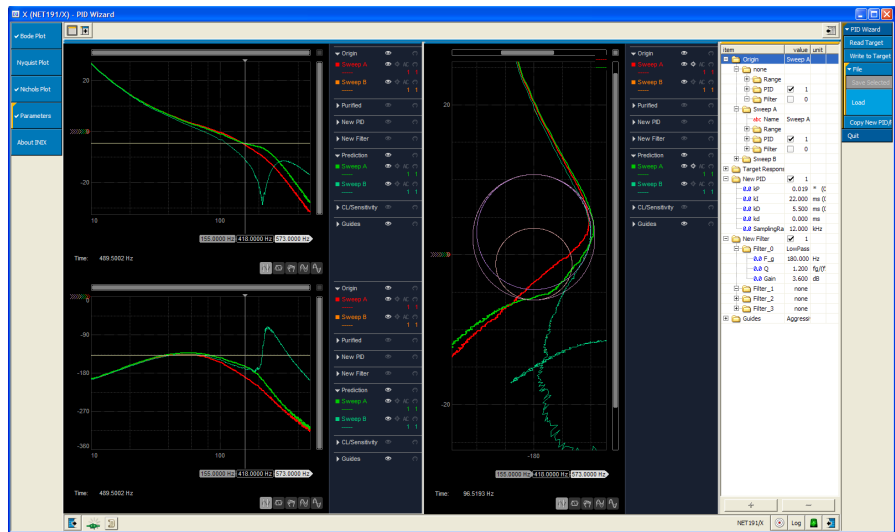


INDEL
INDUSTRIELLE ELEKTRONIK
AG

Indel SAC-Drives Inbetriebnahme Manual



Made in Switzerland

Rev 1.38 © Indel AG, 23.07.2020

Inhaltsverzeichnis

1	Einleitung	7
1.1	Allgemeine Hinweise.....	7
1.2	Übersicht des Dokuments.....	7
2	INIX-Motion	8
2.1	Bedienung.....	8
2.2	Logger.....	9
2.3	Laden von Firmware, Motor-Konfiguration, Controller-Konfiguration	11
2.4	Brennen von dt2-Konfigurations Files.....	14
2.5	Test Modi.....	15
3	Motor Konfiguration	16
3.1	Versionierung Motor-Konfiguration.....	16
3.2	Firmware Version	17
3.3	Absolut Encoder	18
3.3.1	Spezifikation der Absolut Encoder.....	18
3.3.2	Konfiguration Absolut Encoder.....	19
3.3.3	Ist-Werte Absolut Encoder.....	20
3.3.4	Errors.....	21
3.3.5	Beispiele.....	23
3.4	Encoder	24
3.4.1	Konfiguration Inkrementalgeber an Encoder- Feedback.....	24
3.4.2	Konfiguration Inkrementalgeber an SinCos Interface.....	24
3.4.3	Ist-Werte Inkrementalgeber.....	24
3.5	Resolver	25
3.5.1	Konfiguration Resolver	25
3.5.2	Ist-Werte Resolver	26
3.6	SinCos	26
3.6.1	Konfiguration SinCos	26
3.6.2	Konfiguration Inkrementalgeber an SinCos Interface.....	27
3.6.3	Ist-Werte SinCos.....	27
3.7	Sinus Cosinus / Resolver Pegel überprüfen	28
3.8	Auto Kommutierung	29
3.8.1	Konfiguration der Auto-Kommutierung.....	29
3.8.2	Auto-Kommutierung mit UVW Puls.....	30
3.8.3	Auto-Kommutierung mit Two-Phase Stepper Methode.....	31
3.8.4	Auto-Kommutierung mit Absolut Encodern.....	32
3.8.5	Auto-Kommutierung mit 360° Felddrehung.....	32
3.8.6	Flag „Not Undwind“ (360° Kommutierung).....	33
3.8.7	Auto-Kommutierung mit Hall-Sensoren für Maxon Motoren	34
3.8.8	Ist-Werte Auto-Kommutierung.....	39
3.9	Stromregler: Current Control	40
3.9.1	Stromregler Varianten.....	40
3.9.2	Stromregler Parameter.....	41

3.9.3	<i>I2t</i> Regelung.....	42
3.10	Extern Enable.....	43
3.10.1	<i>Konfiguration Extern Enable Eingang.....</i>	43
3.10.2	<i>Konfiguration der Not-Stop Brems-Rampe.....</i>	44
3.10.3	<i>Ist-Werte Extern Enable.....</i>	45
3.11	Feedback Motor Field.....	46
3.12	Feedback Position Control	47
3.13	GinLink	48
3.14	Motor	50
3.14.1	<i>Motor Konfiguration.....</i>	50
3.14.2	<i>Ke umrechnen für Linearmotoren.....</i>	53
3.14.3	<i>Ke umrechnen für Maxon Motoren</i>	53
3.15	PWM-Einstellungen.....	54
3.16	Positions Regler.....	55
3.16.1	<i>Konfiguration Positions-Regler.....</i>	56
3.17	Power Supply	58
3.18	Speed Filter.....	59
3.18.1	<i>Average Speed-Filter.....</i>	59
3.18.2	<i>Speed Observer.....</i>	59
3.19	Hardware-Istwerte.....	60
3.20	Info-Link Motor-Konfig Files auf GinLink portieren.....	61
4	Safety Konfiguration.....	62
4.1	Betrieb mit Safe Torque OFF (STO)	62
4.2	Safe Torque Off anfordern	64
5	IMD-Konfiguration.....	66
5.1	GinLink Konfiguration in IMD.....	66
5.2	Achsen Konfiguration in IMD.....	67
5.2.1	<i>Motor.dt2.....</i>	67
5.2.2	<i>PosCtrl.dt2.....</i>	67
6	Regler Konfiguration.....	68
7	Fehlermeldung vom Servo-Drive.....	69
7.1	Fehlermeldungen	69
7.2	Warnungen.....	69
8	Indel Positions Regler.....	70
8.1	Move Kommandos.....	70
8.2	Fehlermeldungen vom Positions Regler im Feldbus Master.....	71
8.2.1	<i>Achsen bewegen.....</i>	72
9	Trapezregler.....	73
9.1.1	<i>ACS-Show.....</i>	73
9.1.2	<i>Vorgaben für Trapez- und S-Profil.....</i>	73
9.1.3	<i>Regelkonstanten.....</i>	74

9.1.4	Ist-Werte.....	74
9.1.5	Achsen Status	74
9.1.6	Fahrbefehle, VRG_BEF.....	74
9.1.7	Normierung, VRG_FLG.....	75
9.1.8	Betriebsart, VRG_TST.....	75
9.1.9	Normfaktoren.....	76
9.1.10	Fehler Meldungen.....	76
10	Inbetriebnahme Schritt für Schritt.....	77
10.1	Motor vor Überlast schützen.....	77
10.2	Motorparameter eintragen.....	77
10.3	Temperatur-Schalter.....	78
10.3.1	Temperaturfühler im Resolver/SinCos Kabel.....	78
10.3.2	Temperatur-Endschalter im Resolver/SinCos Kabel.....	78
10.3.3	Endschalter in den Motor-Leitungen.....	78
10.3.4	Temperaturfühler in den Motor Leitungen.....	79
10.4	Feedback Konfigurieren.....	80
10.5	Feldbus-Kommunikation auf dem Regler konfigurieren.....	81
10.5.1	Konfigurations Beispiel.....	81
10.6	Konfiguration der Feldbus-Kommunikation In der Software	82
10.7	Messsystem in Betrieb nehmen.....	83
10.7.1	Drehrichtung überprüfen.....	83
10.7.2	Normdrehrichtung.....	83
10.7.3	Auflösung der Geber überprüfen.....	84
10.8	Ist-Position im Feldbusmaster überprüfen.....	84
10.9	Externe Reglerfreigabe.....	85
10.10	PWM.....	85
10.11	Power.....	85
10.12	Positionsregler.....	85
10.13	Polpaarzahl finden, verifizieren.....	86
10.14	Drehrichtung verifizieren (vor Kommutierung).....	87
10.15	Stromregler abgleichen.....	88
10.16	Kommutierung.....	90
10.16.1	Auto-Kommutierung mit Sinus-Cosinus und Inkrementalgeber.....	91
10.16.2	Auto-Kommutierung mit Absolut-Encodern.....	92
10.16.3	Resolver Offset von Hand abgleichen.....	93
10.16.4	Spezialfall Z-Achse mit 360 deg FieldRotation.....	94
10.17	Drehrichtung verifizieren (nach Kommutierung).....	95
10.18	Gain- Offset Korrektur für Resolver und SinCos.....	95
10.18.1	Abgleich Resolver.....	96
10.18.2	Abgleich SinCos Geber	96
10.19	PID Parameter abgleichen.....	97
10.19.1	Optimierungsverfahren nach Ziegler-Nichols.....	97
10.19.2	Vorgehen beim Abgleich der PID-Parameter.....	98
10.20	Vorhaltewerte abgleichen.....	102

