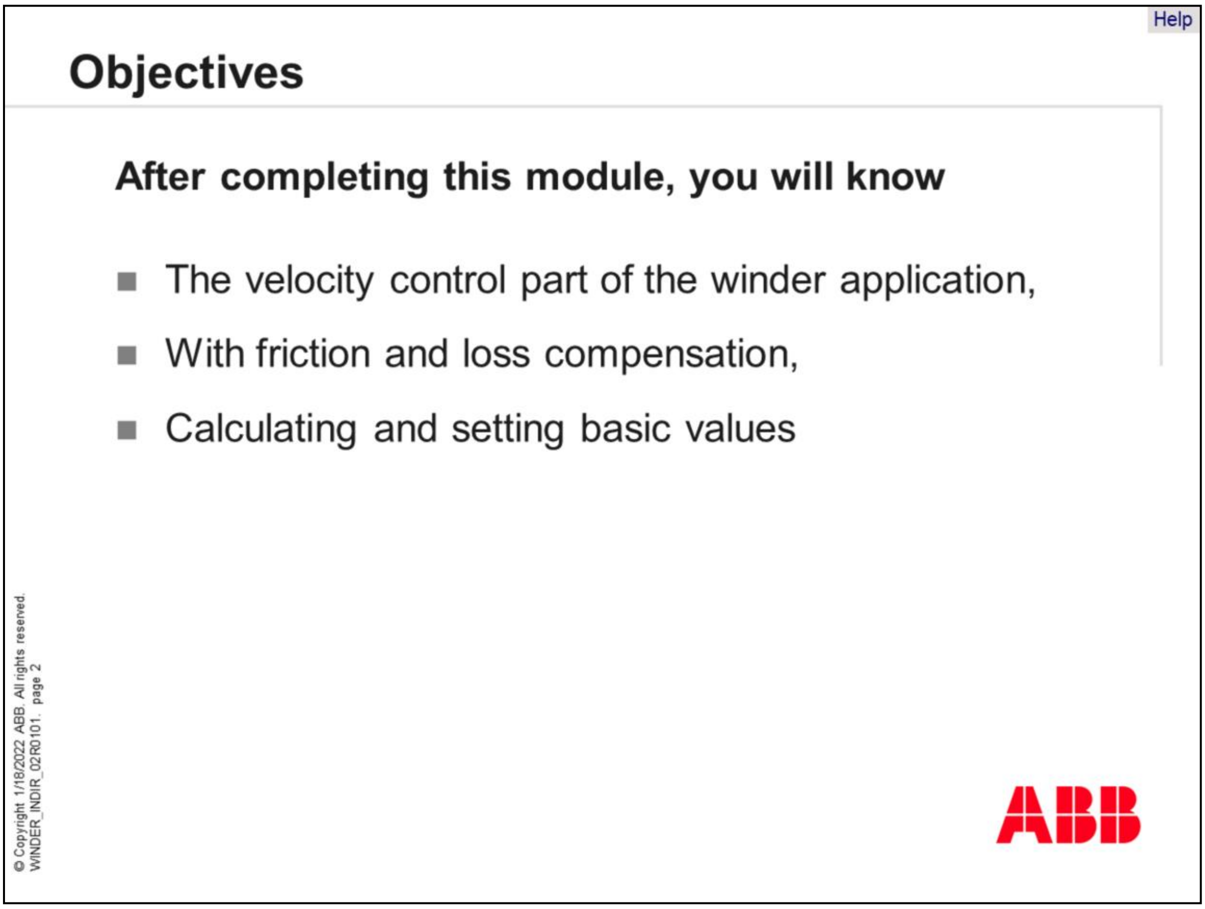
Welcome to the winder application training module "Winder application" for the DCS800, one of the ABB DC Drives.

If you need help navigating this module, please click the Help button in the top right-hand corner. To view the presenter notes as text, please click the Notes button in the bottom right corner.



After completing this module, you will know and be familiar with

- The velocity control part of the winder application,
- Friction and loss compensation and
- Calculating and setting basic values

Indirect Tension Control

Help

- Diagram of indirect tension control
- Tension reference and motor current
- Diameter calculation
- Velocity or line speed control
- Friction and loss compensation
- Inertia calculation for acceleration and deceleration
- Winder logic control
- Significant equations

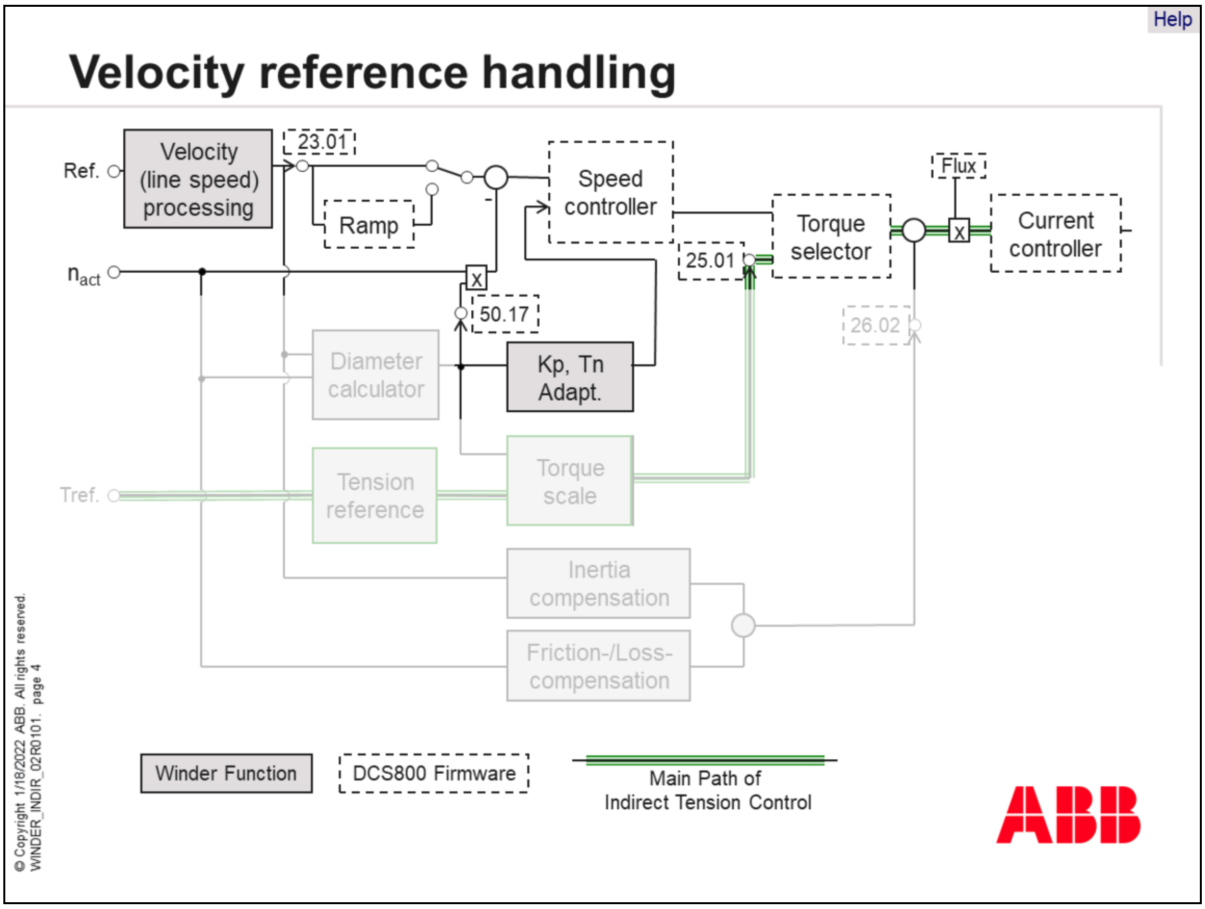
This module describes the "Indirect Tension Control" for the DCS800 winder application.

