AMKmotion simulation models Function blocks for simulation with MATLAB® / Simulink®

Version: 2023/26

Part no.: 204310

Translation of the "Original Dokumentation"

AMKmotion

MEMBER OF THE ARBURG FAMILY

Imprint

Name: PDK_204310_Simulation_KW-R06

Version:

Version	Change	Letter symbol
2023/26	AMKmotion Design	LeS

Previous version: 2018/07

Product status:

Product	Firmware Version (Part no.)	Hardware Version
AMK Simulation	SimKW V1.00 2012/46 (204313)	-

Publisher: AMKmotion GmbH + Co KG
Gaußstraße 37-39
73230 Kirchheim unter Teck
Germany

Phone +49 7021 50 05-0
Fax +49 7021 50 05-176

E-mail: info@amk-motion.com

Registration court: AG Stuttgart, HRA 230681, Kirchheim unter Teck,
Tax Id no.: DE 145 912 804

Complementary: AMKmotion Verwaltungsgesellschaft mbH, HRB 774646

Service: Phone +49 7021 50 05-190, Fax -193

For fast and reliable troubleshooting, you can help us by informing our Customer Service about the following:

- Type plate data for each unit
- Software version
- Device configuration and application
- Type of fault/problem and suspected cause
- Diagnostic messages (error messages)

E-mail service@amk-motion.com

Internet address: www.amk-motion.com

Content

1 About this documentation

1.1 Storage

This document must permanently be available and readable at the place where the product is in use. If the product is used at another place or changed the owner, the document must be passed on.

The software licensee must store the documentation and make it available to the user.

1.2 Target group

Any person that is qualified and intends to work with this product must read, understand and follow this document:

- Projecting
- Startup
- Service and repair

A knowledge of how to work with the simulation program MATLAB® / Simulink® is a prerequisite.

1.3 Objectives

This document describes the function blocks 'SimKW' and 'SimMot', which can simulate drives under MATLAB® / Simulink®.

- The procedure for installation is described.
- The application of the function blocks is shown using an example simulation.
- The interfaces of the function blocks are described.
- Possible operating errors and error responses are covered.

1.4 Validity

This documentation describes the following function blocks:

- SimKW; based on the control software AER5-6 V1.10 2013/15 (204486)
- SimMotor

1.5 Display conventions

Display	Meaning
!	This symbol points to parts of the text to which particular attention should be paid!
0x	0x followed by a hexadecimal number, e. g. 0x500A
'Names'	Names are represented with apostrophes e. g. parameters, variables, etc.
'Text'	Menu items and buttons in a software or on a controller, e. g.: Click the **'OK'** button in the **'Options'** menu to call up the 'Delete PLC program' function
>xxx<	Placeholder, variables, e. g. IP address of the controller: >192.168.0.1<
See 'chapter name' on page x	Executable cross-reference in electronic output media
Blue text	Executable link in electronic output media

1.6 Appendant documents

Functional documentations

AMK part-no.	Title
25786	Diagnostic messages
202234	Software description AIPEX PRO (PC software for startup and parameterization)
203704	Parameter description KW-R06 / -R16 / -R07 / -R17 (properties of controller parameters)
203878	Function descriptions (functions of the controller firmware)

2 Introduction

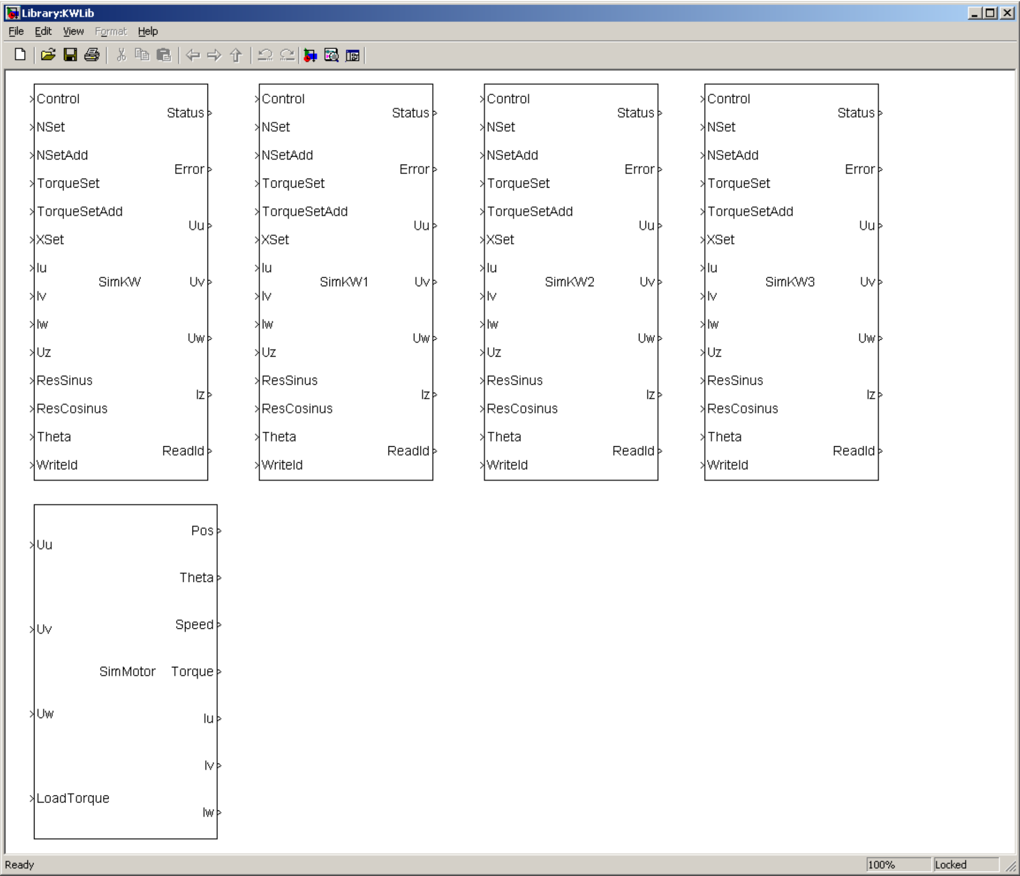
2.1 Product description

The software package allows the simulation of a servo drive comprising an AMK compact inverter with controller card and a permanently energized synchronous motor under MATLAB® / Simulink®. The control core is simulated; the simulation does not include fieldbus or safety functions.

A library containing function blocks that can be inserted in Simulink and linked to each other and to further elements is provided.

This makes it possible to simulate the behavior of a drive that has not yet been realized, also in combination with a machine model, to assess the effects of parameter changes or examine sequences that are difficult to recreate in reality.
It must be noted, however, that these are idealized models that cannot simulate all influences occurring in reality precisely.

Simulink library 'KWLib'



Function block SimKW

The function block SimKW simulates the behavior of a compact inverter with the controller card. This includes the control software and the IGBT bridge for output of the phase voltages. The original software source texts as they are used in the controller have been included.

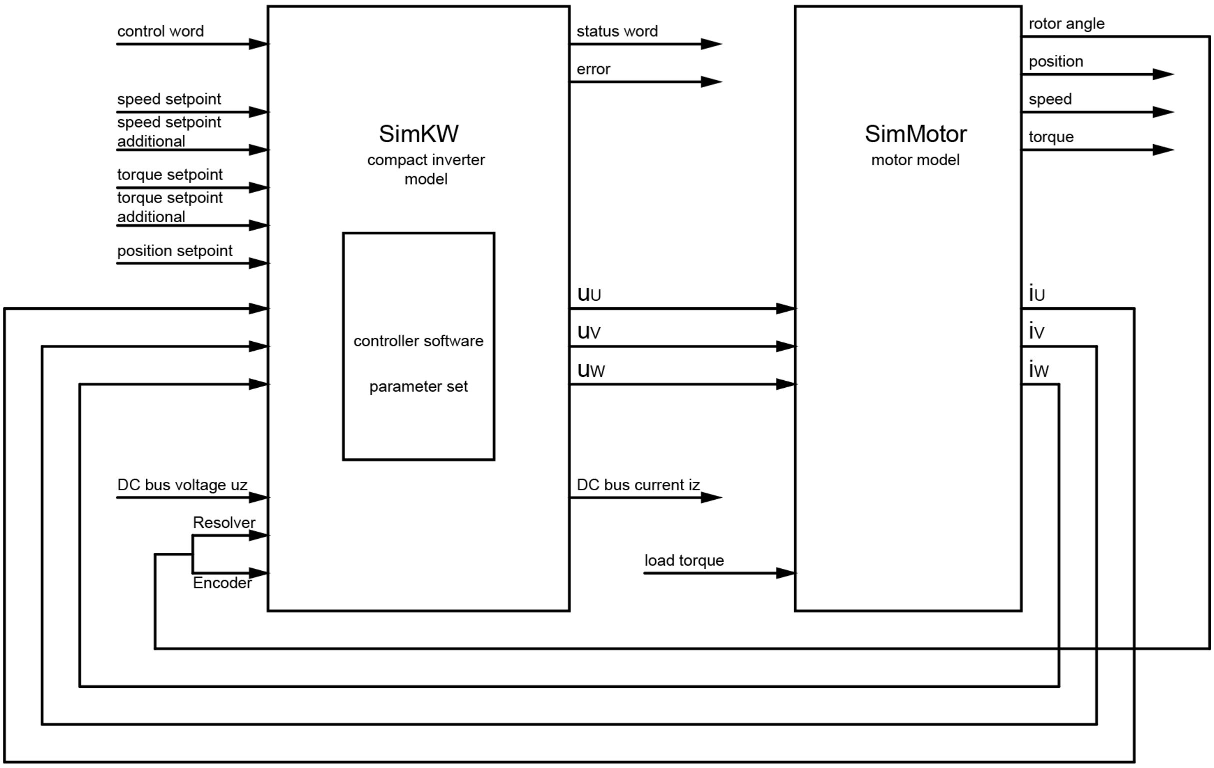
Function block SimMotor

The function block SimMotor simulates a permanently energized synchronous motor.