10.21	Feinabgleich von Ke, Rs und Ls.....	103
10.22	Resonanzen beseitigen.....	104
11	Bode-Sweep – PID-Wizard	105
11.1	Motion Tool Einstellungen.....	105
11.2	PID-Wizard Einstellungen.....	106
11.3	Bode Sweep aufzeichnen.....	107
11.4	Vorgehen beim Optimieren der Regelstrecke.....	108
11.5	Bode Sweep auswerten.....	109
11.6	Wirkung der PID-Parameter.....	113
11.7	Strom-Filter.....	115
11.8	Optimierungsregeln.....	118
11.8.1	Observer Filter.....	118
11.8.2	Average Filter.....	119
11.9	Gantries.....	120
12	Schrittmotor ohne Feedback in Betrieb nehmen.....	121
13	Firmware Update, Parameter Update.....	122
13.1.1	Updates von Parametern und Software	122
13.1.2	Firmware oder Motor-Parameter ins Flash-Prom brennen.....	122
13.1.3	Motorparameter in File speichern.....	123
13.1.4	Parameter vom RAM ins Flash-Prom kopieren.....	123
13.1.5	Informationen.....	123
13.1.6	Flash-PROM Updates automatisieren.....	124
13.1.7	Version und Hilfe.....	124
13.1.8	Updates mit Laptop.....	125
13.2	Notsystem.....	125
14	Trouble Shooting.....	126
14.1	INFO-Link Probleme.....	126
14.2	Probleme mit analogen Gebern: SinCos, Resolver	126
14.3	Verschmutzung.....	127
14.4	Speisung.....	128
14.4.1	Zwischenkreis Spannung	128
14.4.2	Spannungseinbrüche.....	128
14.4.3	Einspeisung MAX-Board.....	129
14.5	Last	130
14.6	PID-Parameter.....	131
14.7	Störungen.....	132
14.8	Vorhaltewerte.....	136
14.9	Normierungs Fehler	137
14.10	Falsches Ke.....	138
14.11	Falscher Resolver Offset.....	139
15	Weiterführende Dokumentationen.....	141

16	Abbildungsverzeichnis.....	142
17	Dokumentenstatus.....	144

1 Einleitung

1.1 Allgemeine Hinweise

Lesen Sie vor Installation und Inbetriebnahme diese Dokumentation sowie Dokumentationen auf die verwiesen wird, vollständig durch. Falsches Handhaben der Module kann zu Personen- oder Sachschäden führen. Halten Sie die technischen Daten und die Angaben zu den Anschlussbedingungen sowie sämtliche Vorschriften unbedingt ein.

1.2 Übersicht des Dokuments

Kapitel 2 beschreibt grob die Bedienung der neuen INIX Tools von Indel.

Kapitel 3 geht detailliert auf die vielen einzelnen Parameter des MotorenConfig Files ein.

Kapitel 4 beinhaltet Informationen bezüglich dem Betrieb mit Safe Torque Off welcher von den SAC3 Regler zu Verfügung gestellt wird.

Kapitel 5 veranschaulicht die notwendigen Konfigurationen auf der Software Seite, welche für einen einwandfreien Betrieb nötig sind. Es werden hauptsächlich die Parameter hervorgehoben, welche in der SAM Konfiguration sowie auf dem Regler bzw. in dem MotorConfig File identisch sein müssen.

Kapitel 6 geht ganz kurz auf das ControlerConfig File ein. Dieses sollte im Normalfall für den Benutzer nicht von Interesse sein.

Kapitel 7 beschreibt Fehlermeldungen vom Drive, welche Früher noch nicht direkt ausgewertet wurden. Heute werden diese jedoch direkt auf dem Regler ausgewertet und sind für den Benutzer direkt ersichtlich.

Kapitel 8 geht auf den Positionsregler und dessen einzelnen Kommandos ein.

Kapitel 9 beschreibt kurz das alte ACS-Show Tool und die Einstellungen des Trapezreglers.

Kapitel 10 bietet eine Anleitung für eine Inbetriebnahme einer Achse. Für eine Inbetriebnahme sollte dieses Kapitel Schritt für Schritt durchgegangen werden.

Kapitel 11 beschreibt die Durchführung eines Bode-Sweep mit dem Axis Tool. Für den Abgleich der PID-Parameter sowie konfigurieren von Strom-Filtern kommt das Tool Indel Axis Tuner zum Einsatz, welches auch in diesem Kapitel beschrieben ist. Damit können Heute bereits komplexe Regelstrecken mit wenig Aufwand in einen stabilen Zustand gebracht werden.

Kapitel 12 beschreibt die Inbetriebnahme von Open-Loop-Schrittmotoren. Also Schrittmotoren welche ohne Feedback betrieben werden.

2 INIX-Motion

2.1 Bedienung

Das Inix-Motion zusammen mit dem Logger-Tool Inix-Varlog wird für die Motor-Inbetriebnahme benötigt.

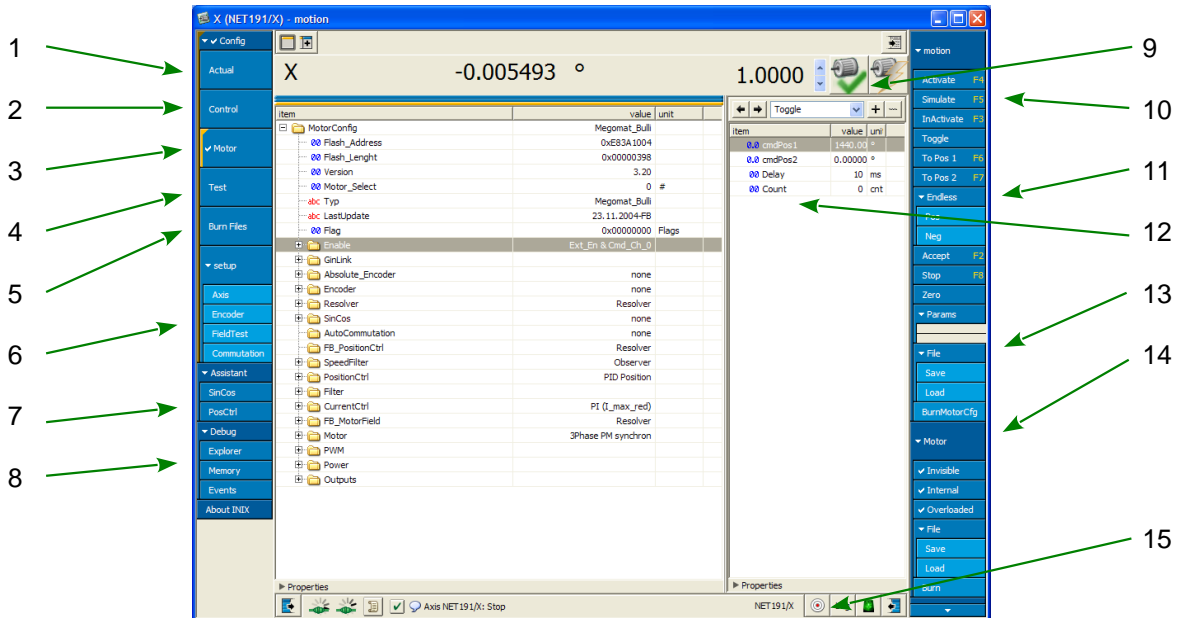


Abb 1: INIX Motion

- 1) Alle Ist-Werte der Achse
- 2) Konfiguration des Drives (inkl. Abtastrate vom Drive)
- 3) Motor-Konfiguration
- 4) Test-Fenster: Strom-Mode, Spannungs-Mode, Kommutierung, usw
- 5) Laden von Konfigurations-Files und Drive-Software
- 6) Verschiedene Inbetriebnahme-Hilfen
- 7) Assistenten für Inbetriebnahme
- 8) Debug: Variablen Explorer, Memory-Dump, usw.
- 9) Symbol für Externe Freigabe, Achse Aktive
- 10) Move-Kommandos
- 11) Endlos fahren
- 12) Parameter-Fenster: Vordefinierte Parameter-Sätze, eigene Parameter-Sätze
- 13) Laden/Speichern von Motor-, bzw. Controller-Konfigurations-Files
- 14) Ein- Ausblenden von internen oder unsichtbaren Parametern
- 15) Target Auswahl

2.2 Logger

Der Indel Variablen Logger ist in das Motion-Tool integriert:

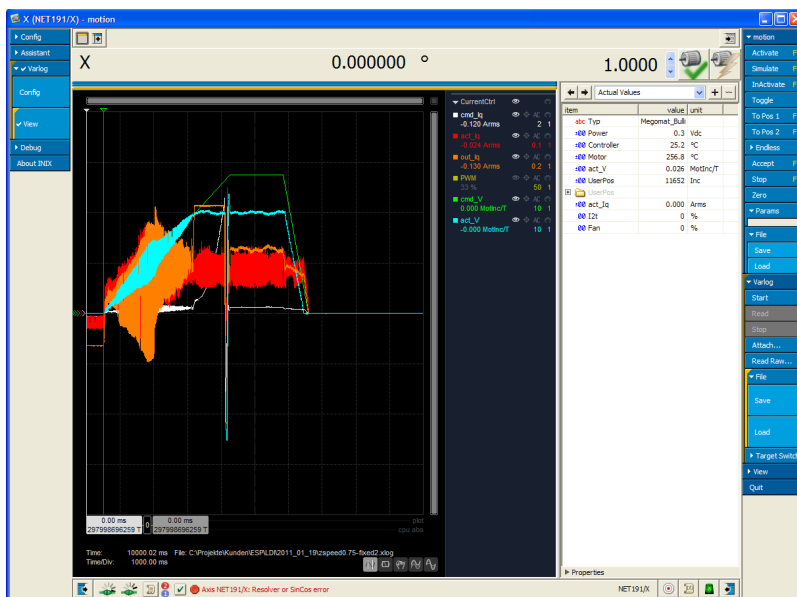


Abb 2: Variablen Logger

Im Variablen Logger wird die Geschwindigkeit in MotInc/T angezeigt.

MotInc Motoren-Inkremente: Die Geber-Auflösung wird im Regler auf 4096 Inkremente pro Motor-Umdrehung normiert. Bei hochauflösenden SinCos Gebern mit z.B. 1'048'576 Inkrementen Auflösung ergeben sich Motor-Inkremente mit Kommastellen.