In part 1 of the module "Indirect Tension Control" the following aspects were covered:

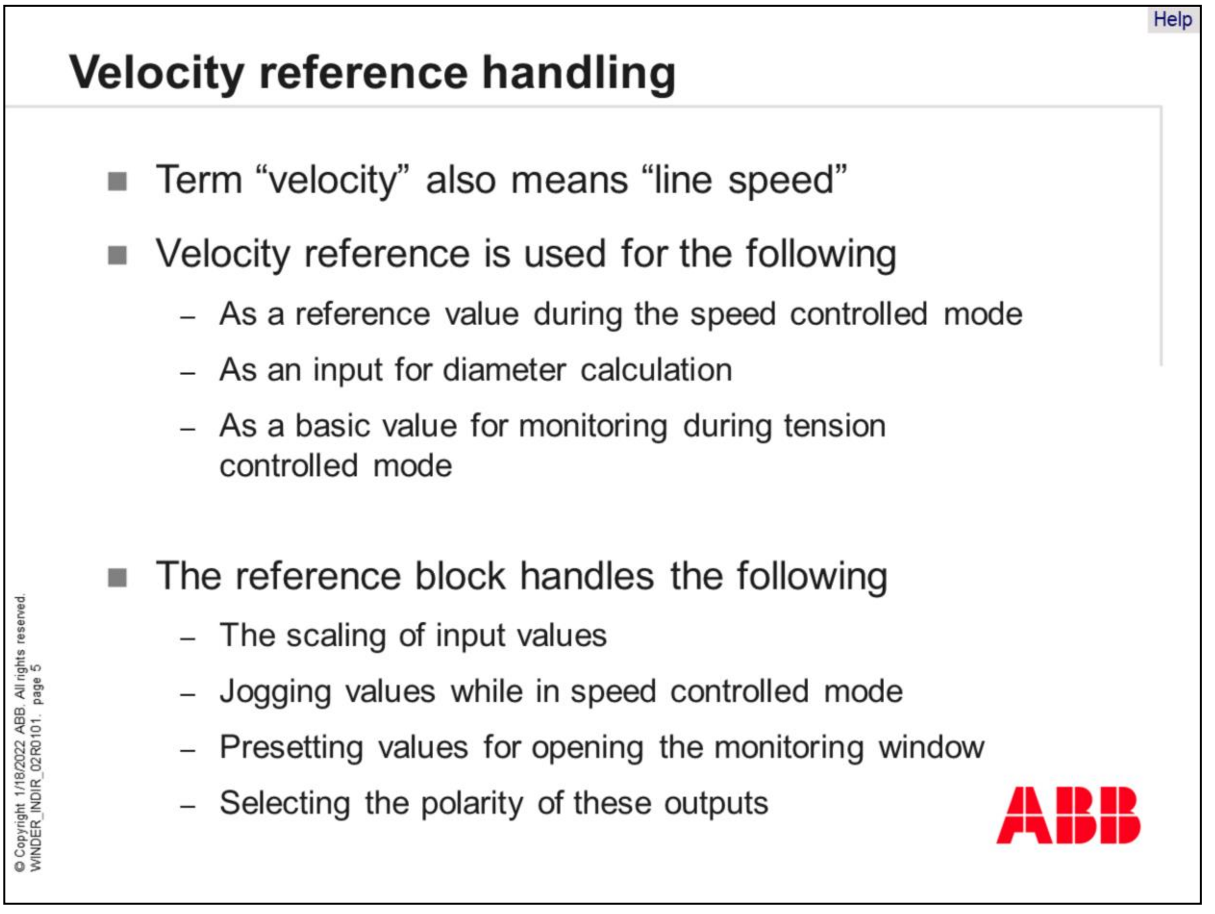
- The control diagram of indirect tension control,
- The tension reference and motor current part and
- The diameter calculation

Now in part 2 the following topics will be discussed:

- Velocity or line speed control
- Friction and loss compensation
- Inertia calculation for acceleration and deceleration
- Winder logic control
- Significant equations



The topic “velocity reference” discusses handling velocity references, including jogging. This part will also describe the adaptation of the PI values of the speed controller.



The term velocity, which is used here, also means line speed.

Velocity reference is used for the following:

- as a reference value during the speed controlled mode
- as an input for diameter calculation. For proper diameter calculations, the input value has to be scaled exactly.
- as a basic value for monitoring during tension controlled mode

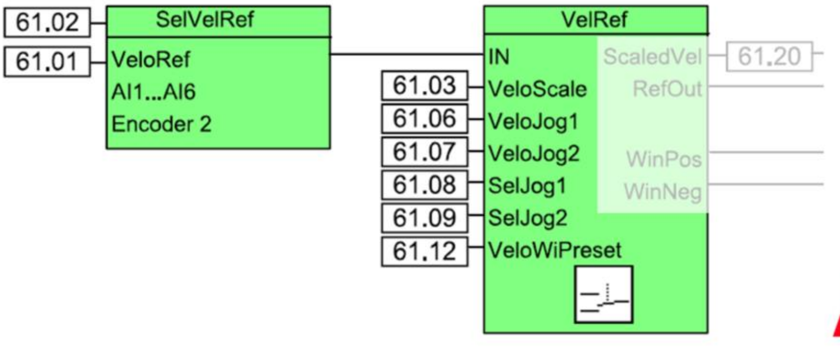
The reference block handles the following

- The scaling of input values
- Jogging values while in speed controlled mode
- Presetting values for opening the monitoring window
- Selecting the polarity of these outputs depending on winding mode and winding direction. These signals come from the winder logic.

Help

Velocity reference handling

- Velocity input can be selected by using parameter 61.02
- Reference can be scaled, if needed
- Internal jogging references during speed mode
  - VeloJog1 (61.06) and VeloJog2 (61.07)
  - Selectable by using commands
- Preset value during tension controlled mode, parameter 61.12



The input of velocity reference can be selected by using SelVelRef, parameter 61.02. The velocity reference from a serial communication can be written to parameter 61.01.

The incoming value can be scaled, if needed, by using VeloScale and can be set using parameter 61.03.