2.2 Control principle

The diagram below shows the principle of the signal flow between the function blocks SimKW and SimMotor.

Diagram of signal flow between 'SimKW' and 'SimMotor'



The setpoints for speed, torque or position are defined for SimKW. SimKW also requires a value for the DC bus voltage.

The control procedure is started with the control word.

The motor returns the rotor angle, on the basis of which the inverter determines actual values of position and speed .

The phase currents are used to calculate the actual torque value.

Based on the controller deviation between setpoint and actual values, SimKW determines the phase voltages that need to be applied to the motor in order to achieve the setpoints.

3 Installation of the simulation blocks

3.1 PC Requirements

PC with Windows XP

Free hard disk space 2.8 MB

AIPEX PRO from Version 1.08 (203927) with SP05 (204264)

MATLAB/Simulink from R2008b

3.2 Software package

The AMK simulation software package contains the following files:

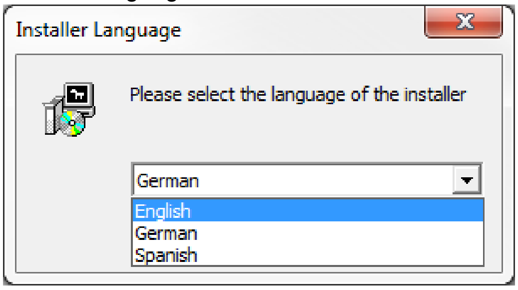
File	Description
Installation / Deinstallation	
SetupSimKW_100.exe	Installation archive
uninstall.exe	Removes all files and registry entries
Simulink library	
KWLib.mdl	Library with the blocks SimKWand SimMotor
Function blocks for controller simulation	
SimKW.mexw32	Executable file for the block SimKW Siehe 'Function block SimKW' auf Seite 15.
R06Sim.dll	Is required by SimKW
SimImportParam.dll	Is required by SimKW
SimKW1.mexw32	Executable file for the block SimKW1
R06Sim1.dll	Is required by SimKW1
SimKW2.mexw32	Executable file for the block SimKW2
R06Sim2.dll	Is required by SimKW2
SimKW3.mexw32	Executable file for the block SimKW3
R06Sim3.dll	Is required by SimKW3
Function block for motor simulation	
SimMotor.mexw32	Executable file for the block SimMotor Siehe 'Function block SimMotor' auf Seite 25.
LookUp.dll	Is required by SimMotor
MCalc.dll	Is required by SimMotor
Simulation example	
SimKWTest.mdl	Example project under MATLAB/Simulink
KW2_DT4_1_10.aipex	AIPEX PRO project: Compact inverter KW2 with controller card KW-R06 and motor DT4-1-10
KW2_DT4_1_10.xml	Exported parameter set
KW60_DT7_72_20.aipex	AIPEX PRO project: Compact inverter KW60 with controller card KW-R06 and motor DT7-72-20
MCE200_DS13_170_6.aipex	AIPEX PRO project: MCE200 with controller card MCE-R06 and motor DS13-170-6
Motoren.txt	Motor description file

3.3 Installing the software

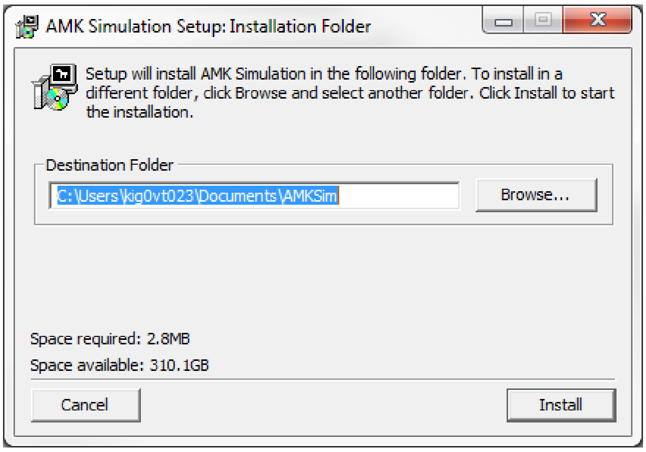
The software package is installed from the file SetupSimKW_100.exe.

1. Start SetupSimKW_100.exe

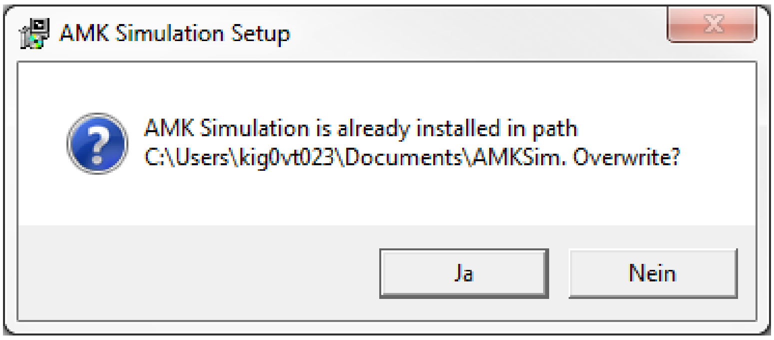
2. Select a language for the installer:



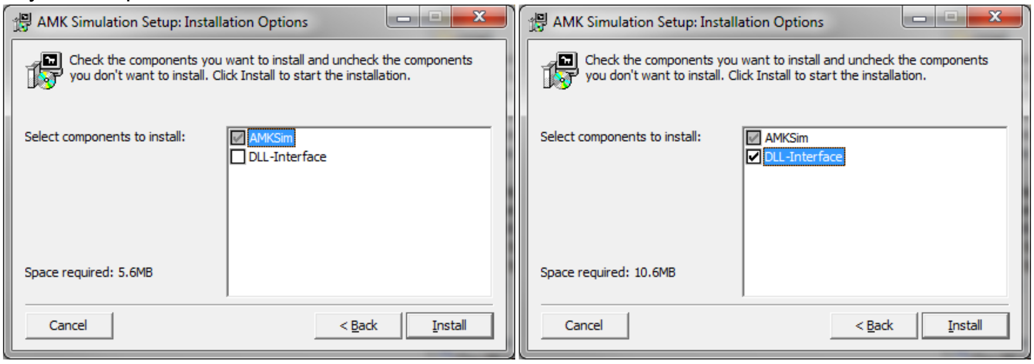
3. Set the target folder and click **'Install'** to start the installation:
   As the target folder, select the path that will be used as the MATLAB working directory in which the Simulink models are created and executed.



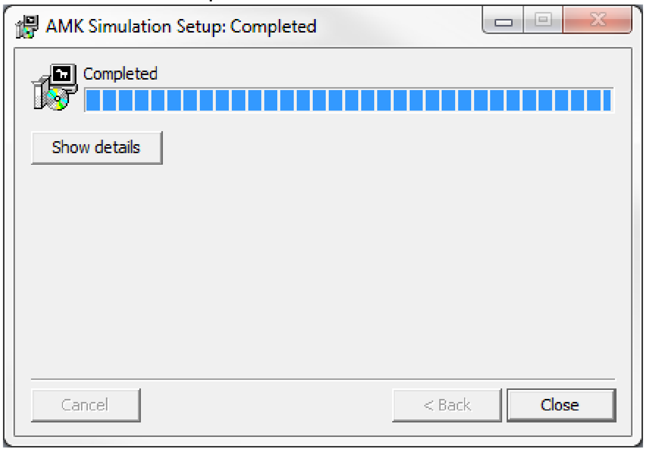
4. If files from the AMK Simulation package are already installed in the selected path, you will be asked if you wish to overwrite these files:



   Confirm with Yes or cancel the procedure.

5. Installation option
The simulation of the inverter, motor and power supply is carried out with the help of functions in various DLLs. Instead of using the Simulink blocks, the functions can also be called directly from your own simulation software. For this you need the 'DLL-Interface'. With the installation of the DLL interface, the headers for integration into your own C code and a sample project 'DriveSimTest.cbp' for the Code :: Blocks development environment are made available. The example project simulates all three components in the same way as the Simulink-example 'SimKWTest.mdl' and outputs the recorded data in columns to a text file. Select whether the additional files for using the DLL interface should also be copied to your computer:


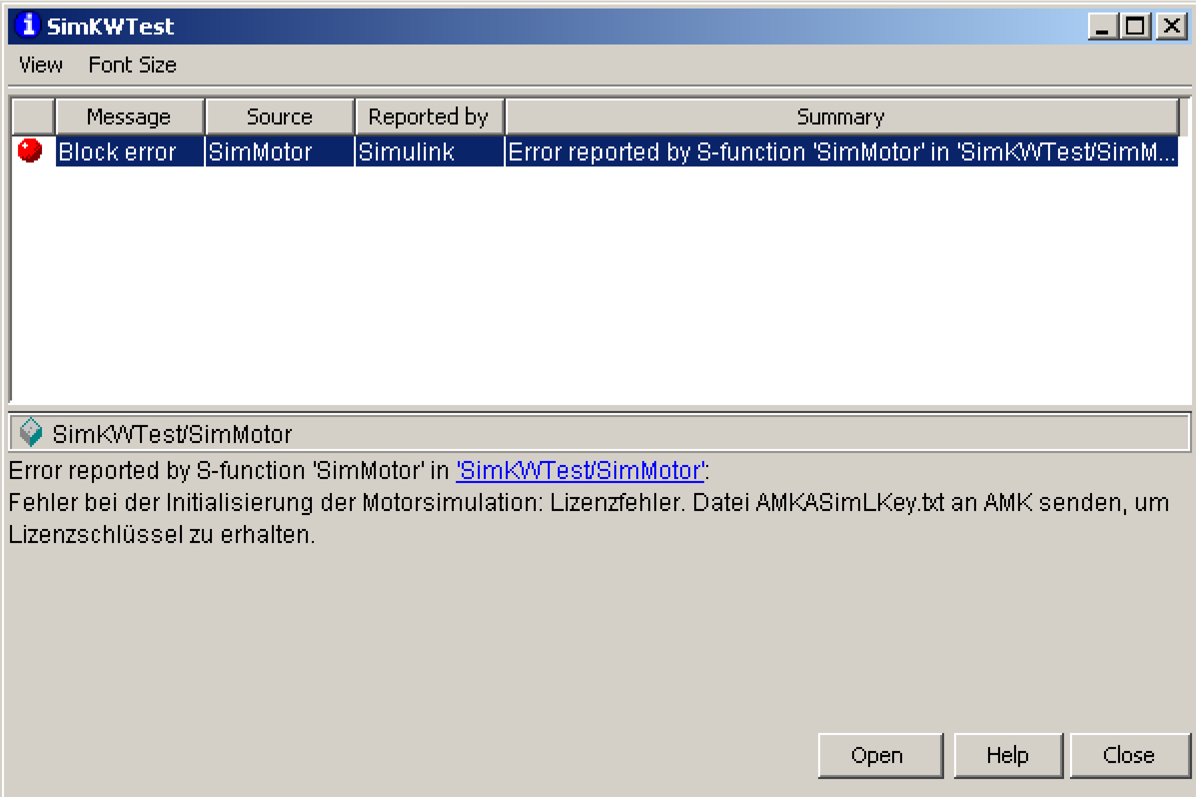
6. The files from the software package are copied to the selected target folder

7. Click ‘Close’ to complete installation.


3.4 License key

The simulation model can be created without a license key. The license file is only required when starting the first simulation, also for the example project.

If no license key is available, the simulation is aborted with the following messag:



A file with the name 'AMKASimLKey.txt' is created in the simulation folder. Please send this file to your AMK account manager.

You will receive a license key in the form of file 'AMKASim.key'. This file is copied to the Windows \ System32 directory on the PC from which the request was generated.

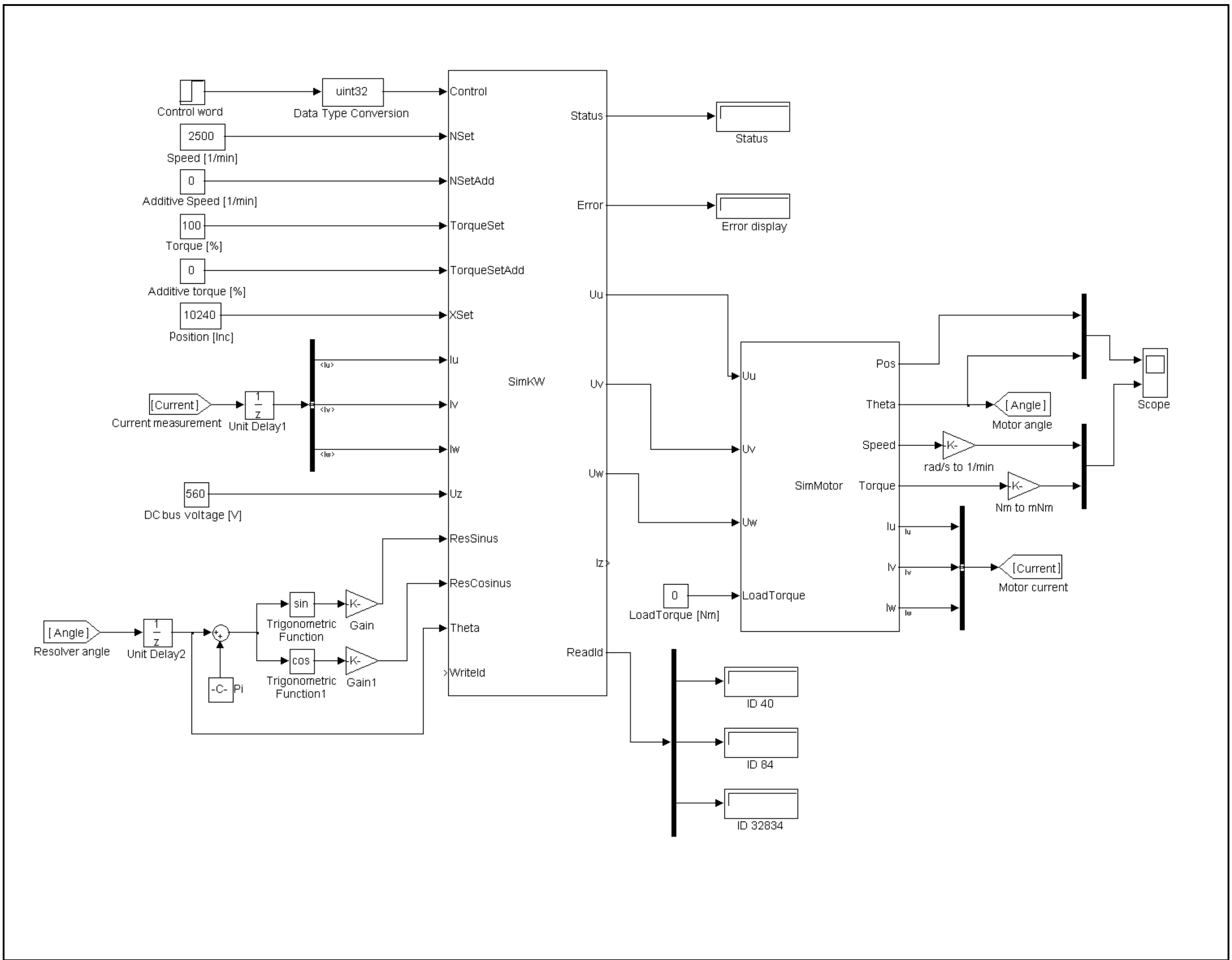
The software can be used once the license key has been copied.

The license key is only valid for the PC on which the request file was generated.

4 Example project ‘SimKWTest’

The software package ‘AMK Simulation’ is supplied with the example project ‘SimKWTest.’ It is an example of the application of the both blocks SimKW and SimMotor and is based on the AIPEX PRO project KW2_DT4_1_10.aipex.

Example project SimKWTest



This project was planned in such a way that the connection of the function blocks allows processing of any encoder type and any operating mode.

The rotor position can be transmitted either via a simple resolver simulation as a sine and cosine signal or directly as the angular value of an incremental encoder.

The motor current is fed back from the simulation of the motor to the inputs of the KW.

The setpoints and DC bus voltage are defined as constants.

In the project example KW2_DT4_1_10.aipex, speed control is selected as the operating mode. The setpoint speed is 2500 rpm. The remaining setpoints for position and torque are not active.

The rotor position is returned with an S encoder, so that the actual position value is evaluated via the SimKW input Theta. The resolver inputs are not active.

SimKW calculates the speed based on the change in the actual position value.

For the feedback of the currents and the angle, it must be noted that only the values from the last cycle are made available. For this reason, a delay element (1/z) is required in each case.

4.1 Setting

A sampling time of 1 µs is specified for the two blocks. This time must be entered as fixed sampling time in the settings for the model:

Menu Simulation -> Configuration Parameters -> Solver:

- Type: Fixed-step
- Solver: Discrete
- Fixed-step size: 0.000001

4.2 Angle feedback

The function block ‘SimMotor’ provides the rotor angle as a radian measure [–π..π]. The value can be returned to ‘SimKW’ via the Theta input or the resolver inputs.

Angle feedback via ‘Theta’ input

An incremental encoder (E- / F- / I- / P- / Q- / S- / T- / square-wave pulse encoder) must be set in ID32953 'Encoder type'.

In the case of I-encoders and square-wave pulse encoders, the rotor is aligned after the first setting of the controller enable. With the other encoder types, the transmitted angle is immediately applied as an absolute value.