T Abtastperiode: Je nach Einstellung beträgt die Abtastfrequenz im Regler 8kHz, 12kHz, 16kHz, 24kHz oder 32kHz

Berechnung der Geschwindigkeit in %/s

v Geschwindigkeit in %/s

f Abtastfrequenz-Frequenz in Hz

$$v = \frac{MotInc / T \cdot f \cdot 360}{4096}$$

Berechnung des Schleppfehlers

s	Weg in	mm
K	Spindelsteigung in	mm
P	Polabstand in	mm
G	Getriebefaktor, wird nur benötigt wenn Messsystem am Motor befestigt ist und das Getriebe nachgeschaltet ist.	

bei rotativen Motoren:

$$s = \frac{MotInc \cdot 360 \cdot G}{4096}$$

bei Spindel Motoren:

$$s = \frac{MotInc \cdot K \cdot G}{4096}$$

bei Linear Motoren:

$$s = \frac{MotInc \cdot P}{4096}$$

2.3 Laden von Firmware, Motor-Konfiguration, Controller-Konfiguration

Mit dem Motion-Tool können Motor-Konfiguration, Controller-Konfiguration und Firmware in alle Motion-Boards und Servo-Regler geladen werden.

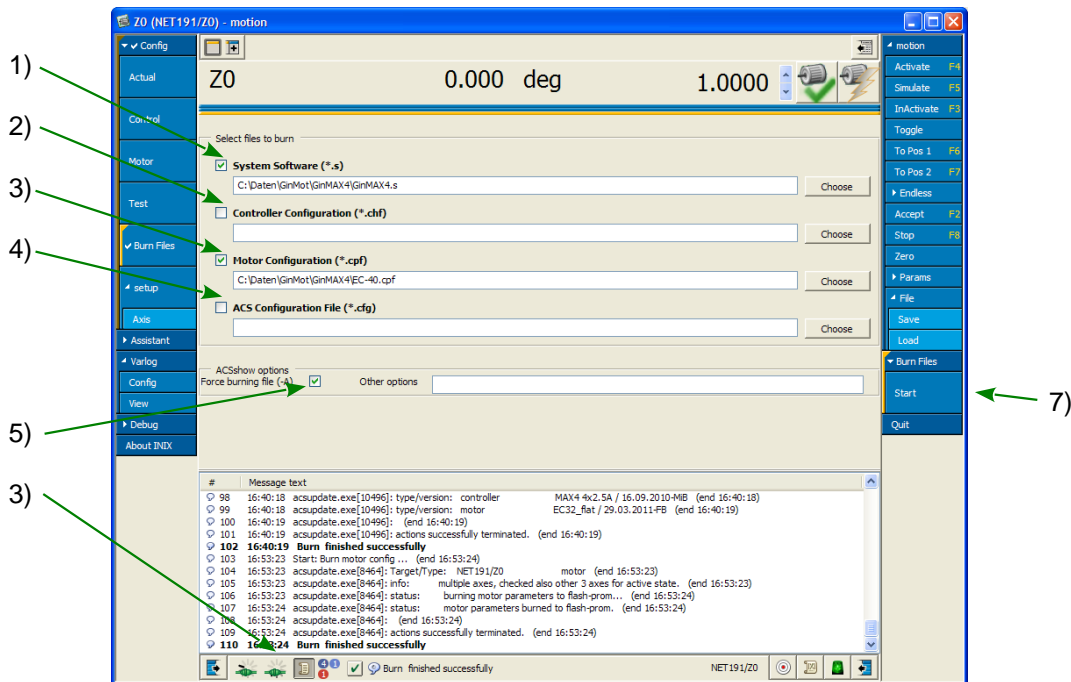


Abb 3: Firmware, Motor- Controller-Konfiguration laden

- 1) Auswahl der Firmware z.B. GinMAX4.s
Bei GinLink targets kann auch das motor_gin.zip File angegeben werden, damit wird automatisch die richtige Software ausgewählt und auf das target gebrannt.
- 2) Auswahl der Controller-Konfiguration

Die Controller-Konfiguration enthält Abgleichdaten der Strom- Spannungsmessung, SinCos Eingänge, sowie ie Angaben zu den maximal Strömen der IGBTs.

Diese Controller-Konfiguration von einem spezifischen Motion-Board oder Servo-Drive darf auf keinen Fall in ein anderes Gerät geladen werden. Dadurch kann der Drive im schlimmsten Fall beschädigt werden.

- 3) Auswahl der Motor-Konfiguration
- 4) Siehe Kapitel 13.1.6 Flash-PROM Updates automatisieren
- 5) -A wird dieses Flag aktiviert werden die angewählten Files in jedem Fall ins Flash-Prom gebrannt.

Ist das Flag nicht angewählt, wird geprüft ob das zu ladende File aktueller ist als das File im Flash-Prom.
Bei der Firmware wird auf die Version geachtet, bei den Konfigurations-Files wird das File-Datum im Parameter: `LastUpdate` beachtet, siehe weiter unten.
- 6) Message-History Fenster einblenden
- 7) Start Button um File Infos anzuzeigen

Abspeichern und Brennen der Motor-Konfiguration

Nachdem Änderungen an der Motor-Konfiguration vorgenommen worden sind, müssen diese in ein File abgespeichert werden und in das Flash-Prom im Regler gebrannt werden. Ansonsten gehen die Änderungen bei Power-Off verloren. Falls beim Brennen der Parameter ein Fehler entsteht und der Drive nicht mehr gebootet werden kann, besteht die Möglichkeit den Drive im Notsystem zu booten. Siehe dazu Kap. 13.2 Notsystem.

Die Motor-Konfiguration kann auch mit dem Tool ACSUpdate.exe gespeichert, bzw. gebrannt werden, siehe Kapitel: 13 Firmware Update, Parameter Update

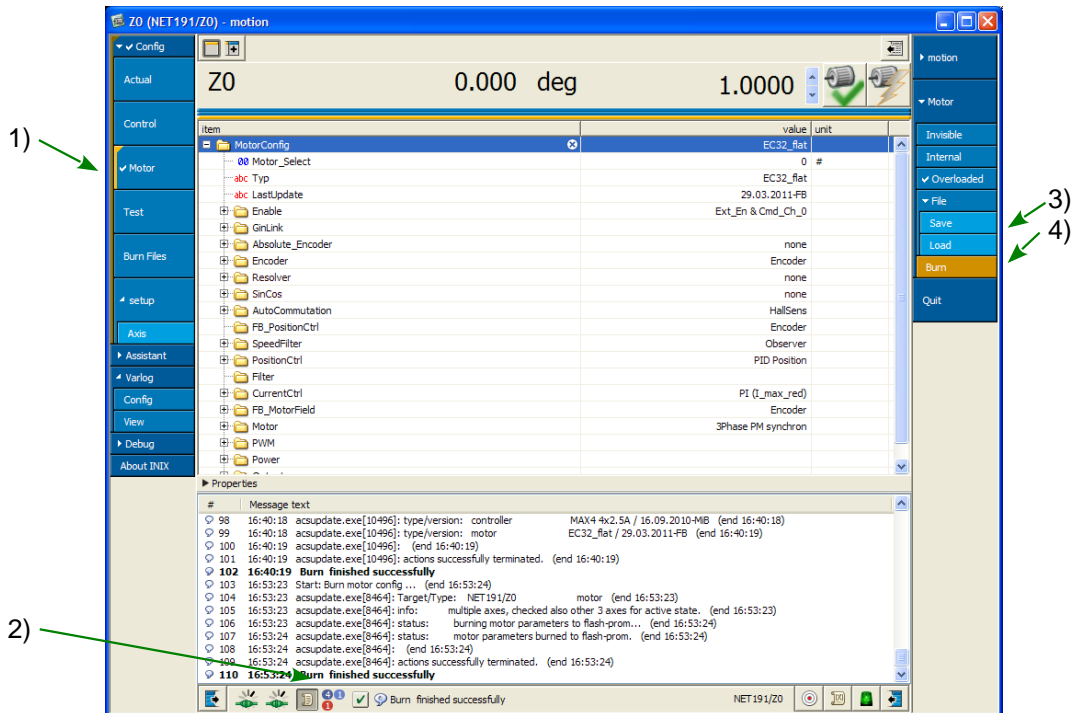


Abb 4: Motor-Config Parameter brennen, speichern

- 1) Motor-Konfiguration
- 2) Message-History Fenster einblenden. Den Text in Message Fenster ungedingt beobachten

um sicher zu gehen, dass die Parameter auch gebrannt wurden.

Wenn die Achse noch aktive ist, können die Parameter nicht gebrannt werden!

- 3) Save, Load Motor-Konfiguration in ein File. Mit Load werden die Motor-Parameter ins RAM des Drives geladen. Um die Parameter dauerhaft zu speichern müssen diese noch zusätzlich mit Burn ins Flash-Prom gebrannt werden.
- 4) Burn Button um Motor-Konfiguration ins Flash-Prom zu laden. Der Burn Button wird orange, sobald ein Motor-Parameter verändert wird.

Über ein Kontext Menü, aufrufbar mit der rechten Maustaste können ebenfalls Motor-Konfig Daten gebrannt und gespeichert werden.