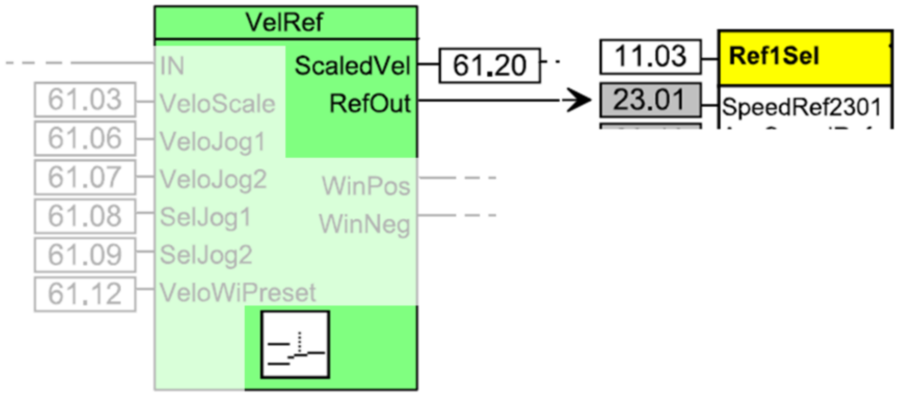
Internal jogging references can be defined during speed mode with the values of VeloJog1, parameter 61.06, and VeloJog2, parameter 61.07. The commands of each jogging value can be selected by using SelJog1, parameter 61.08, and by using SelJog2, parameter 61.09.

During winding mode, the parameter 61.12, called VeloWiPreset, will open the monitoring window. More details will follow at the end of this block.

Help

Velocity reference block outputs

- Outputs of velocity reference block
  - ScaledVel (61.20) are used for diameter calculation
  - RefOut is written to SpeedRef (23.01) of standard firmware

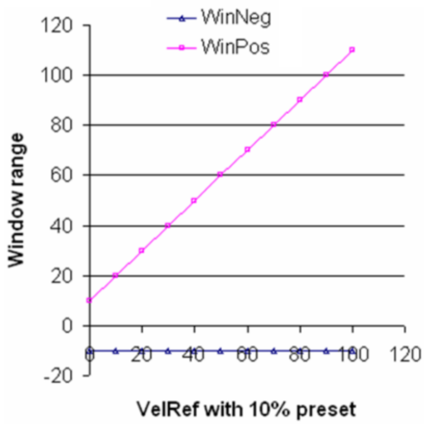


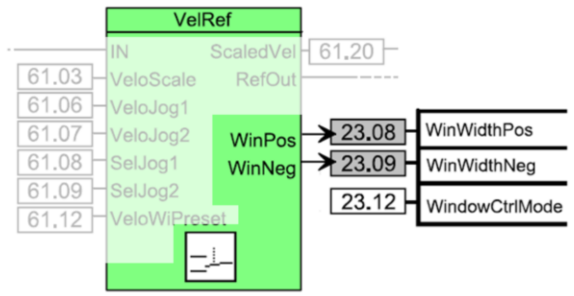
Two of the outputs are the ScaledVel (61.20), which is used for diameter calculation, and RefOut, which is written to SpeedRef, parameter 23.01 of the standard firmware.

Help

Velocity reference block outputs

- Window levels for monitoring
  - Outputs WinPos and WinNeg (for positive and negative window level) according winding mode and direction.
  - WinPos is written to **WinWidthPos** (23.08 of DCS800 firmware) and WinNeg to **WinWidthNeg** (23.09)
  - Example: 0…100% velocity reference and VeloWiPreset = 10%





During tension control mode, the speed controller operates in window mode for monitoring the actual velocity.

The output values "WinPos" and "WinNeg" are calculated with velocity reference and parameter 61.12, called "Velo Wi Preset". Winding mode and winding direction are taken into account in these calculations.

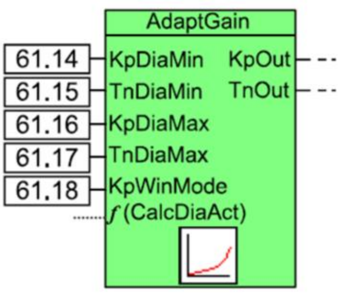
The positive value "WinPos" will be written to DCS800 parameter 23.08, called "Win Width Pos". In the same way, the negative value "WinNeg" will be written to parameter 23.09, called "Win Width Neg".

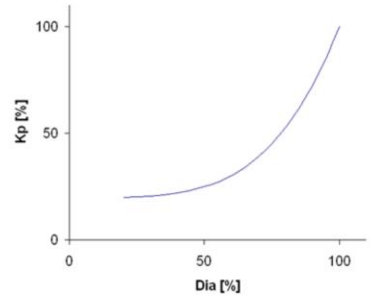
The diagram on the left hand side shows an example of a 10% limitation if a rewinder is working in advance.

Help

Velocity Adaptation

- Adaptation of velocity (speed) controller, interpolated according the diameter.
  - Kp values for minimum diameter and maximum diameter
  - Tn values for minimum diameter and maximum diameter
  - Kp value for monitoring (window mode)
- Outputs
  - KpOut is written to KpS (24.03) of speed controller
  - TnOut is written to TiS (24.09) of speed controller





During velocity controlled mode, the controller needs Kp and Tn values depending upon diameter, which modification causes the change of inertia. This is solved with the block AdaptGain. Each value can be set for minimum diameter, KpDiaMin (61.14) and TnDiaMin (61.15), and for maximum diameter, KpDiaMax (61.16) and TnDiaMax (61.17). Between these points the control values are interpolated depending on the diameter.

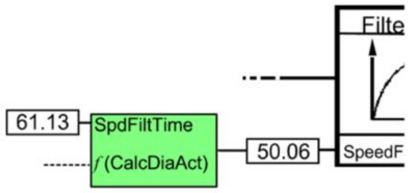
During indirect tension controlled mode, the Kp value of KpWinMode, parameter 61.18, is active, which value can be set high.

The result of the adaptation will be written to the standard parameters 24.03 as KpS and 24.09 as TiS.

Help

Velocity control

- Ramp generator and speed controller, running as velocity controller, are used from standard firmware
- Speed filter time (61.13)
  - Adjusted by diameter
  - Written to SpeedFiltTime (50.06) of standard firmware.
  - Please note:
    Filter time influences the gain of SpeedCtrlAutotuning, if used.