Angle feedback via resolver inputs

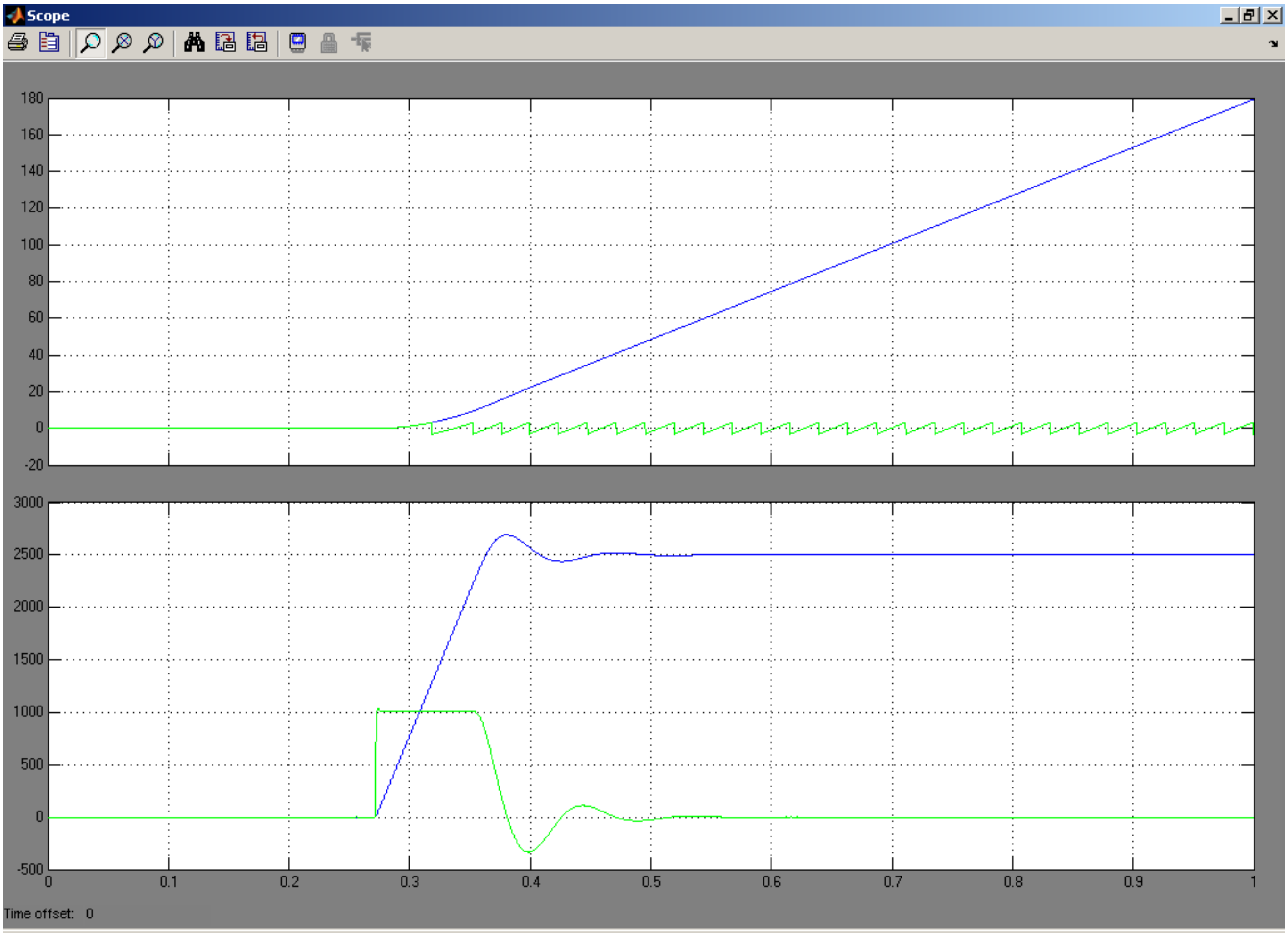
Resolver must be set in ID32953 'Encoder type'.

The radian measure of the rotor angle is split into its sine and cosine proportion and returned to the inputs ‘ResSinus’ and ‘ResCosinus’ with an amplitude of 2.5 V.

For the feedback of the angle, it must be noted that the values from the last cycle are processed in each case. For this reason, a delay element (1/z) is required in each case.

4.3 Simulation result

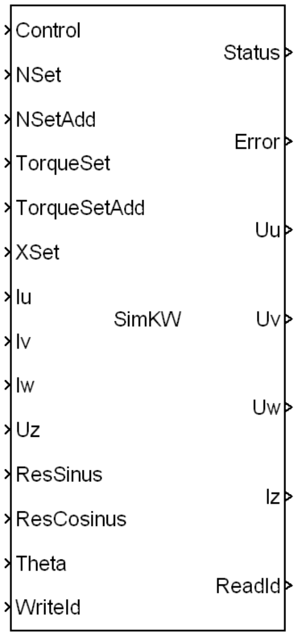
Recorded values from simulation of the model: speed control



- Top diagram:
  - blue: position [rad]
  - green: rotor angle [rad]
- Bottom diagram:
  - blue: speed [rpm]
  - green: torque [0.001 Nm]

5 Function block SimKW

Depiction of the function block in MATLAB/Simulink



The function block SimKW simulates the behavior of a compact inverter with the controller card. This includes the control software and the IGBT bridge for output of the phase voltages. The original software source texts as they are used in the controller have been included.
Control word, voltage supply, setpoints and the actual values are transmitted from the motor to SimKW as input values.
Output values are the phase voltages, status information, configurable oscilloscope data and the DC bus current.

The following simplifications have been made in comparison to the real device:

- The following functions are not part of the controller simulation:
  - fieldbus connection
  - analog inputs
  - binary inputs and outputs
- Setpoints are entered directly as floating point values
- Control commands are specified at the input as bits in the control word
- Phase voltages:
  Not the actual voltage switched by the PWM is output, but an average value across every PWM period. Otherwise, a shorter sampling time would have to be selected, which increases the time required for calculation unnecessarily.

If effects in connection with the setpoint specification via a bus system are important for the simulation result, these boundary conditions must be modeled externally.

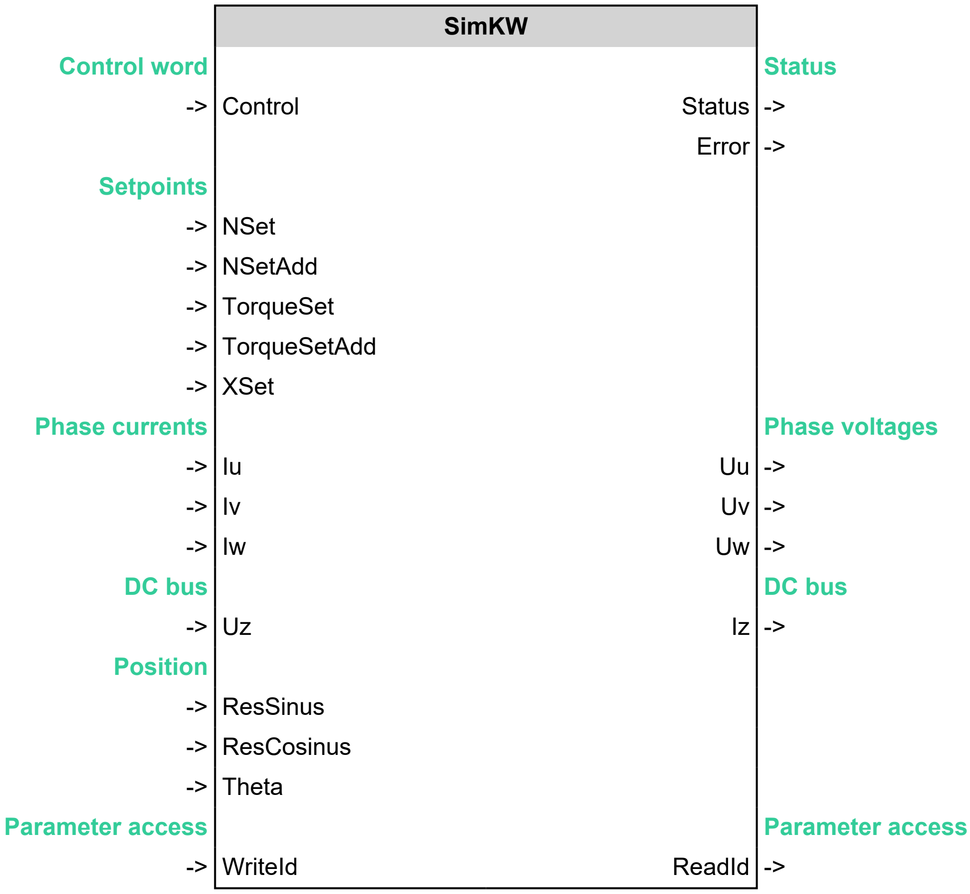
The drive parameters are exported directly from an AIPEX PRO project and read in by the function block SimKW. This means that a parameter set from a real drive can be used very quickly for the simulation. Vice versa, a parameter set determined in the simulation can be uploaded to a drive.

The same function block may not be used multiple times in a Simulink model.
To enable simulation of up to four drives, the library also offers the blocks SimKW1, SimKW2 and SimKW3, which correspond to SimKW.

5.1 Interface of the SimKW function block

The illustration below shows the SimKW function block in detail, comparing the functionally related inputs and outputs.

User interface



5.2 Inputs of the function block SimKW

5.2.1 Control word

Input variables

Name	Design	Unit	Description	Drive parameters *)
Control	UINT32	-	Control word	-

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

In order to start the control, the ‘Controller enable’ command must be set at the ‘Control’ input during the simulation time via the control word, bit 0. The status of the controller is checked internally and set to free of error status QRF. Control is started as soon as ‘Acknowledgment controller enable’ (QRF) = 1.

The control word must be created at the 'Control' input with data type UINT32. If, as in the example, a step block that outputs a value of type DOUBLE is used, the data type must be converted.

In order to generate the 0 -> 1 edge that activates the controller enable in bit 0 during the simulation time, a time delay can, for example, be applied to the step block.

Control word

Bit	State	Description
0	0	Revoke controller enable
	1	Activate controller enable (RF)
1	0 -> 1	Clear error command
2 - 31		Reserved

5.2.2 Setpoints

Input variables

Name	Design	Unit	Description	Drive parameters *)
NSet	DOUBLE	rpm	Speed setpoint	ID36
NSetAdd	DOUBLE	rpm	Additive speed setpoint	ID37
TorqueSet	DOUBLE	% M_N	Torque setpoint	ID80
TorqueSetAdd	DOUBLE	% M_N	Additive torque setpoint	ID81
XSet	DOUBLE	Incr	Position setpoint	ID47

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

Depending on the operating mode selected in ID32800 'AMK main operating mode', the corresponding setpoint inputs are active:

Active setpoints, depending on the operating mode

ID32800	Meaning	Active setpoints
0x003C0043	Speed control Setpoint ramps for acceleration and deceleration active	Speed setpoint Speed setpoint, additive
0x003C0002	torque control Torque limitation via ID82 / ID83 active	Torque setpoint Torque setpoint, additive
0x003C0004	Position control Actual position source motor encoder	Position setpoint

This means that setpoints can be entered at all inputs. With the start of the control procedure (control word controller enable = 1), the setpoints for the selected operating mode become active.