Parameter ins Flash brennen:

rechte Maustaste im Inco-Tree → Motor → BurnMotorCfg
 oder rechte Seite: File: BurnMotorCfg

Parameter speichern

rechte Maustaste im Inco-Tree → Motor → File → Safe/Load
 oder rechte Seite File: File: Safe/Load

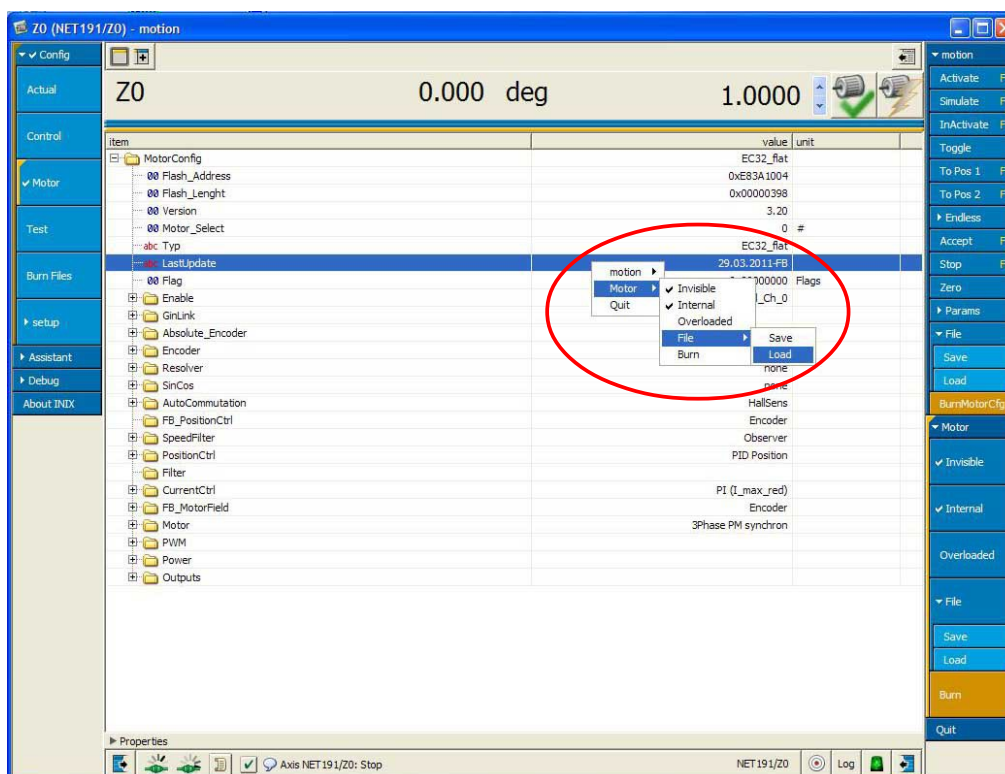


Abb 5: Motor-Konfig Parameter speichern und brennen

2.4 Brennen von dt2-Konfigurations Files

Motion-Boards benötigen eine Konfiguration für die analoge und digitale Peripherie auf dem Motion-Board, sowie die Konfiguration der Rampen für alle Achsen.

Dazu wird eine Dos-Box geöffnet.

Komplette dt2-Konfiguration brennen:

```
trans32 SAM192\AX0 -k AX_Config\dt2config -G -B
```

Ein einzelnes Konfiguration-File brennen:

```
trans32 sio -k -l max-inp.dt2 -b
```

```
trans32 sio -k -l config\maxbus\digital\ip_adr.dt2 -b
```

Gesamte Konfiguration löschen:

```
trans32 SAM192\AX0 -G -B
```

Backup der Konfiguration erstellen:

```
trans32 sio -b -W backup
```

Siehe auch: <http://doc.indel.ch/doku.php?id=software:application:trans32>

2.5 Test Modi

Für die Inbetriebnahme stehen verschiedene Test-Modi zur Verfügung:

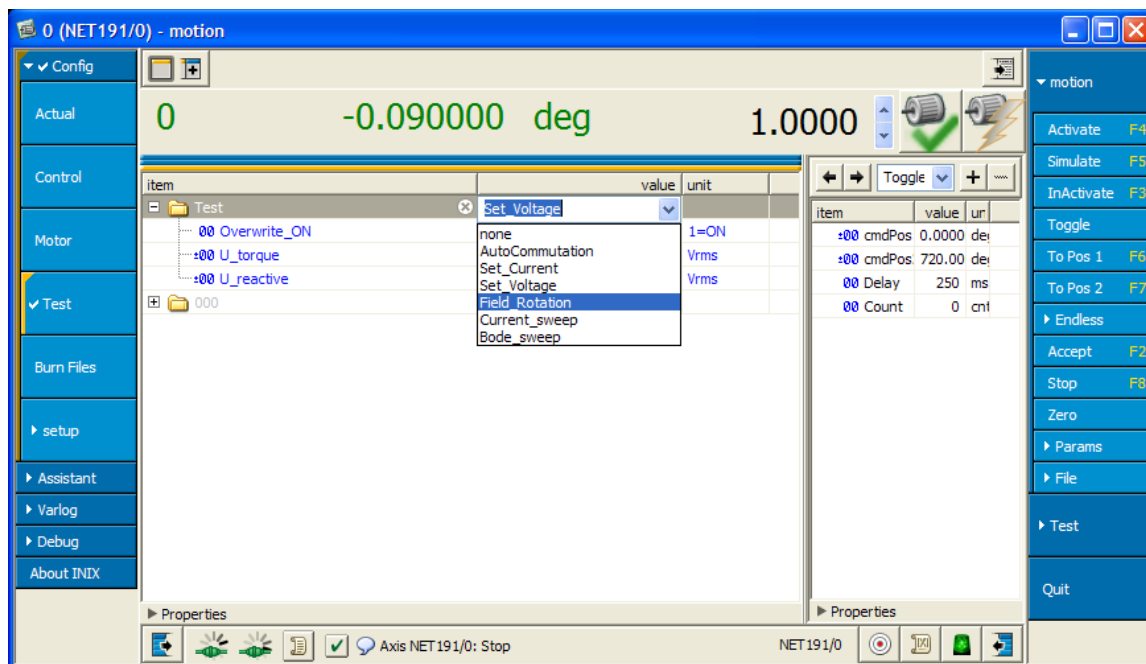


Abb 6: Test-Modi

AutoCommutation	Auto-Kommutierung: automatischer Abgleich des Feldoffsets.
Set_Current	Strom-Mode: Einstellen eines konstanten Wirk- oder Blindstromes. In diesem Mode wird die Kommutierung noch nicht benötigt.
Set_Voltage	Spannungs-Mode: Einstellen einer konstanten Wirk- oder Blindspannung. In diesem Mode muss die Kommutierung erfolgt sein.
Field_Rotation	Feld-Mode. In diesem Mode wird die Kommutierung noch nicht benötigt.
Bode_sweep	Aufzeichnen eines Bode-Diagrammes.

Die einzelnen Test-Modi werden in den nachfolgenden Kapiteln detailliert beschrieben.

3 Motor Konfiguration

3.1 Versionierung Motor-Konfiguration

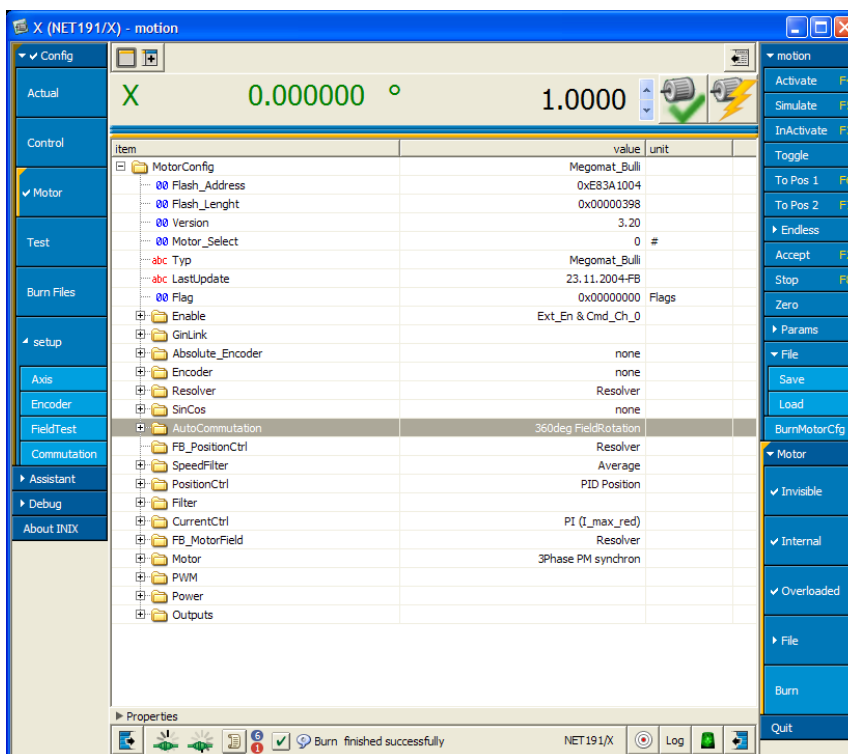


Abb 7: INIX Motion

LastUpdate Dieser Parameter wird für die File-Version des Kunden verwendet.

Typ Motor-Typ hier eingeben.

Version: Interne Versionsnummer Motor-Konfiguration File-Struktur.

Der Parameter `version` darf nicht angepasst werden! Es handelt sich dabei um die Versions-Nummer der Motor-Konfiguration File-Struktur.

Diese Nummer wird von der Drive-Software für die Interpretation von verschiedenen File-Versionen (ältere Regler-Generationen) benötigt.

Bei Mehrfach Drives, bzw. Motion-Boards wie SAC3x3, MAX2, MAX4 und AX-4 mit Firmware-Version älter als Rev. 6.413 - 847 (Rev. 6.4D - 847) muss für alle Motoren ein Motor-Konfig File in den Drive geladen werden.

Ist nur ein Motor-Konfig File geladen, werden die analogen und digitalen I/Os nicht behandelt!

Ab Firmware-Version Rev. 6.413 – 847 muss mindestens für die Achse 0 ein Motor-Konfig File geladen werden.