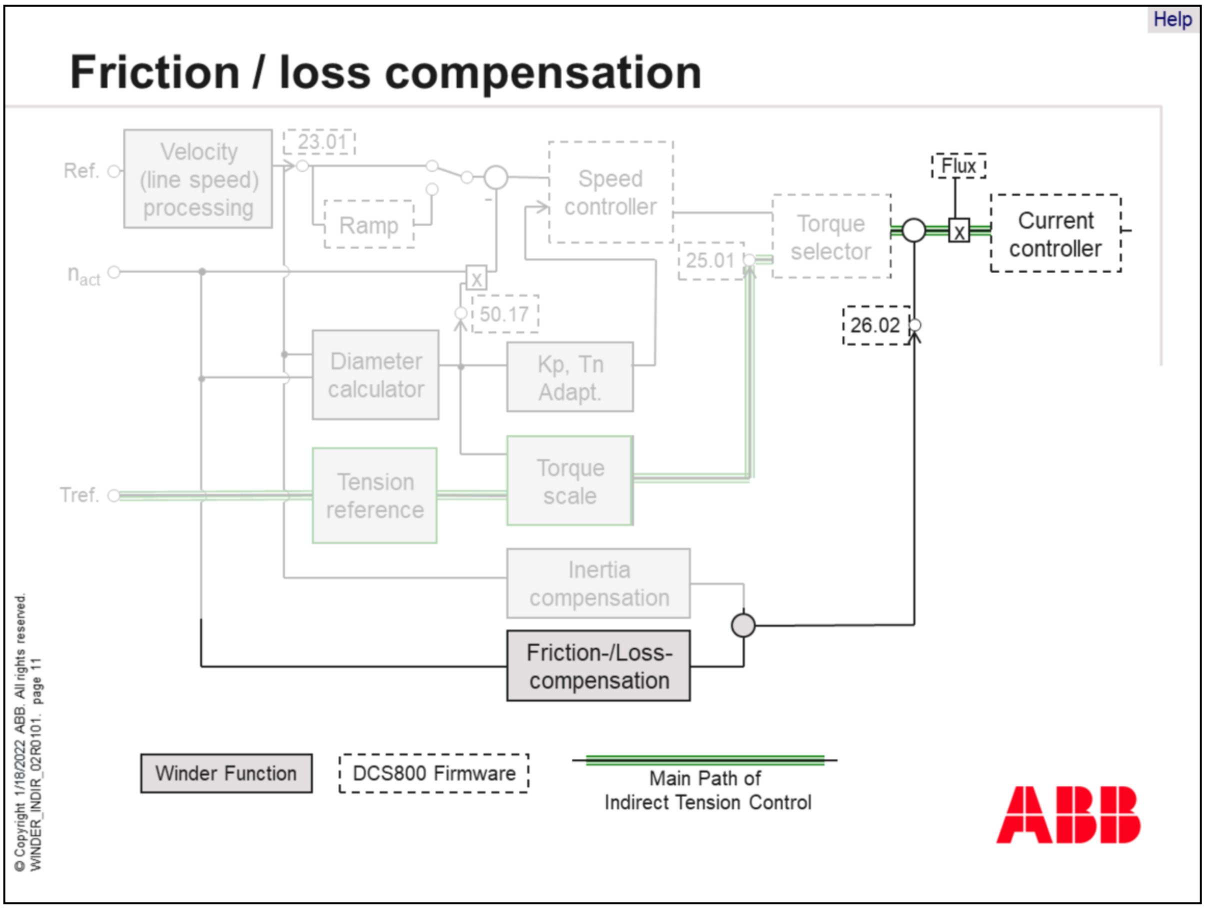
In velocity controlled mode, the velocity reference will be directed via the ramp generator from the DCS800 firmware.

The speed controller, also from the DCS800 firmware, is used as a velocity controller in velocity controlled mode and as a monitoring system in tension controlled mode.

The speed feedback of the motor will be filtered depending on the diameter by using parameter 61.13 SpdFiltTime These values are written to parameter 50.06 SpeedFiltTime of DCS800 firmware. This variable filter time takes the mechanical load into account with higher diameters.

The filtered motor speed will be scaled to actual velocity by parameter 50.17 WinderScale, whose signal comes from the diameter calculator.

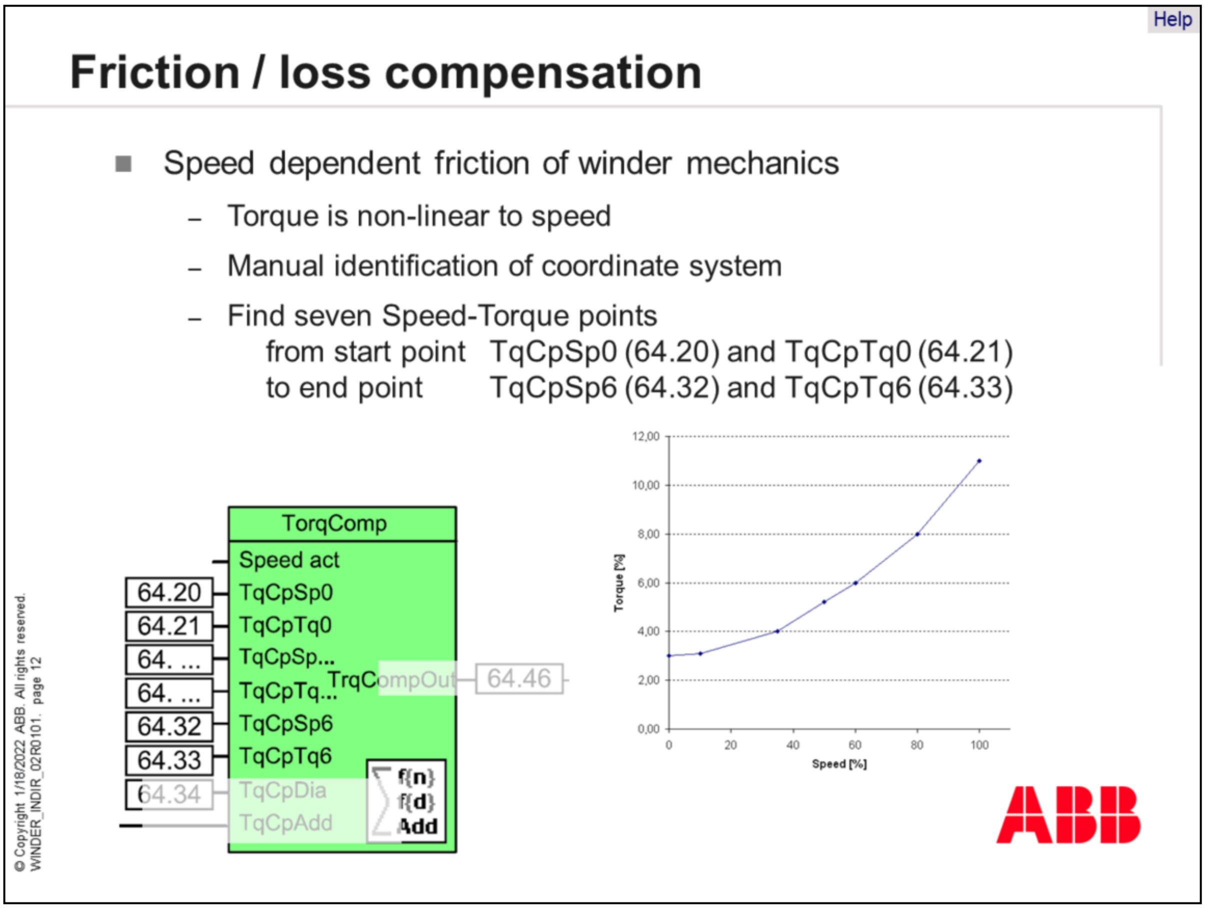
The diameter dependent filter time SpdFiltTime, parameter 61.13, takes the mechanical load into account with higher diameter.



This part is responsible for the compensation of the mechanical losses.

In principle, it is a feed-forward of a measured losses curve.