5.2.3 Phase currents

Input variables

Name	Design	Unit	Description	Drive parameters *)
Iu	DOUBLE	A	Measured phase current Phase U	ID32828
Iv	DOUBLE	A	Measured phase current Phase V	ID32829
Iw	DOUBLE	A	Measured phase current Phase W	ID32830

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The phase currents I_u, I_v, I_w calculated and output by the motor simulation are returned to SimKW as inputs.
A time delay z^{-1} must be set for these feedback values.

Observe the phase sequence when feeding back the currents!

5.2.4 DC bus

Input variables

Name	Design	Unit	Description	Drive parameters *)
Uz	DOUBLE	V	DC bus voltage	ID32836

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

At input Uz, the DC bus voltage is entered as a value of type DOUBLE.

Possible errors:

1. DC bus voltage below 'DC bus voltage monitoring' entered
   - Entry
     Uz < ID32837
   - Error response
     Error = 1049 'DC bus'
     Controller is not enabled (QRF = 0)'

5.2.5 Position

Input variables

Name	Design	Unit	Description	Drive parameters *)
ResSinus	DOUBLE	V	Sine signal from resolver	ID51
ResCosinus	DOUBLE	V	Cosine signal from resolver	ID51
Theta	DOUBLE	rad	Mechanical rotor position (Value range -π ... π)	ID51

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

ID32953 'Encoder type' determines the motor encoder type. Depending on this encoder type, the actual position value is either processed as sine / cosine track of a resolver or as the angle ‘Theta’ of an incremental encoder.

ID32953 'Encoder type'

0x	00	x	x

| L Encoder type
| 2 T-encoder
| 5 I-encoder
| 7 S-encoder
| 8 Resolver
| 9 square-wave pulse encoder
| A E resp. F encoder
| C P- / Q-encoder

L Field weakening
1 Synchronous motor not field-weakening
3 Synchronous motor field-weakening

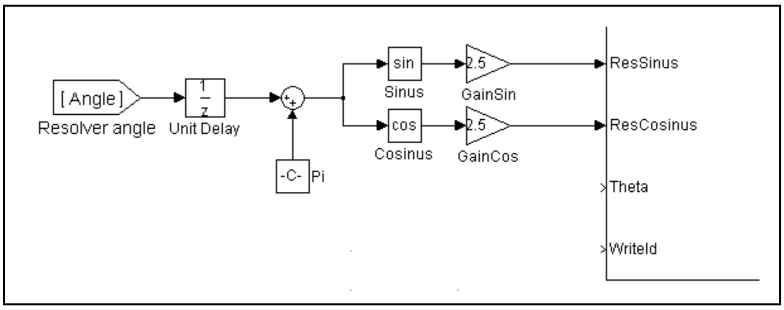
Resolver

If a resolver is selected as the encoder type, the motor angle must be split into a sine and a cosine proportion. ResSinus and ResCosinus simulate an analog resolver input.

A time delay z^{-1} must be set for the returned motor angle.

In addition, a multiplied gain simulating the voltage level of the resolver is added in each case.

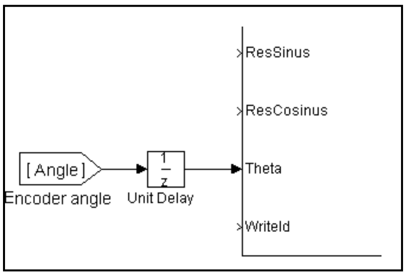
Connection of the inputs ResSinus and ResCosinus



Incremental encoder (E- / F- / I- / P- / Q- / S- / T- / square-wave pulse encoder)

A time delay z^{-1} must be set for the returned motor angle.
Further conversions are not necessary; the value can be transmitted directly to the 'Theta' input.

Connection of the Theta input



5.2.6 Parameter access

Input variables

Name	Design	Unit	Description	Drive parameters *)
WriteId	ARRAY	-	Vector with up to 32 values of type DOUBLE	-

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

With the help of the 'WriteId' input, the values for up to 32 drive parameters can be written to the block.
To do this, the list of the desired parameters must first be entered in the Configuration of the function block parameters under 'IDs input.'
The drive parameters can then be entered at the 'WriteId' input with the help of a multiplex function block.

The values must be entered in the order specified under 'IDs input.'

The values at the input are evaluated with the internal scaling.
(e. g. ID82 'Positive torque limit' is scaled with 0.1 % M_N)

Example

For example, the controller parameters for the speed controller can be introduced so that the control loop can be optimized.

- Function block parameters 'IDs input'
  ID100 'Speed control proportional gain KP'
  ID101 'Integral-action time speed control TN'
  ID102 'Differentiating time speed control TD'
  => IDs input = [100 101 102]
- Connection of the 'WriteId input'

5.3 Outputs of the function block SimKW

5.3.1 Status

Output variables

Name	Design	Unit	Description	Drive parameter *)
Status	UINT32	-	Status word	-
Error	UINT16	-	Error code of the first pending error Error = 0: no error Error > 0: Error number	ID390

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The status display for the control procedure includes the status word and the error display.

5.3.1.1 Status of the controller

The status word is output at the function block SimKW as a decimal value. To determine the status, this value must be converted to a binary value. This allows determination of the individual bits of the status word.

Status word

Bit	State	Description
0 - 9		Reserved
10	0	No warning
	1	Warning
11	0	QRF not pending
	1	Acknowledgment of controller enable (QRF), control active
12	0	No error
	1	Error; diagnostic number is output at the ‘Error’ output
13	0	System booting completed
	1	System booting active
14 - 31		Reserved

The screenshot of the Example project 'SimKWTest' shows the value 3181056. Converted to binary code, this indicates the following status of the simulation:

```
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
| | | | | | | | | | | | | | | | | | | | | | L L L L L L L L L L Reserved
| | | | | | | | | | | | | | | | | | | | | L _ _ _ _ _ _ _ _ _ _ No warning
| | | | | | | | | | | | | | | | | | | | L _ _ _ _ _ _ _ _ _ _ _ QRF set, control active
| | | | | | | | | | | | | | | | | | | L _ _ _ _ _ _ _ _ _ _ _ _ No error
| | | | | | | | | | | | | | | | | | L _ _ _ _ _ _ _ _ _ _ _ _ _ System booting terminated
L L L L L L L L L L L L L L L L L L _ _ _ _ _ _ _ _ _ _ _ _ _ _ Reserved
```

5.3.1.2 Error display

In the event of an error, the output ‘Error’ supplies the number of the diagnostic message.
All diagnostic messages that can be generated by the controller are explained in the document PDK_025786_Diagnose.

5.3.2 Phase voltages

Output variables

Name	Design	Unit	Description	Drive parameters *)
Uu	DOUBLE	V	Voltage at output Phase U (balanced)	
Uv	DOUBLE	V	Voltage at output Phase V (balanced)	
Uw	DOUBLE	V	Voltage at output Phase W (balanced)	

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The three motor phase voltages represent the connections to the motor.
The outputs 'Uu', 'Uv' and 'Uw' are connected 1:1 to the inputs of the function block for motor simulation.

Observe the phase sequence when transmitting the motor phase voltages!

5.3.3 DC bus

Output variables

Name	Design	Unit	Description	Drive parameters *)
Iz	DOUBLE	A	Drawn DC bus current	

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The current drawn from the DC bus is output at output ‘Iz.’

5.3.4 Parameter access

Output variables

Name	Design	Unit	Description	Drive parameter *)
ReadId	ARRAY	-	Vector with up to 32 values of type DOUBLE	-

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The values for up to 32 drive parameters can be read out via the output ‘ReadId.’

To do this, the list of the desired parameters must first be entered in the Configuration of the function block parameters under 'IDs output.'

The values are separated with the help of a demultiplexer function block. The probe values can be recorded with the oscilloscope.

The values at the output are evaluated with the internal scaling.
(e. g. ID84 'Torque feedback value' is scaled with 0.1 % M_N)

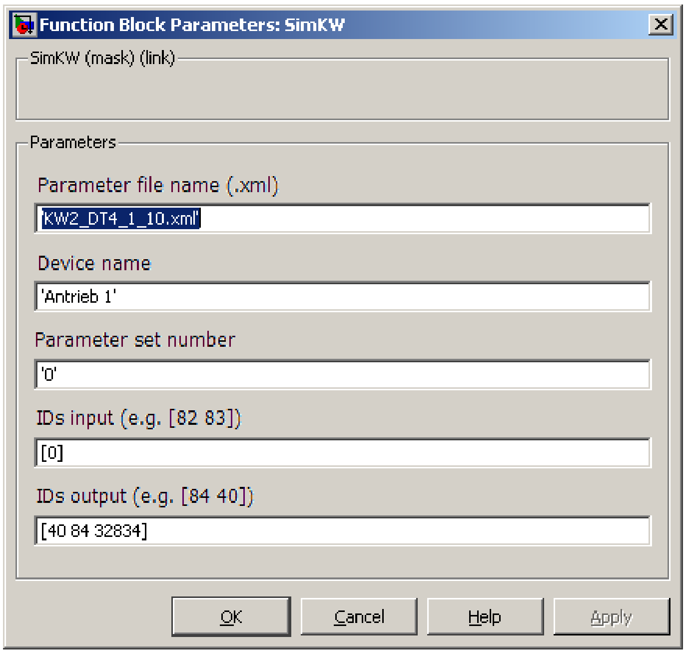
Example

For example, the actual values for speed, torque and torque-forming current can be read out.