3.2 Firmware Version

Die Firmware-Version kann über das Motion-Tool oder im Inco-Explorer betrachtet werden:

Im Motion-Tool unter **Burn Files** sämtliche Check-Boxen deaktivieren, danach **Start** drücken. In der **Message-History** wird die Firmware-Version sowie Motor-Config und Controller-Config Version angezeigt.

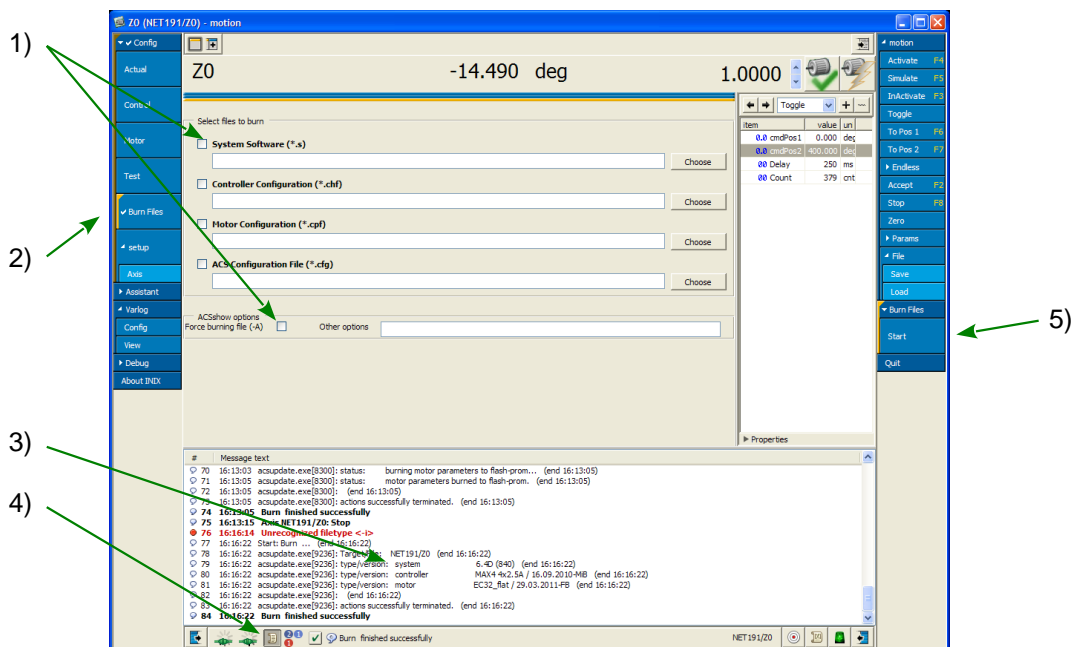


Abb 8: Firmware Version im Motion-Tool

- 1) Check-Boxen deaktivieren
- 2) Burn Files Menue
- 3) File-Versionen
- 4) Message-History Fenster einblenden
- 5) Start Button um File Infos anzuzeigen

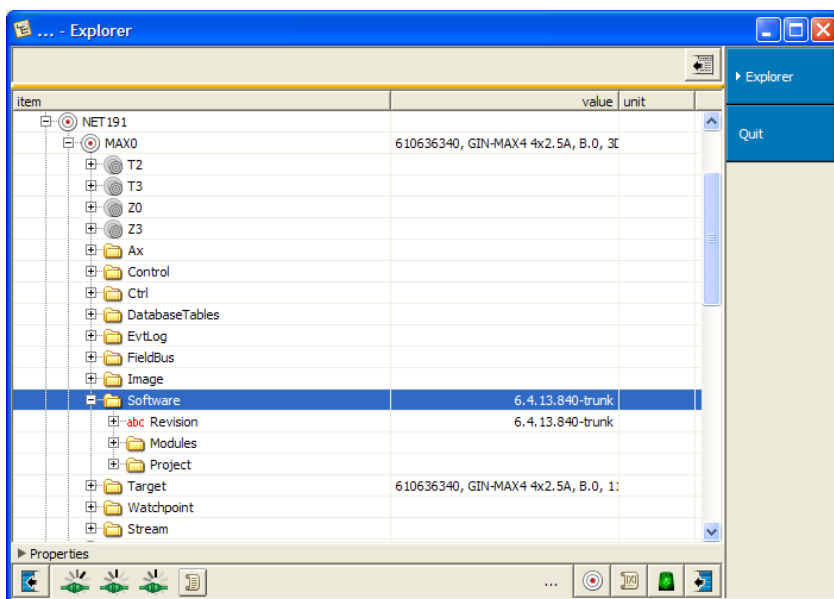


Abb 9: Firmware Version im Inco-Explorer

3.3 **Absolut Encoder**

Die Refresh-Rate aller Absolut-Encoder beträgt 1ms.

3.3.1 **Spezifikation der Absolut Encoder**

Auswahl an Geber-Systemen:

Endat	max. 26 Bit Daten (Inkmente pro Umdrehung + Umdrehungen) Clock Frequenz: 400kHz
Hiperface	max. 48 Bit Daten (Inkmente pro Umdrehung + Umdrehungen) Clock Frequenz 9600 Bits / s
SSI	max. 32 Bit Daten (Inkmente pro Umdrehung + Umdrehungen) Clock Frequenz 400kHz
Used as Encoder	Am Absolut Encoder Eingang kann auch ein Inkrementalgeber angeschlossen werden. Dies gilt für folgende Drives: SAC3x3 und AX4 Damit können auch schnelle digitale Encoder Signale verarbeitet werden. Weitere Details siehe Hardware-Manual-Motion-Boards.pdf und Hardware-Manual-SAC3.pdf

Die Konfiguration des Encoders bleibt im Unterordner `Encoder`.

3.3.2 Konfiguration Absolut Encoder

Inco-Pfad zur Konfiguration der Absolut-Encoder: `Ctrl.MotorConfig.Absolute_Encoder`

IncPerMotorTurn Anzahl Inkremente pro Motor-Umdrehung. Bei Linear-Motoren die Anzahl Indremente pro Polabstand eingeben. Dieser Parameter wird nur für die Autokommutierung benötigt.

BitsPerEncTurn Anzahl Bits pro Encoder Umdrehung. Diesen Wert aus dem Encoder-Datenblatt beziehen.

NrOfEncTurns Bei Multiturn Encodern: hier die Anzahl Turns eingeben. Bei Singelturn Gebern muss hier 1 eingegeben werden.

Flags:
GrayCode 0 für Geber ohne Gray-Code
 1 für Geber mit Gray-Code

direction 0 für CW
 1 für CCW

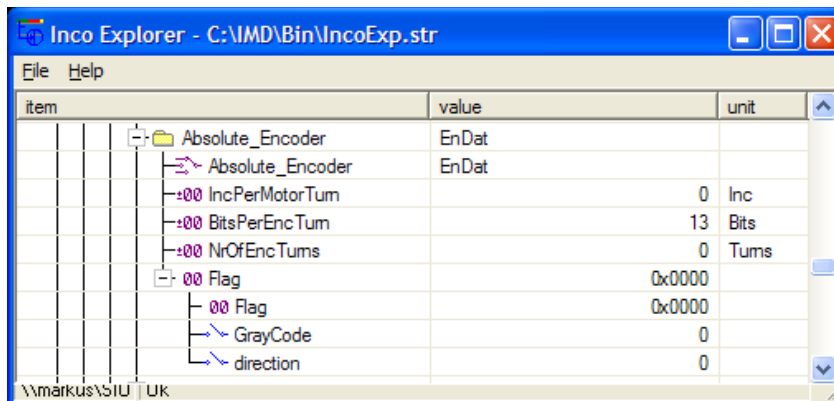


Abb 10: Absolut Encoder Konfiguration

3.3.3 Ist-Werte Absolut Encoder

Inco-Pfad zu den Ist-Werten der Absolut-Encoder: `Ctrl.Actual.Absolute_Encoder`

Status	0	Request data
	1	Read data
	2	Reset
	3	Hardware Reset

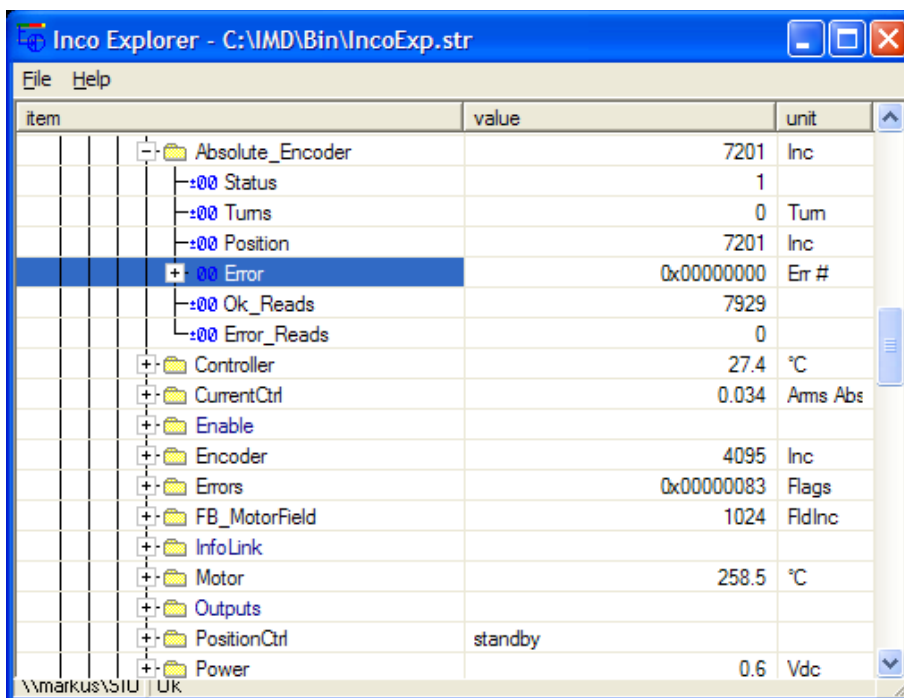
Turns Anzahl Turns (nur bei Multiturngebern)