Without taking the losses into account, the actual tension will be lower as rewinder or higher as unwinder than the desired value.



The speed dependent friction / loss of winder mechanics, like motors, gear boxes, transmissions and cores, is not linear to the speed and so the torque for compensation is also non-linear.

Losses of a winding system can be manually measured and saved in a coordinate system in supporting points. The torque will be measured with an empty core in speed control mode and saved in the coordinate system. This curve will be pictured in a curve with seven supporting points.

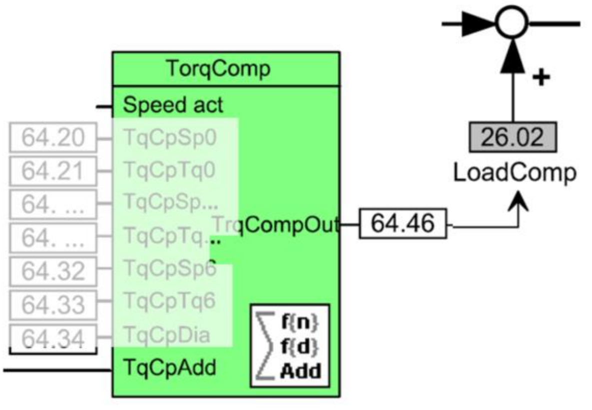
The starting point of the curve is index zero and the last point index 6. These values are connected directly with the function block "TorqComp". Speed zero must be saved in parameter 64.20 together with the starting torque, which is saved in parameter 64.20.

In the same way all other supporting points can be saved until the maximum speed parameter 64.32 with torque in parameter 64.33 is reached.

Help

Friction / loss compensation

- Diameter dependent friction via TqCpDia (64.34)
- Input TqCpAdd for summation of inertia compensation
- Output TrqCompOut contains the sum of friction and accelerating and decelerating compensations
- Transfering to LoadComp (26.02) of standard firmware.

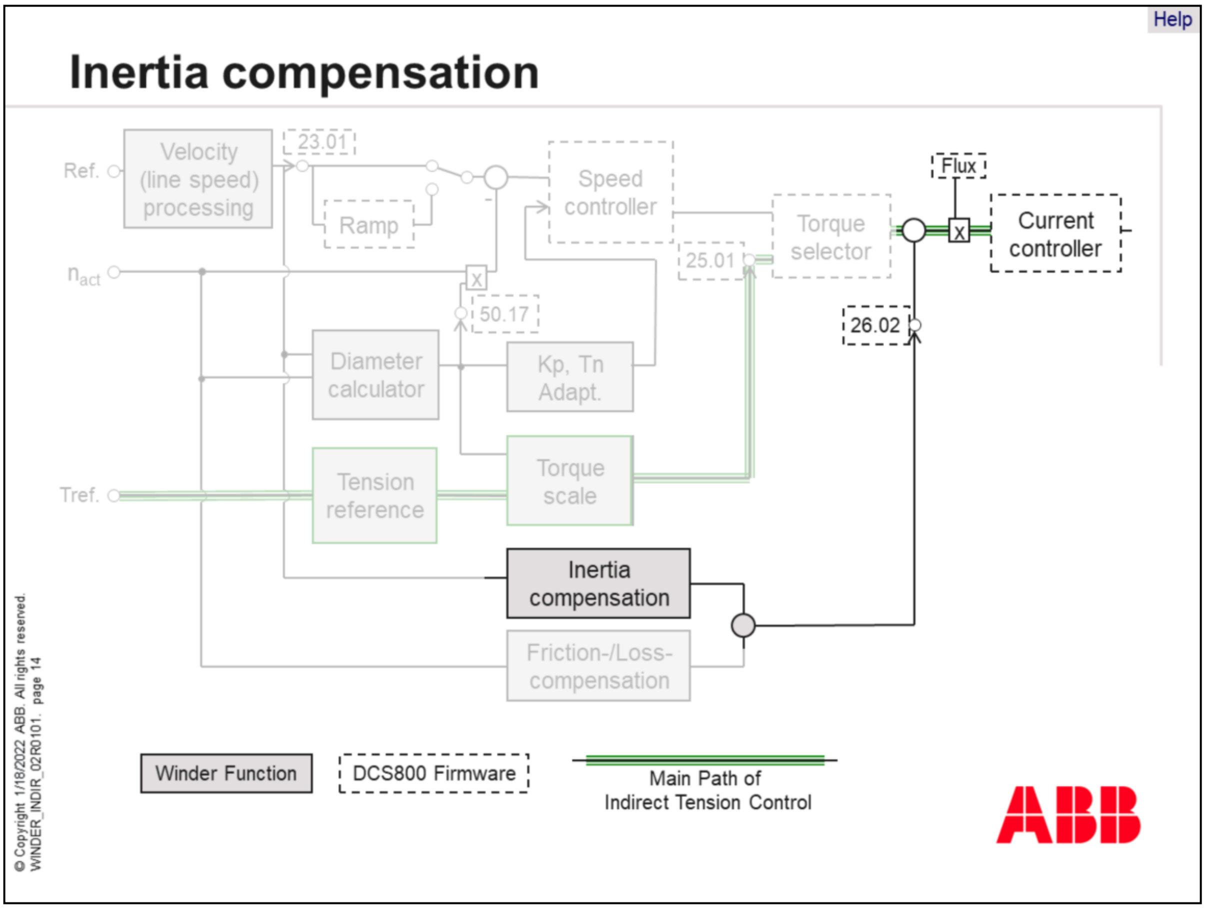


Diameter dependent friction can be set using TqCpDia, parameter 64.34, if needed.

The input TqCpAdd will get the summation of inertia compensation. More details of inertia can be seen and heard in the next chapter.

The output of this compensation block, TrqCompOut, contains the sum of friction and accelerating and decelerating compensations.

The complete value will be transferred to LoadComp (26.02) of the standard firmware.



In addition to the torque, which compensates the losses, the torque for the inertia compensation during accelerating and decelerating torque is needed.

Help

Inertia compensation

- Compensation of inertia while accelerating and decelerating
- A good derived velocity value is needed
- Inertia of the whole winder mechanics
- Inertia of increasing or decreasing coil

The material tension must be constant during acceleration and deceleration. Thus it is necessary to compensate the required torque of inertia.

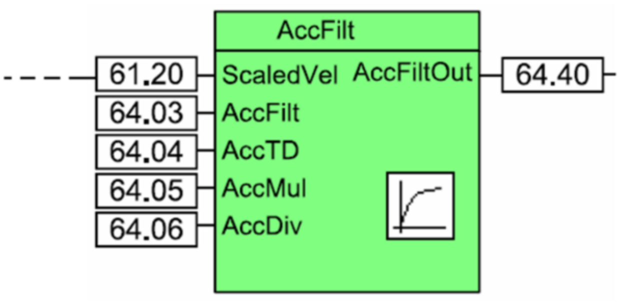
For this purpose, a very good acceleration signal dv / dt is needed.

Also all values of inertia for rotating mechanics and winding material are important.