- Function block parameters ‘IDs output’
  ID40 'Velocity feedback value'
  ID84 'Torque feedback value'
  ID32834 'Torque current feedback'
  => IDs output = [40 84 32834].
- Connection of the ‘ReadId’ output

5.4 Configuration of the SimKW function block

The parameters of the SimKW function block are set in a dialog accessible via the context menu **'Mask Parameters.'**



5.4.1 Parameter file

Name	Design	Description
Parameter file	STRING	Name of the file with the drive parameters [*.xml]

The parameter file contains the drive parameters. It is exported from an AIPEX PRO project (Siehe 'Reading out and exporting drive parameters with AIPEX PRO' auf Seite 30.).
The name of the parameter file is entered as '>Parameterdatei<.xml' in inverted commas.

Pay attention to correct spelling of the file and device names.
'Test.xml' ≠ 'TEST.xml'

To ensure that all device-specific data are included, a parameter set must be downloaded from the device and exported to the parameter file.
Siehe 'Reading out and exporting drive parameters with AIPEX PRO' auf Seite 30.

To modify parameters subsequently, e. g. ID32800 'AMK main operating mode' or ID32953 'Encoder type', open the corresponding AIPEX PRO project, edit the parameters there and re-export the parameter set to the *.xml file.

5.4.2 Device name

Name	Design	Description
Device name	STRING	Name of the drive in the simulation

The device name enables you to distinguish between the individual drives processed together in a simulation. The content from ID34071 'System name' must be entered in inverted commas.

5.4.3 Parameter set number

Name	Design	Description
Parameter set number	STRING	Number of the parameter set to be used for simulation

Enter the number of the selected parameter set to be used for simulation in inverted commas here.

The same parameter set entered here must be selected when exporting the parameters from AIPEX PRO. Only this parameter set is exported in each case.
Siehe 'Reading out and exporting drive parameters with AIPEX PRO' auf Seite 30.

5.4.4 IDs input

Name	Design	Description
IDs input	ARRAY	Vector with up to 32 values of type UINT16

The list of IDs to be written cyclically with the values from the WriteID input must be specified here (Siehe 'Parameter access' auf Seite 20.).
The number and order of the signals at the Writeld input must match the list in this parameter. A maximum of 32 values is permissible.
Enter the list of IDs in square brackets.

Example:

The following controller parameters are transmitted:

ID100 'Speed control proportional gain KP'

ID101 'Integral-action time speed control TN'

ID102 'Differentiating time speed control TD'

ID104 'Position loop factor KV'
=> 'IDs input' = [100 101 102 104]

5.4.5 IDs output

Name	Design	Description
IDs output	ARRAY	Vector with up to 32 values of type UINT16

The list of IDs to be output cyclically at the ReadId output must be specified here (Siehe 'Parameter access' auf Seite 22.). This means that the number and order of the signals at the ReadId output matches the list in this parameter. A maximum of 32 values is permissible.

Enter the list of IDs in square brackets.

Example:

The following parameters are to be output:

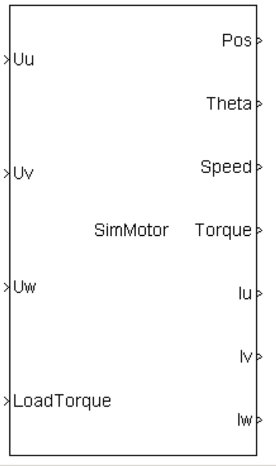
ID40 'Velocity feedback value'

ID84 'Torque feedback value'

=> 'IDs output' = [40 84]

6 Function block SimMotor

Depiction of the function block in MATLAB/Simulink



The function block SimMotor is used for simulation of the motor. A permanently energized synchronous motor are modeled in this function block.

Input values of the function block SimMotor are the phase voltages and the load torque.

The motor data are provided separately in a text file.

Output values are the phase currents, the rotor angle, speed and motor torque.

As an alternative to the simulation of the synchronous motor, other motor models not contained in this software package may be used, for example for a asynchronous motor.

It is important that the phase voltages are provided as input values and the phase currents and the rotor angle as output values.

The block can be used multiple times in a Simulink model. A ‘SimMotor’ is assigned to each 'SimKW’ block.

6.1 Interface of the SimMotor function block

The illustration below shows the SimMotor function block in detail, comparing the functionally related inputs and outputs.

User interface

	SimMotor		
			Rotor position
		Pos	->
		Theta	->
			Actual values
		Speed	->
		Torque	->
Phase voltages			**Phase currents**
->	Uu	Iu	->
->	Uv	Iv	->
->	Uw	Iw	->
Load torque			
->	LoadTorque		

6.2 Inputs of the function block SimMotor

6.2.1 Phase voltages

Input variables

Name	Design	Unit	Description	Drive parameter *)
Uu	DOUBLE	V	Phase voltage Phase U	
Uv	DOUBLE	V	Phase voltage Phase V	
Uw	DOUBLE	V	Phase voltage Phase W	

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The three motor phase voltages represent the connections between the compact inverter and the motor. The outputs 'Uu', 'Uv' and 'Uw' of the function block ‘SimKW’ are connected 1:1 to the inputs of the motor simulation.

Observe the phase sequence when transmitting the motor phase voltages!

6.2.2 Load torque

Input variables

Name	Design	Unit	Description	Drive parameter *)
LoadTorque	DOUBLE	Nm	Load torque	

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

At the input ‘LoadTorque,’ a load torque can be defined in order to simulate the retroactive effect of the machine on the motor. This allows, for example, simulation of a hanging axis.
The value is entered as a decimal value in Nm.

6.3 Outputs of the function block SimMotor

6.3.1 Rotor position

Output variables

Name	Design	Unit	Description	Drive parameter *)
Pos	DOUBLE	rad	Rotor position	
Theta	DOUBLE	rad	Rotor angle (-π ... π)	

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The rotor angle is output at the ‘Pos’ output. 'Theta' gives the angle of the rotor based on one rotor revolution.
Via feedback of the rotor position, the function block SimKW receives the actual position value of the motor for speed and position control.

6.3.2 Actual values

Output variables

Name	Design	Unit	Description	Drive parameter *)
Speed	DOUBLE	rad/s	Angular velocity	
Torque	DOUBLE	Nm	Inner motor torque	

*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The outputs 'Speed' and 'Torque' supply the actual values for the velocity in rad/s and the torque in Nm respectively. The values can be recorded with the oscilloscope function.

6.3.3 Phase currents

Output variables

Name	Design	Unit	Description	Drive parameter *)
Iu	DOUBLE	A	Phase current Phase U	
Iv	DOUBLE	A	Phase current Phase V	
Iw	DOUBLE	A	Phase current Phase W	

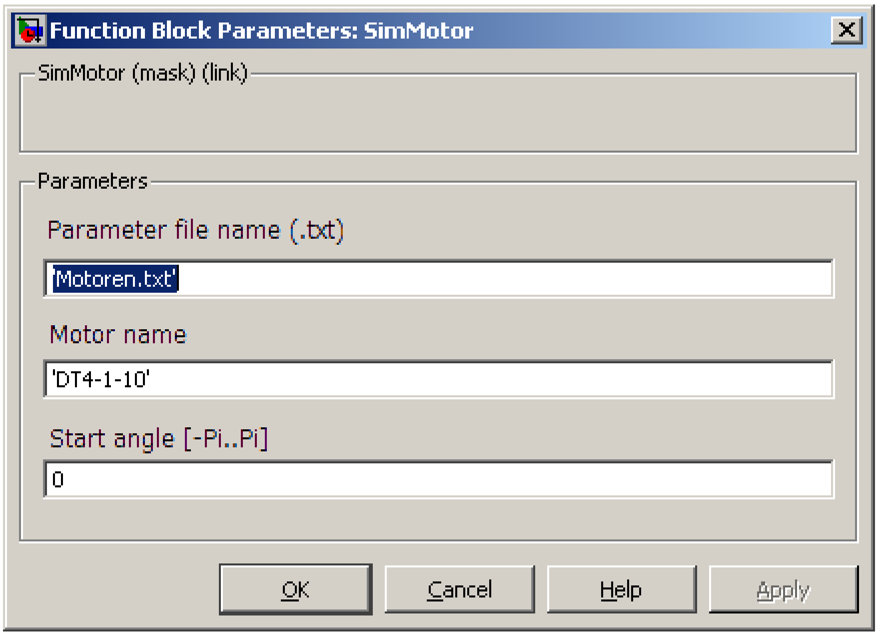
*) The listed drive parameters form the interface to the real compact inverter and the AIPEX PRO PC software. The parameters are not relevant for the simulation.

The outputs 'Iu', 'Iv' and 'Iw' report the phase currents calculated by the function block SimMotor. The values are returned as inputs to SimKW.

Observe the phase sequence when feeding back the currents to SimKW!

6.4 Configuration of the SimMotor function block

The parameters are set in a dialog accessible via the context menu **'Mask Parameters.'**



6.4.1 Parameter file

Name	Design	Description
Parameter file	STRING	Name of the file with motor parameters

The properties of the motor are defined in the parameter file.
Severals motors can be defined in one file.
The file is edited manually in any text editor and must be saved in the simulation path.
The file name is entered at the block SimMotor as '>Parameterdatei<.txt' in inverted commas.

The precision of the simulation results depends heavily on the quality of these motor parameters.