Position Position in Inkrementen

Error Error Code, siehe weiter unten

Ok_Reads Anzahl empfangener Telegramme

Error_Reads Anzahl empfangener Telegramme mit Fehler



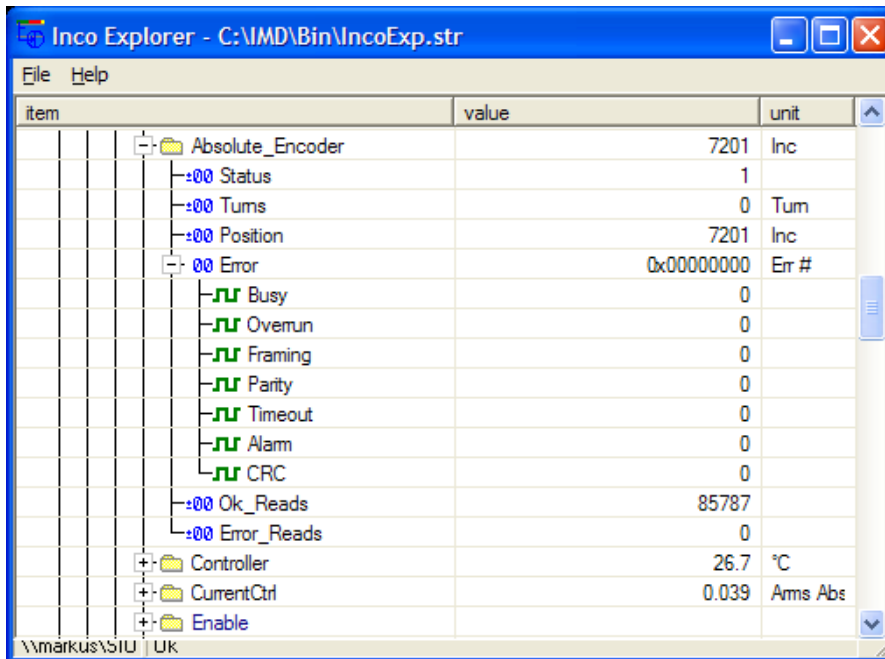
item	value	unit
Absolute_Encoder	7201	Inc
+00 Status	1	
+00 Turns	0	Turn
+00 Position	7201	Inc
+00 Error	0x00000000	Err #
+00 Ok_Reads	7929	
+00 Error_Reads	0	
+ Controller	27.4	°C
+ CurrentCtrl	0.034	Ams Abs
+ Enable		
+ Encoder	4095	Inc
+ Errors	0x00000083	Flags
+ FB_MotorField	1024	FldInc
+ InfoLink		
+ Motor	258.5	°C
+ Outputs		
+ PositionCtrl	standby	
+ Power	0.6	Vdc

Abb 11: Ist-Werte Absolut Encoder

3.3.4 Errors

Error Flags von Endat

Busy	Schnittstelle ist belegt busy
Overrun	Fehler vom UART
Framing	Fehler vom UART
Parity	Fehler vom UART
Timeout	Nicht das komplette Telegramm empfangen innerhalb der Timeout Zeit von 1ms
Alarm	Encoder Alarm Bit, siehe Spezifikation vom Encoder
CRC	Checksum Fehler, tritt z.B. Auf, wenn die Anzahl Bits nicht stimmt.

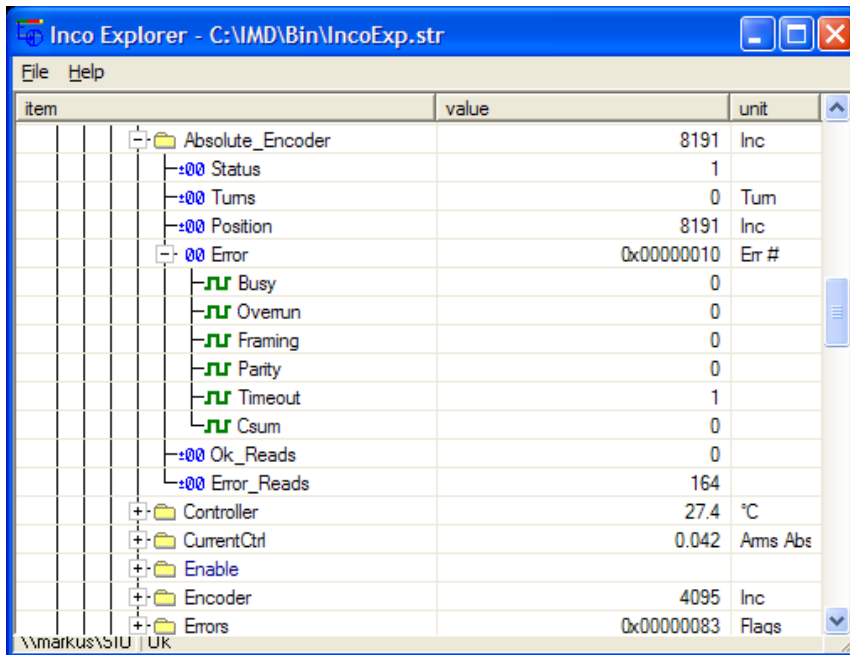


item	value	unit
Absolute_Encoder	7201	Inc
+00 Status	1	
+00 Tums	0	Tum
+00 Position	7201	Inc
-00 Error	0x00000000	Err #
Busy	0	
Overrun	0	
Framing	0	
Parity	0	
Timeout	0	
Alarm	0	
CRC	0	
+00 Ok_Reads	85787	
+00 Error_Reads	0	
+ Controller	26.7	°C
+ CurrentCtrl	0.039	Amps Abs
+ Enable		

Abb 12: Fehler Absolut Encoder Endat

Error Flags von Hiperface Encoder

Busy	Schnittstelle ist belegt busy
Overrun	Fehler vom UART
Framing	Fehler vom UART
Parity	Fehler vom UART
Timeout	Nicht das komplette Telegramm empfangen innerhalb der Timeout Zeit von 1ms
Alarm	Encoder Alarm Bit, siehe Spezifikation vom Encoder
Csum	Checksum Fehler, tritt z.B. Auf, wenn die Anzahl Bits nicht stimmt.



item	value	unit
Absolute_Encoder	8191	Inc
+00 Status	1	
+00 Turns	0	Turn
+00 Position	8191	Inc
00 Error	0x00000010	Err #
Busy	0	
Overrun	0	
Framing	0	
Parity	0	
Timeout	1	
Csum	0	
+00 Ok_Reads	0	
+00 Error_Reads	164	
Controller	27.4	°C
CurrentCtrl	0.042	Arms Abs
Enable		
Encoder	4095	Inc
Errors	0x00000083	Flags

Abb 13: Fehler Absolut Encoder Hiperface

Error Flags SSI (Synchronous serial interface)

Busy Schnittstelle ist belegt

Ok_Reads Das SSI-Protokoll besitzt keine Möglichkeit der Überprüfung wie Parity-Bit oder CRC. Deshalb werden alle eingelesenen Werte als ok erkannt, selbst wenn kein Geber eingesteckt ist.

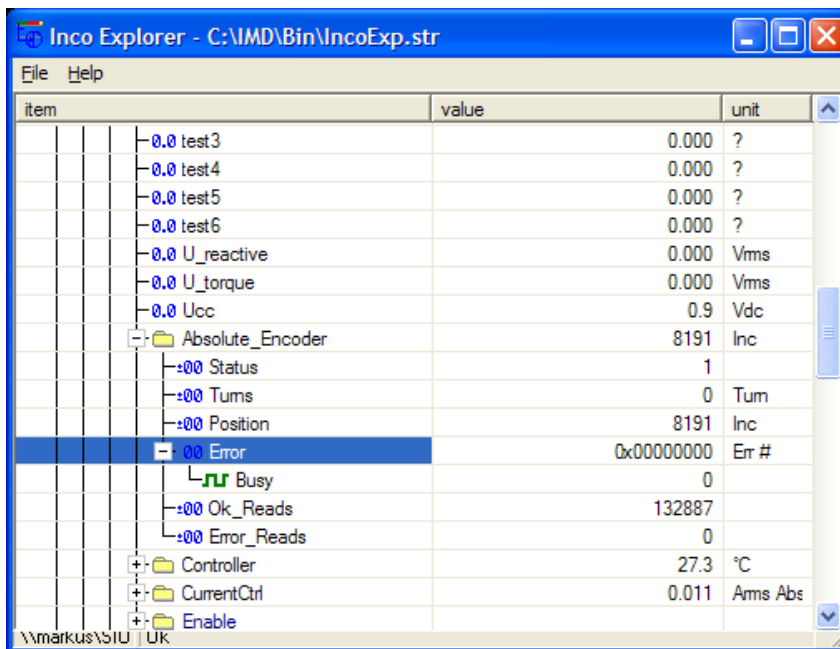


Abb 14: Fehler Absolut SSI

3.3.5 Beispiele

Endat Interface mit 26Bits, Singleturn, keine Autokommutierung:

Absolute_Encoder	EnDat
IncPerMotorTurn	0
BitsPerEncTurn	26
NrOfEncTurns	1

Flags:

GrayCode	0
direction	0

SSI Interface mit 24Bits, Multiturn, mit Gray-Code, keine Autokommutierung:

Absolute_Encoder	SSI
IncPerMotorTurn	0
BitsPerEncTurn	12
NrOfEncTurns	4096

Flags:

GrayCode	1
direction	0

3.4 Encoder

3.4.1 Konfiguration Inkrementalgeber an Encoder- Feedback

Pfad im Inco-Tree zur Inkrementalgeber Konfiguration: `Ctrl.MotorConfig.Encoder`

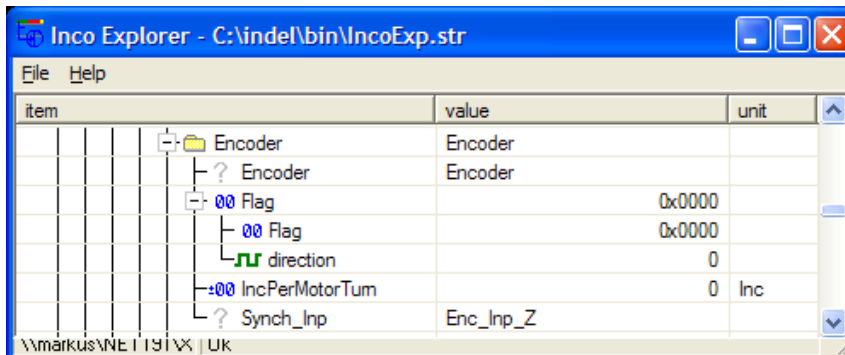


Abb 15: Encoder

- Encoder** Encoder, Nullimpuls von Inkrementalgeber
- Ref_Inp time** Nur Nullimpuls
- IncPerMotorTurn** Anzahl Inkremente pro Motor Umdrehung. Inklusive 4-Quadranten Auswertung: Bei einem Geber mit 1024 Strichen muss hier 4096 eingegeben werden.
- Synch_Inp** kein Nullimpuls, bzw. Nullimpuls von Inkrementalgeber
- Flag.Direction** Flag = 0: CW
Flag = 1: CCW, Invertierte Zählrichtung

3.4.2 Konfiguration Inkrementalgeber an SinCos Interface

Wird ein Inkrementalgeber an der SinCos Schnittstelle angeschlossen und betrieben, so erfolgt die Konfiguration beim SinCos Interface (siehe 3.6.2).

3.4.3 Ist-Werte Inkrementalgeber

Pfad im Inco-Tree zu den Ist-Werten des Inkrementalgebers: `Ctrl.Actual.Encoder`

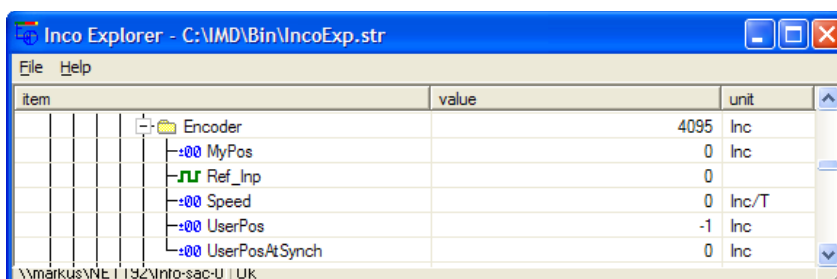


Abb 16: Ist-Werte Encoder

3.5 Resolver

Die Position des Resolvers wird fix mit 16-Bit eingelesen.

3.5.1 Konfiguration Resolver

Pfad im Inco-Tree zum Resolver: `Ctrl.MotorConfig.Resolver`

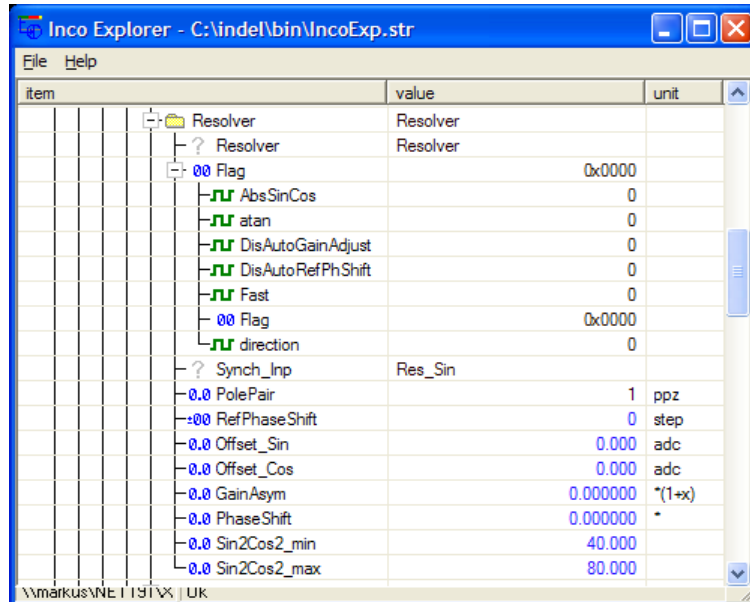


Abb 17: Resolver

PolePair	Pool Paar Zahl						
Sin2Cos2_max	Maximum von Sinus ² Cosinus ² . Defaultwert: 80						
Sin2Cos2_min	Minimum von Sinus ² Cosinus ² . Defaultwert: 40						
DisAutoGainAdjust	Die Amplitude des Resolver Generators (Referenzausgang) wird vom Regler automatisch so eingestellt, dass Sin2Cos2 möglichst im Bereich von 60 liegt. Mit diesem Flag wird die Korrektur ausgeschaltet und dient hauptsächlich nur zur Fehleranalyse.						
DisAutoRefPhShift	Die Phasenverschiebung des Resolver Generators (Referenzausgang) zu den gemessenen Sin und Cos Signalen wird vom Regler automatisch gemessen und korrigiert. Mit diesem Flag wird die Korrektur ausgeschaltet und dient hauptsächlich nur zur Fehleranalyse.						
Flag	<table> <tr> <td>Direction</td> <td>Drehrichtung des Resolvers. (Trotzdem richtig verdrahten!)</td> </tr> <tr> <td>Fast</td> <td>0: Tiefpassfilter beim Resolver mit einer Grenzfrequenz von ca. 400Hz 1: Tiefpassfilter beim Resolver mit einer Grenzfrequenz von ca. 600Hz</td> </tr> <tr> <td>atan</td> <td>0: keine Wirkung 1: Tiefpassfilter ausgeschaltet. Es wird direkt der arctang von der Sinus und Cosinus Spur berechnet und verwendet.</td> </tr> </table>	Direction	Drehrichtung des Resolvers. (Trotzdem richtig verdrahten!)	Fast	0: Tiefpassfilter beim Resolver mit einer Grenzfrequenz von ca. 400Hz 1: Tiefpassfilter beim Resolver mit einer Grenzfrequenz von ca. 600Hz	atan	0: keine Wirkung 1: Tiefpassfilter ausgeschaltet. Es wird direkt der arctang von der Sinus und Cosinus Spur berechnet und verwendet.
Direction	Drehrichtung des Resolvers. (Trotzdem richtig verdrahten!)						
Fast	0: Tiefpassfilter beim Resolver mit einer Grenzfrequenz von ca. 400Hz 1: Tiefpassfilter beim Resolver mit einer Grenzfrequenz von ca. 600Hz						
atan	0: keine Wirkung 1: Tiefpassfilter ausgeschaltet. Es wird direkt der arctang von der Sinus und Cosinus Spur berechnet und verwendet.						

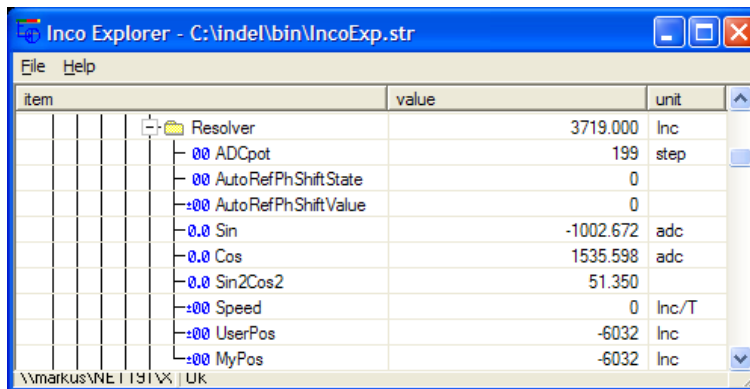
Wird der Tiefpassfilter höher gestellt oder abgeschaltet können auch mehr Störungen auftreten. Es sollte immer mit einem möglichst tiefen Filter gearbeitet werden.

Hinweis

Für den Betrieb mit Resolver darf die Abtastrate im Servo-Drive auf max. 12kHz eingestellt werden. Ansonsten werden die Pegel der Sinus und Cosinus Werte zu stark gedämpft.