Help

Inertia compensation

- Derivation of velocity (line speed)
  - Preferable untroubled value, no noise
  - Filtering via AccFilt (64.03)
  - Derivating via AccTd (64.04)
  - Adapting via AccMul (64.05) and AccDiv (64.06)
  - Alternative „fixed“ values from PLC are welcome and better
  - Shortest ramp time results highest output AccFiltOut



Derivation of velocity or line speed is necessary for calculating acceleration torque.

To calculate the derivation with function block "AccFilt", a speed reference signal free of disturbance is required. The derivation will be generated by using the filter of function block "AccFilt", parameter 64.03 and the differentiation "AccTd", parameter 64.04. The amplitude of a derived signal must have an integer value of 30000 if the incoming speed reference signal has the shortest ramp time.

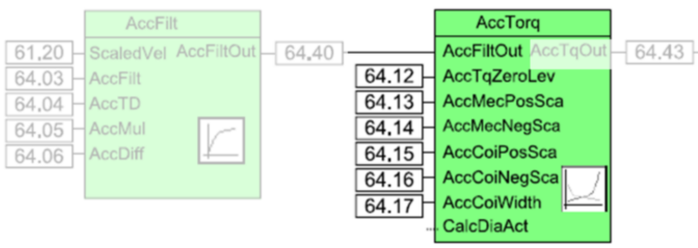
The shortest ramp time is the time in which the drive is still in winding mode. To scale the output signal of function block "AccFilt", parameters 64.05 and 64.06 can be used, which represents a multiplication and a division.

Alternative to the solution with a derived signal, there is also the option of using a fixed value with a ramp coming from a PLC. In this case function block "AccFilt" has to be switched off by setting parameter 64.04 to zero. Then the incoming value will be sent without any calculations to the output and can be used for the next function block.

Inertia compensation

- Compensation
  - Derived value and actual diameter
  - Value near zero is „deleted“ by using AccTqZeroLev (64.12)
- Calculation of mechanical inertia compensation
  - Inertia of winder mechanics
  - Detailed values of core, transmission, gear box and DC motor
  - Manual identification run for AccMecPosSca (64.13) and AccMecNegSca (64.14).

$$64.13 \equiv 64.14 \equiv \frac{Torque}{\Delta Speed} \cdot \frac{100\%}{Tmotor}$$



For compensation of acceleration torque, the derived signal coming from function block "AccFilt" is needed as well as the actual diameter.

Values near zero will be "deleted" by using parameter 64.12. Small values could be a disturbance but does not influence material tension.

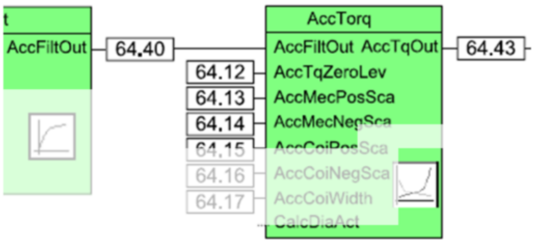
The calculation of mechanical inertia compensation is performed by function block "AccTorq". To do this, values of inertia from complete mechanics are needed for calculations. Oftentimes these values are not available because several mechanical devices are connected like the empty core, the gear box and the rotor of the dc machine. It is however possible to measure these values with an empty core in a small test. In this test, the motor torque will be measured during a constant ramp. This test must be performed for both directions.

Note: Parameters 64.13 and 64.14 represent the acceleration torque which is scaled for motor torque. The values correspond to an empty core which equals, so to speak, the smallest diameter.

Help

Inertia compensation

- Calculation of material (coil) inertia compensation
  - Min. diameter (Core) and actual diameter (CalcDiaAct)
  - Width via AccCoiWidth (64.17), only needed for coil torque compensation
  - Specific weight
  - Calculation for AccCoiPosSca (64.15) and AccCoiNegSca (64.16) (equation will follow)



Calculation of material or coil inertia compensation is the second functionality of the function block AccTorq.

Inertia which exists because of wound material can be calculated with the following values:

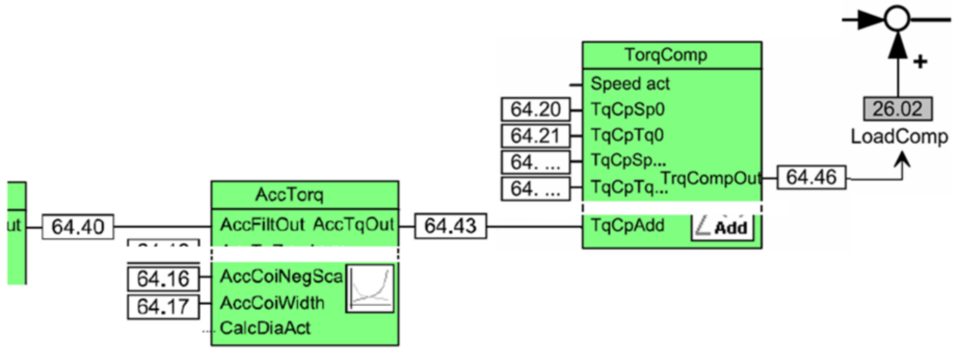
- The minimum diameter (from the core) and the actual diameter which comes from function block "CalcDiaAct"
- The width which must be set in percent by parameter 64.17 and
- The specific weight of the winding material

The calculated values will be set in parameter 64.15 and 64.16 in ratio to motor nominal torque. In these calculations the maximum diameter will be taken into account. The formula will be explained at the end of this module!