Parameter file 'Motoren.txt'

Motoren.txt - Editor

Datei Bearbeiten Format Ansicht ?

```
# Motor-Parameter
#
Motor SYNC_DEFAULT
   R     = 27.9          # Strangwiderstand in Ohm
   Ld    = 0.02588       # Längsinduktivität in H
   Lq    = 0.02974       # Querinduktivität in H
   Psi   = 0.0728        # Fluss in Vs
   Zp    = 5.0           # Polpaarzahl
   T     = 0.00035       # Trägheitsmoment in kgm²
   IL2   = 2000.0        # Strom für halbe Induktivität in A
   A     = 0.0           # Amplitude Cosinus harmonische Störung 1 in Vs
   B     = 0.0           # Amplitude Sinus harmonische Störung 1 in Vs
   Harm = 5.0            # Perioden pro Rotorumdrehung Harmonische Störung 1
   A2     = 0.0          # Amplitude Cosinus harmonische Störung 2 in Vs
   B2     = 0.0          # Amplitude Sinus harmonische Störung 2 in Vs
   Harm2  = 4.0          # Perioden pro Rotorumdrehung Harmonische Störung 2
   INOISE = 0.0          # Amplitude Rauschen bei Strommessung in A
   TNOISE = 0.0          # Amplitude Rauschen bei Positionsmessung in rad
End Motor

Motor DP13-600-12
   R     = 0.02
   Ld    = 0.0002
   Lq    = 0.000174
   Psi   = 0.196
   Zp    = 6.0
   T     = 0.3
   IL2   = 2000.0
   A     = 0.0
   B     = 0.0
   Harm = 6.0
End Motor
```

Zeile 21, Spalte 1

Each motor is described by the following parameters:

Motor parameters

Name	Design	Unit	Description
R	DOUBLE	Ω	Resistance of the motor phase R_s
Ld	DOUBLE	H	d-axis inductance of one motor phase L_d
Lq	DOUBLE	H	q-axis inductance of one motor phase L_q

Name	Design	Unit	Description
Psi	DOUBLE	Vs	Rotor flux • The rotor flux Ψ can be calculated from ID34234'Voltage constant Ke': • Alternatively, the rotor flux Ψ can be calculated from ID32771 'Nominal torque' and ID111'Motor nominal current IN':
Zp	DOUBLE		Pole pair number
T	DOUBLE	kgm^2	Rotor moment of inertia
IL2	DOUBLE	A	Current at which inductance is halved (This parameter is used to describe saturation effects. If saturation is not to be taken into account, a value significantly higher than the nominal current must be entered.)
A / A2	DOUBLE	Vs	Harmonic distortion 1 / 2: amplitude of the cosine proportion
B / B2	DOUBLE	Vs	Harmonic distortion 1 / 2: amplitude of the sine proportion
Harm / Harm2	DOUBLE		Harmonic distortion 1 / 2: periods per rotor revolution (for simulation of cyclic distortions, z. B. cogging, gearing imbalance)
INOISE	DOUBLE	A	Noise amplitude during current measurement
TNOISE	DOUBLE	rad	Noise amplitude during position measurement

6.4.2 Motor designation

Name	Design	Description
Motor designation	STRING	Designation of the motor in the parameter file

Here, the designation of the motor must be entered in inverted commas , as it appears in the parameter file (e.g. 'DP13-600-12').

6.4.3 Start angle

Name	Design	Description
Start angle	DOUBLE	Rotor angle at the start of the simulation

Position of the rotor at the start of the simulation in rad. Value range -π ... π.

7 Reading out and exporting drive parameters with AIPEX PRO

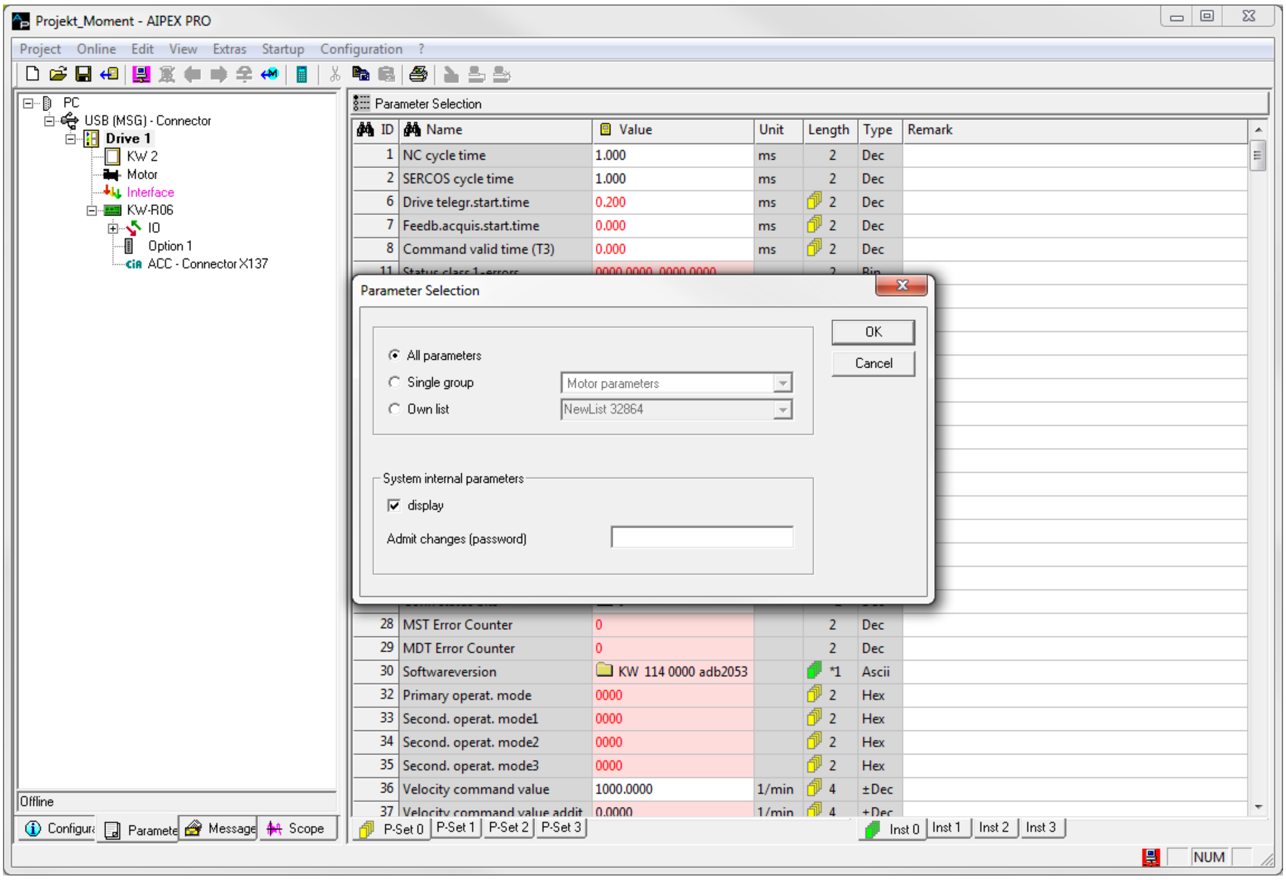
Various AIPEX PRO projects containing the data for various combinations of AMK compact inverters and motors are supplied with the installation data.

The data are configured so that each of these projects can be used together with the file 'Motoren.txt' for the simulation. The corresponding *.xml file must be generated for the desired project.

1. Launch AIPEX PRO
2. Open the desired project
3. Proceed from step 3 (see below)

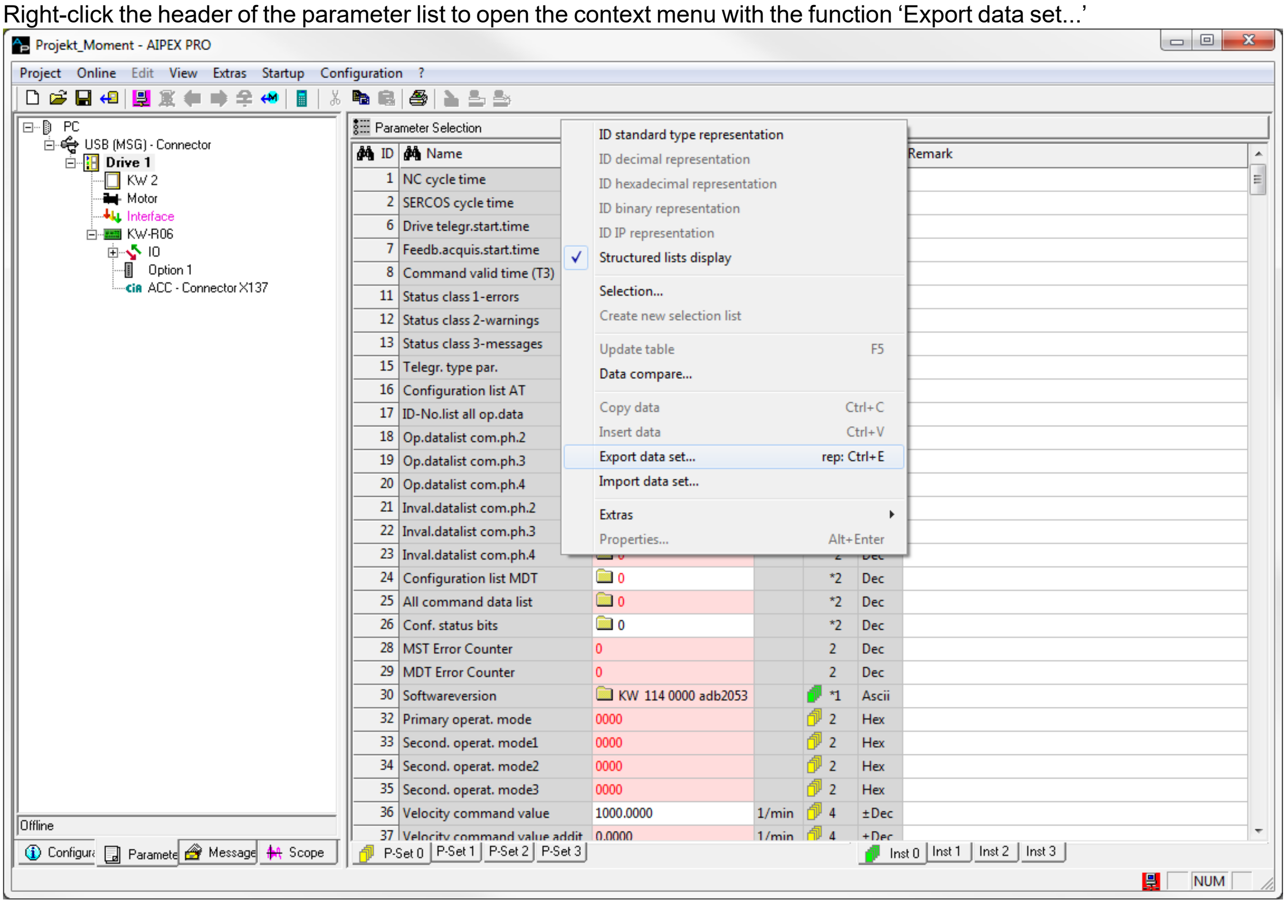
In order to simulate a real drive, you need to read out the drive parameters from a drive and export them to an *.xml file using the AMK PC software AIPEX PRO

1. Connect the PC with the USB port of the controller card, for example.
2. Log in to the device with AIPEX PRO
3. Display all parameters for the drive.
   To do this, click the header of the parameter list, select 'All parameters' and 'Display system internal parameters' and confirm with **'OK'**.



