN manual), which usually needs to be configured manually.

Using this option, the advantage is that no manual configuration needs to be done, the variables are recommended by us and that the initialisation of the cyclic transmission is done properly.

This option is especially useful for simple applications with only a CAN data logger.

Activation condition:

0xDC Bit 22 = 0 → Disable Automatic CAN Message Logic
0xDC Bit 22 = 1 → Enable Automatic CAN Message Logic

| Signal Name: | ID-Address: | Function: | Time: |
|---|---|---|---|
| N cmd ramp | 0x32 | Speed command ramped | 10 ms |
| N actual | 0x30 | Speed actual unfiltered | 10 ms |
| Iq cmd ramp | 0x22 | Iq command ramped (active current) | 10 ms |
| Iq actual | 0x27 | Iq actual unfiltered (active current) | 10 ms |
| Vout | 0x8A | Voltage percentage output (Vq, Vd) | 10 ms |
| Vdc-Bus | 0x66 | Vdc-Bus voltage unfiltered | 10 ms |
| T-igbt | 0x4A | Igbt temperature | 100 ms |
| T-motor | 0x49 | Motor temperature | 100 ms |

### 4.11 SC Compensator – 0xDC Bit 23

This option tries to improve the feedback quality of analogue Sine and Cosine (SC) feedback sensors.
It analyses the amplitude and the mid-point of both oscillating signals and tries to adjust any mid-point offsets and multiplies a gain factor to the amplitude to create a proper compensated Sine and Cosine feedback signal.

It is important to understand that this function has got its limits. If the Sine and Cosine Signals or too distorted or are too far of alignment, then the SC compensator will not be able to adjust this.

Activation condition:

0xDC Bit 23 = 1 → SC compensator enabled

Note:
The SC Compensator can only correct the signals after it has measured the actual state of the SC signals. Meaning the motor will have to rotate a couple of turns for the compensator to make any adjustments.

After a proper compensation has been done, it is recommended to disable the drive and then save to Eprom 0. The compensation offset and gain values are saved to the Eprom and are loaded at the next restart.