Help

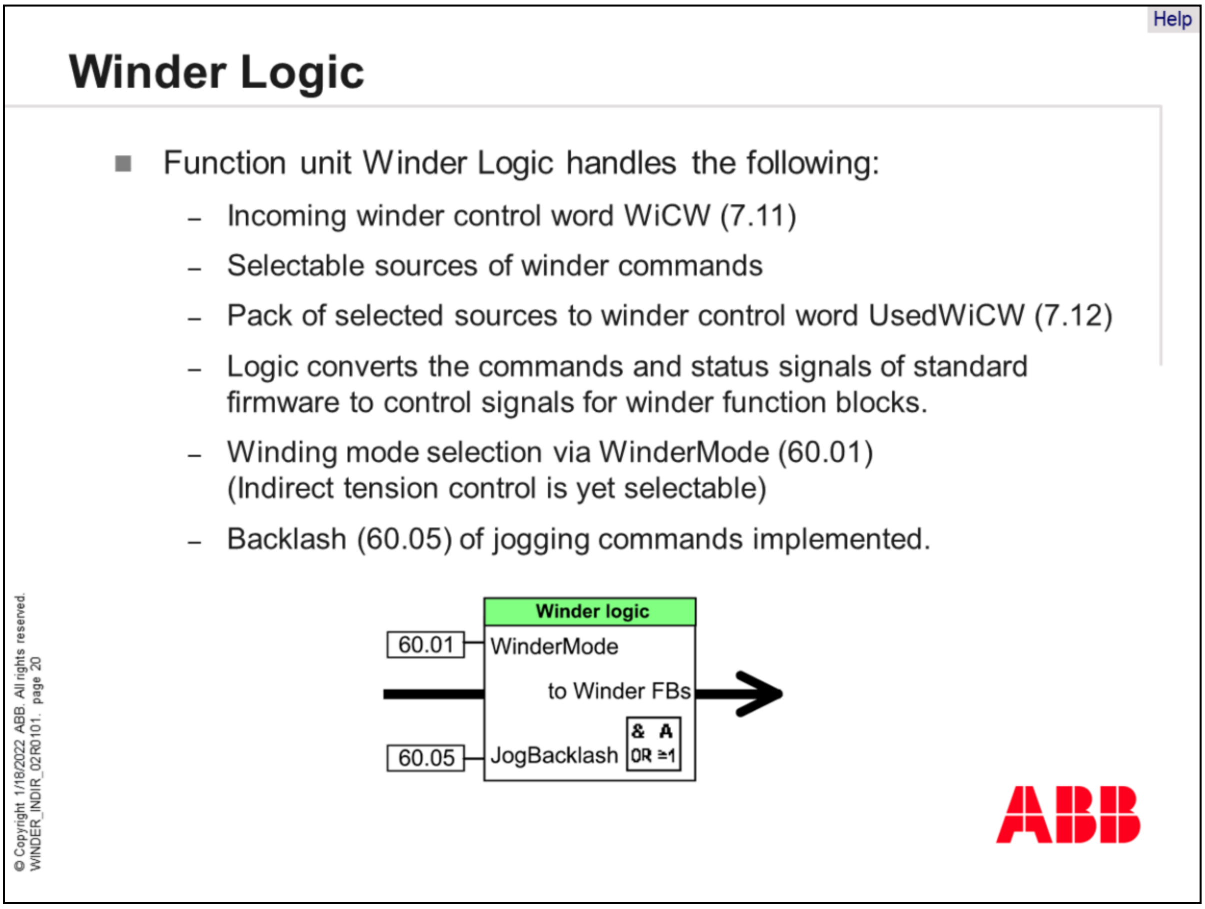
Inertia compensation

- Summary of compensation parts to output AccTqOut, parameter 64.43
  - Accelerating compensation of winder mechanics inertia
  - Accelerating compensation of coil inertia
- Connection via TorqComp to LoadComp (26.02)



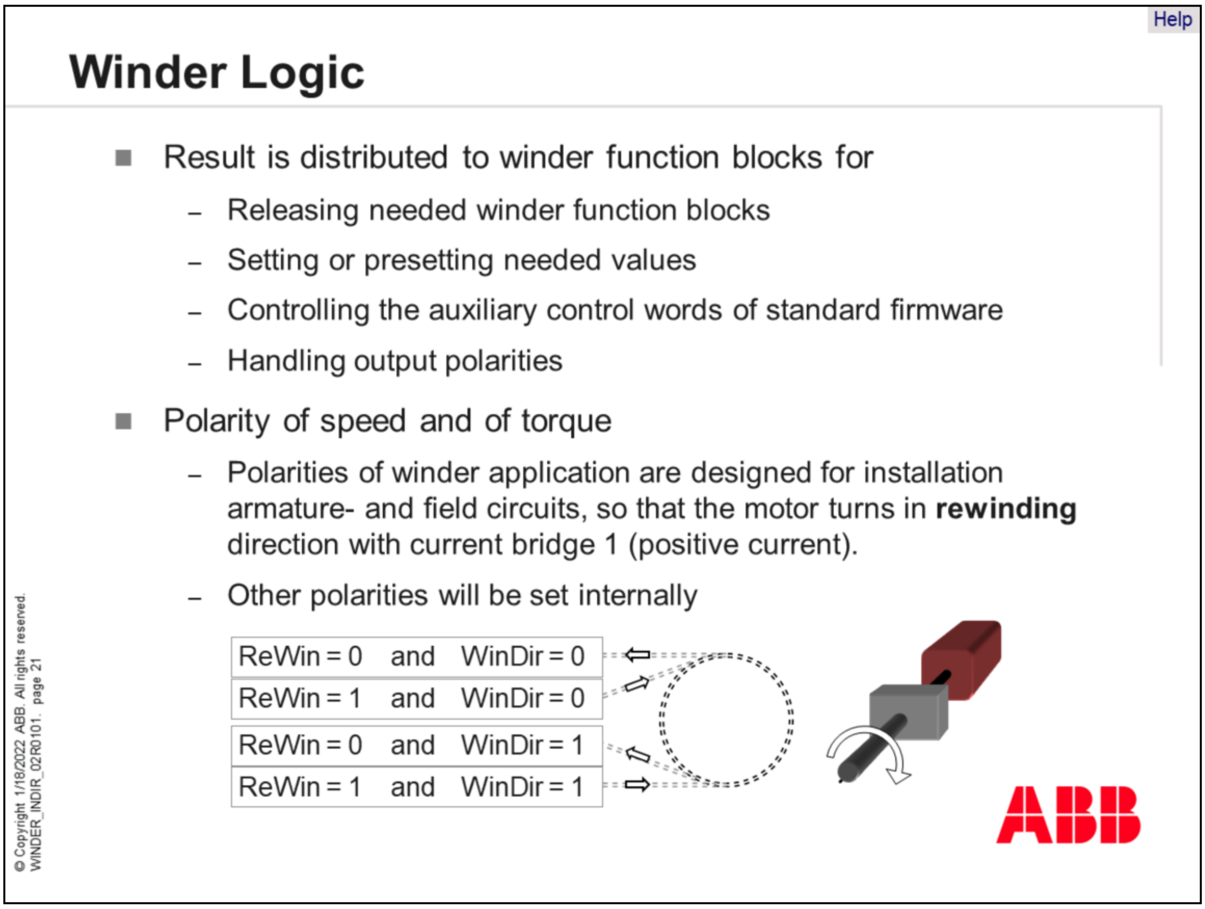
The summary of compensation parts is shown with the output of function block "AccTorq" and in parameter 64.43.

This includes the mechanical and the coil compensation. The result will be connected to function block "TorqComp" which adds further losses and writes the sum to parameter 26.02 in the system software. This affects the torque reference of the winder.



The function unit Winder Logic handles the following:

- Incoming winder control word WiCW (7.11), which can be written by using an overriding system (e.g. PLC).
- Each winder command, which can also be selected out of digital inputs or other bits of main control word.
- The selected sources, which are packed to the used winder control word UsedWiCW (7.12)
- The logic, which converts the commands and some status signals of standard firmware to the control signals for each winder function block.
- Winding mode selection via WinderMode (60.01)
  (Indirect tension control can be selected)
- The implemented Backlash (60.05) jogging commands.
  After resetting an active jog command it needs the Backlash time to release the external velocity reference again.