4. The following parameters must be checked and if necessary set in the parameter list:
   - ID32800 'AMK main operating mode':
     In bit 16 - 23, 0x3C or 0x41 must be set as the setpoint source (cyclical setpoints). If this is not done, the inputs of the SimKW block will not be effective as setpoints.
   - ID32953 'Encoder type':
     The following encoder types are permissible as motor encoders for the simulation:
     - incremental encoder
       I-encoder
       E- / F-encoder
       P- / Q-encoder
       S- / T-encoder
       square-wave pulse encoder
     - Resolver
   - ID34071 'System name':
     The device name cannot be edited directly in the parameter editor, but only in the device tree.
     Change the name there (e.g. 'Drive 1' => 'Drive_1') for this change to be adopted in ID34071. This device name will later be assigned to the corresponding function block 'SimKW.'

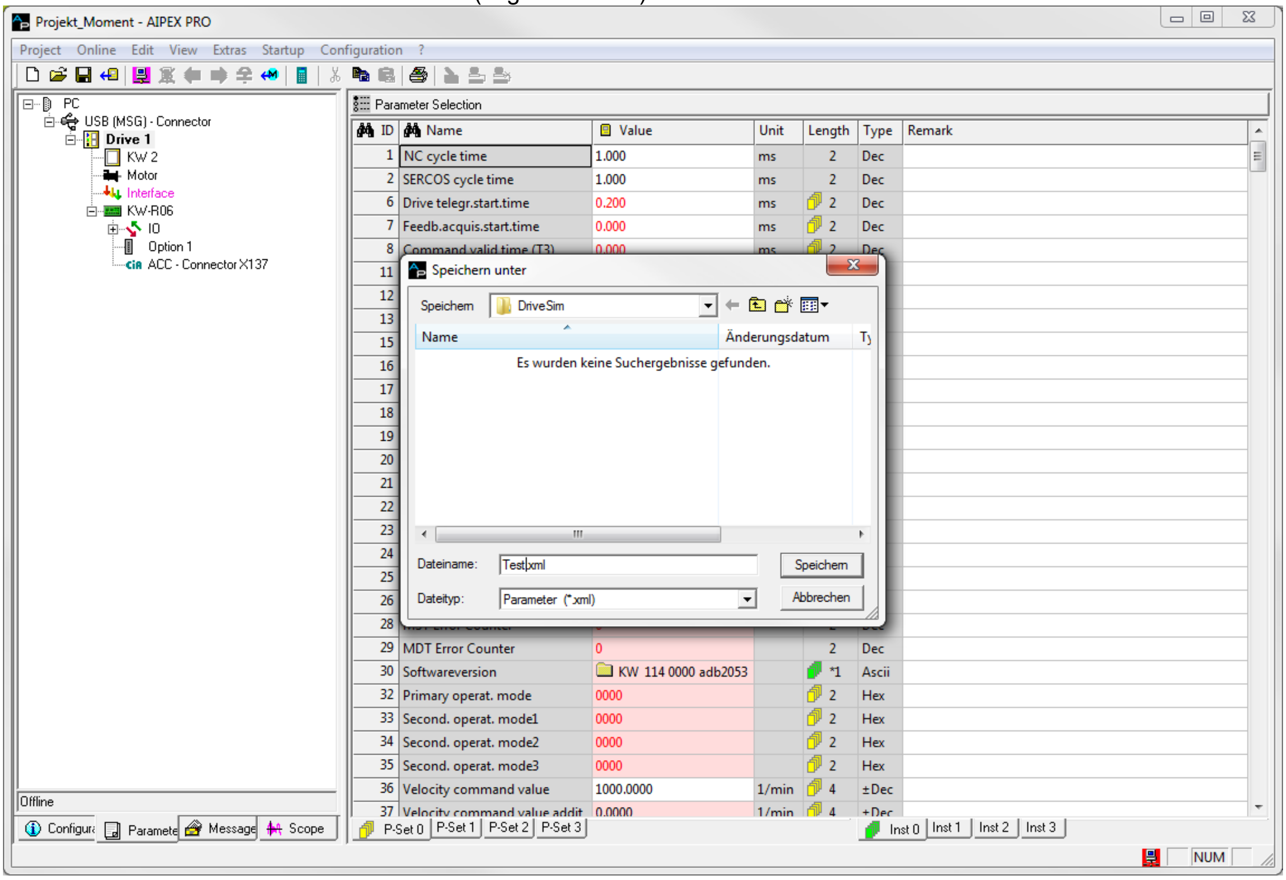
5. Right-click the header of the parameter list to open the context menu with the function 'Export data set...'



Make sure you have selected the correct parameter set (P set 0 ... P set 3) while exporting the project. Only this parameter set will be exported.

The parameter set number is saved in the export file and must match the number entered later in the function block.

6. Save the file in the selected simulation folder (e. g. 'DriveSim').



To export the complete parameter list, proceed by right-clicking in the header of the parameter list as described above (see step 4).

Write-protected and formal parameters are not saved under the menu item ‘Project’ ->’Export data set ...’!

7. For further information on working with the PC software AIPEX PRO, see:
See document PDK_202234_Software_AIPEX_PRO.

Glossary

A

AIPEX
AMK startup and parameterizing software (PC software): Programming, parameterization, configuration, diagnosis, oscilloscope, status information

ARRAY
List with equal format elements

D

DZR
Speed control

E

E-encoder
Absolute encoder, singleturn, EnDAT 2.1 with additional sine and cosine track

F

F-encoder
Absolute encoder, multiturn, EnDAT 2.1 with additional sine and cosine track

FL
Command (Causes a new system run-up)

I

ID
Parameter identification numbers acc. to SERCOS Standard

I-encoder
Incremental encoder, optical encoder with sine and cosine track and zero pulse

IGBT
Power electronic component, e. g. transistor

K

KW-Rxx
AMKASYN controller card for installation into compact inverter

Kv
Position loop factor

KP
Proportional gain (speed control, PID controller)

KW
AMKASYN compact inverter

M

M(N)
Nominal torque

MCE
Motor Controller Electronic

P

Parameter
Identification number acc. to SERCOS standard

PDK_xxxxxx_abcdefgh
Product documentation; xxxxxx - AMK part no. , abcdefgh - name

P-encoder
Absolute encoder singleturn, EnDAT 2.2 light

PWM
Pulse width modulation

Q

Q-encoder
Absolute encoder multiturn, EnDAT 2.2 light

QRF
Acknowledgment controller enable; the drive is controlled in the activated operation mode

R

Resolver
Absolute angle encoder singleturn (1 sine and cosine track per rotation)

RF
Command 'Controller enable'; the drive is energized and will be controlled depending on the selected operation mode. Controller enable can only be set if the device is error-free (SBM = TRUE) and acknowledgement DC bus on is set (QUE = TRUE).Acknowledgement controller enable (QRF) is set.

S

S-encoder
Absolute encoder, singleturn, RS485 Hiperface with sine and cosine track

T

Td
Differentiating time in speed control (PID controller)

T-encoder
Absolute encoder, multiturn, RS485 Hiperface with sine and cosine track

Tn
Integral-action time in speed control (PID controller)

U

UZ
DC bus (voltage)

Your opinion is important!

With our documentation we want to offer you the highest quality support in handling the AMKmotion products.
That is why we are now working on optimizing our documentation.
Your comments or suggestions are always of interest to us.
We would be grateful if you take a bit of time and answer our questions. Please return a copy of this page to us.

e-mail: Documentation@amk-motion.com
or
fax no.: +49 7021/50 05-199

Thank you for your assistance.
Your AMKmotion documentation team

1. How would you rate the layout of our AMKmotion documentation?

   (1) very good (2) good (3) satisfactory (4) less than satisfactory (5) poor

2. Is the content structured well?

   (1) very good (2) good (3) moderate (4) hardly (5) not at all

3. How easy is it to understand the documentation?

   (1) very easy (2) easy (3) moderately easy (4) difficult (5) extremely difficult

4. Did you miss any topics in the documentation?

   (1) no (2) if yes, which ones:

5. How would you rate the overall service at AMKmotion?

   (1) very good (2) good (3) satisfactory (4) less than satisfactory (5) poor

AMKmotion GmbH + Co KG
Phone : +49 7021/50 05-0, fax: +49 7021/50 05-199
E-Mail: info@amk-motion.com
Homepage: www.amk-motion.com