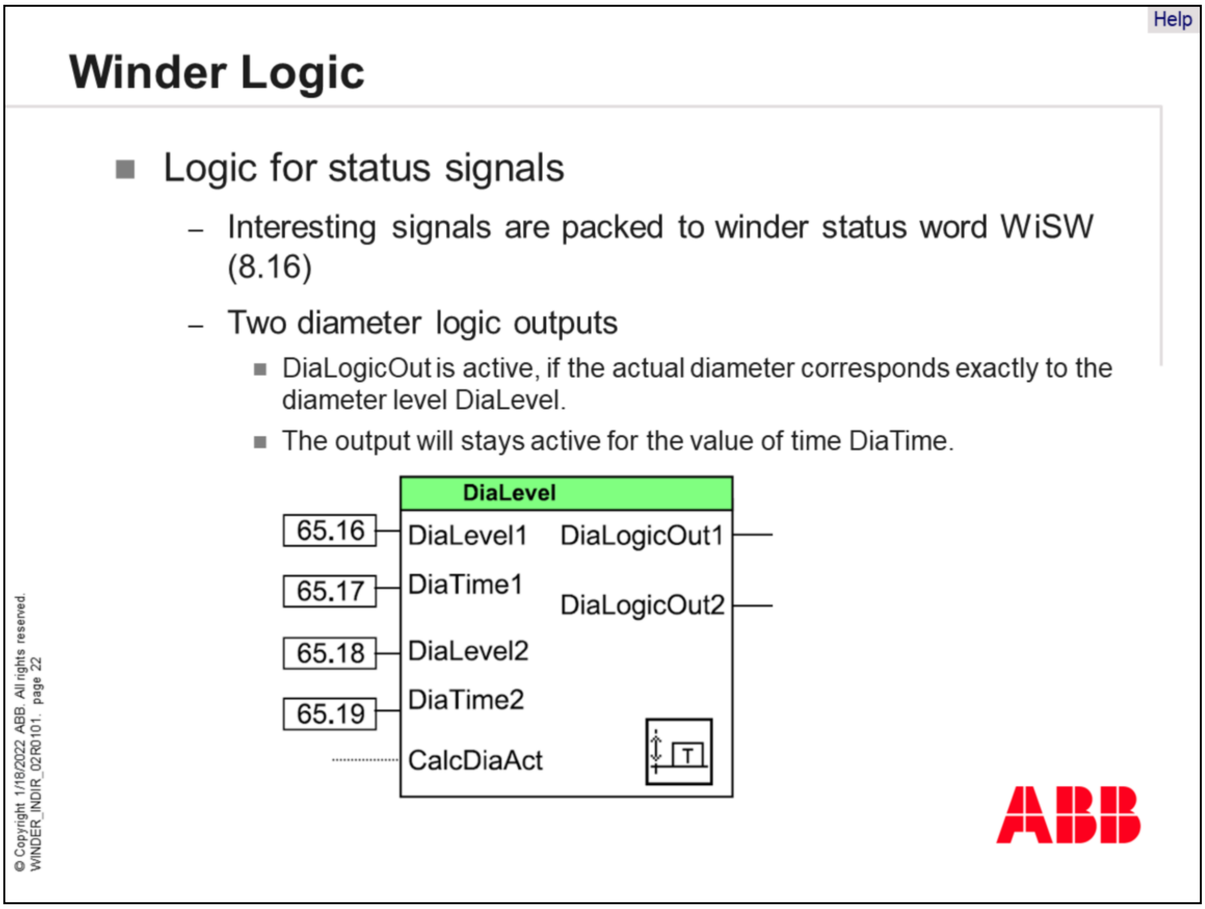


The results are distributed to winder function blocks for

- Releasing needed winder function blocks
- Setting or presetting needed values
- Controlling the auxiliary words of standard firmware for speed-torque change over, window control and limitations
- Handling output polarities

Polarity of speed and of torque

- The polarities of the winder application are designed for an installation of armature- and field circuits, so that the motor turns in **rewinding** direction with current bridge 1 (positive current).
- Other polarities will be set internally according to command ReWin (for rewinding or unwinding) and to command WinDir (for winding direction).



Important signals out of the winder application are packed to winder status word WiSW, parameter 8.16. The winder status word also includes two diameter logic outputs.

The output DiaLogicOut is active if the actual diameter CalcDiaAct corresponds exactly to the diameter level DiaLevel, see parameter 65.16 DiaLevel1 and 65.18 DiaLevel2.

The output will stay active for the value of time DiaTime, see parameter 65.17 DiaTime1 and 65.19 DiaTime2.

Help

Significant equations

- Diameter

$$65.01 = \frac{D_{\min}}{D_{\max}} * 100\% = \_\_\_\_\_\_\_\_\_\_$$

- DiaRamp (first values!)

$$t_{Ramp} < \frac{D_{\max}^{2} \cdot \pi}{v \cdot 2\delta} =$$

(if $D_{max} >> \delta$ (thickness))

$$65.04 \approx 0.5 \cdot t_{Ramp} = \_\_\_\_\_\_\_\_\_\_$$

$$65.05 \approx 0.8 \cdot t_{Ramp} = \_\_\_\_\_\_\_\_\_\_$$

- Torque Scaling

$$66.03 = \frac{F_{\max} \cdot D_{\max}}{2 \cdot i} \cdot \frac{100\%}{T_{motor}} = \_\_\_\_\_\_\_\_\_\_$$

WINDER_INDIR_02R0101. page 23

The next pages will show significant equations for calculating parameter values.

- The diameter ratio is used as minimum diameter, as value of 65.01 MinCoreDia.
- The ramp time of the diameter calculator is used as a guiding value of parameter 65.04 TrampWDir and 65.05 TrampWcDir. It should be clear that the time should not be too long to compensate for a wrong start setting, for example.
- The percental torque scaling in relation to the nominal motor torque as parameter 66.03 TorqScale.

Help

Significant equations

- Accel- and decelerating torque of coil inertia

$$T_{coil} = \frac{\rho \cdot W \cdot \pi}{D_{max} \cdot 16} \cdot \left(D_{max}{}^4 - D_{min}{}^4\right) \cdot \frac{1}{i} \cdot \frac{dv}{dt}$$

Velocity	$[dv] = \frac{m}{s}$	Strip Density	$[\rho] = \frac{kg}{m^3}$	Acceleration time	$[dt] = s$
Diameter	$[D] = m$	Web Width	$[W] = m$	Gear ratio	$[i] = ..$

$$64.15 \equiv \frac{T_{coil}}{T_{motor}} \cdot 100\% = \underline{\qquad\qquad}$$

$$64.16 \equiv 64.15$$

The next step is to calculate the percental acceleration and deceleration torque of coil inertia in relation to nominal motor torque. First, calculate the physical acceleration torque by using the physical parameters of the plant. Then calculate the percental acceleration and deceleration torque for parameter 64.15 and 64.16.

Help

Summary

Key points of this module

- Velocity control part of the winder application
- Friction and loss compensation
- Inertia calculation
- Calculating the basic parameter values

Here are the key points of this module.

- Velocity control of the winder application
- Friction and loss compensation
- Inertia calculation
- Calculating the basic parameter values

Help

Glossary

- **Dancer roll**
  swinging roll for balancing
- **Lead rolls**
  rolls to control the line speed
- **Load cell**
  roll to measure the tension
- **Core**
  metal shaft in the middle of the roll
- **Unwinder**
  winding up
- **Rewinder**
  winding
- **Acceleration torque**
  motor torque to accelerate the plant
- **Torque losses**
  speed dependent losses cause of coupled mechanic (non linear)

Power and productivity
for a better world™