3.5.2 Ist-Werte Resolver

Pfad im Inco-Tree zu den Ist-Werten vom Resolver: `Ctrl.Actual.Resolver`



item	value	unit
Resolver	3719.000	Inc
00 ADCpot	199	step
00 AutoRefPhShiftState	0	
00 AutoRefPhShiftValue	0	
0.0 Sin	-1002.672	adc
0.0 Cos	1535.598	adc
0.0 Sin2Cos2	51.350	
00 Speed	0	Inc/T
00 UserPos	-6032	Inc
00 MyPos	-6032	Inc

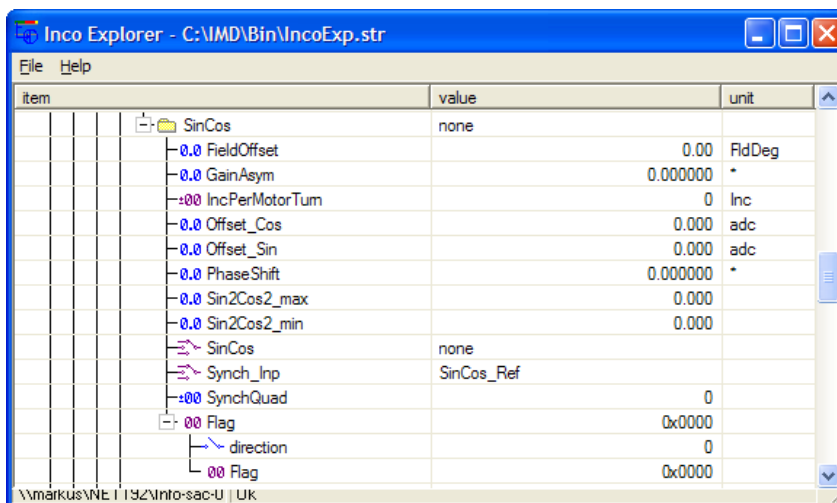
Abb 18: Ist-Werte Resolver

Bei „Resolver-Fehler“ muss zuerst der Wert Sin2Cos2 überprüft werden. Dieser muss innerhalb der konfigurierten Limiten sein.

3.6 SinCos

3.6.1 Konfiguration SinCos

Pfad im Inco-Tree zum SinCos: `Ctrl.MotorConfig.SinCos`



item	value	unit
SinCos	none	
0.0 FieldOffset	0.00	FldDeg
0.0 GainAsym	0.000000	*
00 IncPerMotorTurn	0	Inc
0.0 Offset_Cos	0.000	adc
0.0 Offset_Sin	0.000	adc
0.0 PhaseShift	0.000000	*
0.0 Sin2Cos2_max	0.000	
0.0 Sin2Cos2_min	0.000	
SinCos	none	
Synch_Inp	SinCos_Ref	
00 SynchQuad	0	
00 Flag	0x0000	
direction	0	
00 Flag	0x0000	

Abb 19: SinCos

IncPerMotorTurn Anzahl Inkremente pro Motor-Umdrehung
Anzahl Perioden * 1024. Die Auflösung der Sinus Cosinus Werte beträgt 10Bit.

Beispiel: SinCos Geber mit 2048 Strichen: $2048 * 1024 = 2'097'152$ Inc/T

Sin2Cos2_max Maximum von $\text{Sinus}^2 \text{Cosinus}^2$. Defaultwert: 80

Sin2Cos2_min Minimum von $\text{Sinus}^2 \text{Cosinus}^2$. Defaultwert: 20

Flag Direction Drehrichtung des SinCos Gebers

3.6.2 Konfiguration Inkrementalgeber an SinCos Interface

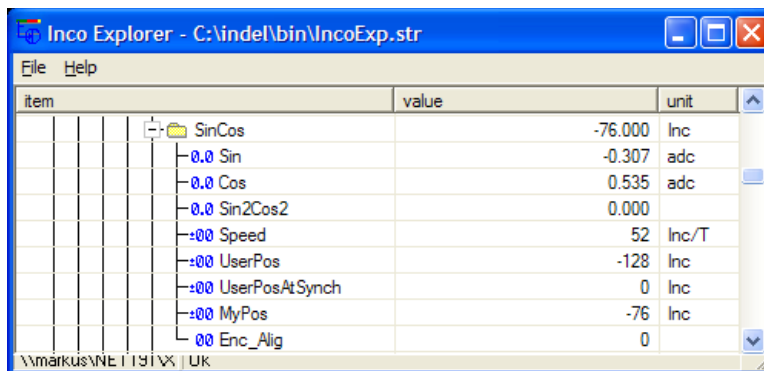
Wird ein Inkrementalgeber an der SinCos Schnittstelle angeschlossen und betrieben, so erfolgt die Konfiguration beim SinCos Interface.

<code>IncPerMotorTurn</code>	Anzahl Inkremente pro Motor-Umdrehung * 1024 (Ohne 4 Quadranten Auflösung)
<code>Sin2Cos2_min</code>	Minimum von $\text{Sinus}^2 \text{Cosinus}^2$. Defaultwert: 20
<code>Sin2Cos2_max</code>	Maximum von $\text{Sinus}^2 \text{Cosinus}^2$. Defaultwert: 400
Flag	Direction Drehrichtung des SinCos Gebers

Achtung: Da nun der Inkrementalgeber auf dem SinCos konfiguriert ist, muss für Positionsregelung sowie GinLink Feedback auch der SinCos definiert werden.

3.6.3 Ist-Werte SinCos

Pfad im Inco-Tree zu den Ist-Werten vom SinCos: `Ctrl.Actual.SinCos`



item	value	unit
SinCos	-76.000	Inc
0.0 Sin	-0.307	adc
0.0 Cos	0.535	adc
0.0 Sin2Cos2	0.000	
+00 Speed	52	Inc/T
+00 UserPos	-128	Inc
+00 UserPosAtSynch	0	Inc
+00 MyPos	-76	Inc
00 Enc_Align	0	

Abb 20: Ist-Werte SinCos

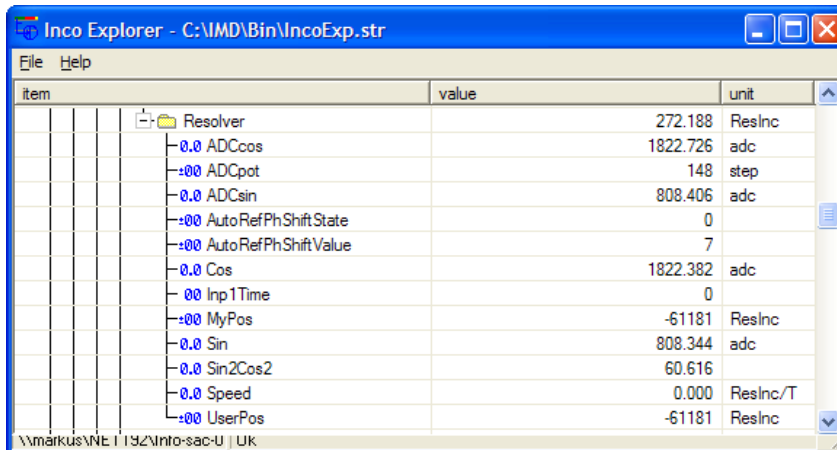
Bei „Resolver-Fehler“, bzw. „SinCos-Fehler“ muss zuerst der Wert Sin2Cos2 überprüft werden. Dieser muss innerhalb der konfigurierten Limiten sein.

3.7 Sinus Cosinus / Resolver Pegel überprüfen

Inco-Pfad zu den Istwerten von Resolver/SinCos wechseln:

Ctrl.Actual.Resolver

Ctrl.Actual.SinCos



item	value	unit
Resolver	272.188	ResInc
-0.0 ADCcos	1822.726	adc
-0.0 ADCpot	148	step
-0.0 ADCsin	808.406	adc
-0.0 AutoRefPhShiftState	0	
-0.0 AutoRefPhShiftValue	7	
-0.0 Cos	1822.382	adc
-0.0 Inp1Time	0	
-0.0 MyPos	-61181	ResInc
-0.0 Sin	808.344	adc
-0.0 Sin2Cos2	60.616	
-0.0 Speed	0.000	ResInc/T
-0.0 UserPos	-61181	ResInc

Abb 21: SinCos

Folgende Parameter müssen stimmen:

Einspeisung		Resolver		SinCos	
		Maximum	Minimum	Maximum	Minimum
ADCcos	bits	2048	-	2048	-
ADCsin	bits	2048	-	2048	-
ADCpot	bits	255	-	255	-
Sin2+Cos2	(V ²)	80	40	80	20

Sollten gewisse Parameter in der Nähe ihrer Grenzen liegen, kann dies eine Ursache sein für sporadische Fehler der Geber.

3.8 Auto Kommutierung

3.8.1 Konfiguration der Auto-Kommutierung

Inco-Pfad zur Auto Commutation: `Ctrl.MotorConfig.AutoCommutation`

Es stehen verschiedene Verfahren für die Auto-Kommutierung zur Verfügung:

<code>360deg FieldRotation</code>	Auto-Kommutierung mit 360° Felddrehung
<code>UVW pulse</code>	3 Spannungsimpulse
<code>Two-Phase Stepper</code>	Spannungsimpuls, Motor bewegt sich, auch geeignet für 3-phasige Motoren mit hängender Last
<code>Hiperface</code>	Digitales Interface, Motor bewegt sich nicht
<code>EnDat</code>	Digitales Interface, Motor bewegt sich nicht
<code>SSI</code>	Digitales Interface, Motor bewegt sich nicht

Flags

<code>ON_If_Ok=1</code>	Wenn die Kommutierung erfolgreich war, bleibt die Achse aktiv. Ansonsten wird die Endstufe inaktive geschaltet.
<code>Again=0</code>	Im normalen Betrieb wird die Achse nach der Kommutierung aktiv geschaltet. Die Betriebsart wechselt von <code>Commutation</code> nach <code>Active</code> ; <code>Again=0</code> Bei der Inbetriebnahme ist es nötig, die Kommutierung einige male zu wiederholen. Dazu wird das Flag auf <code>Again = 1</code> gesetzt, Damit bleibt die Betriebsart immer in <code>AutoCommutation</code> .
<code>Z_Axis_FieldOffset</code>	FieldOffset bei z-Achsen (siehe Kapitel 10.16.4)

Die Achse niemals aktivieren (weder im Simulations-Mode noch im Aktive-Mode) wenn die Kommutierung nicht einwandfrei funktioniert!

3.8.2 Auto-Kommutierung mit UVW Puls

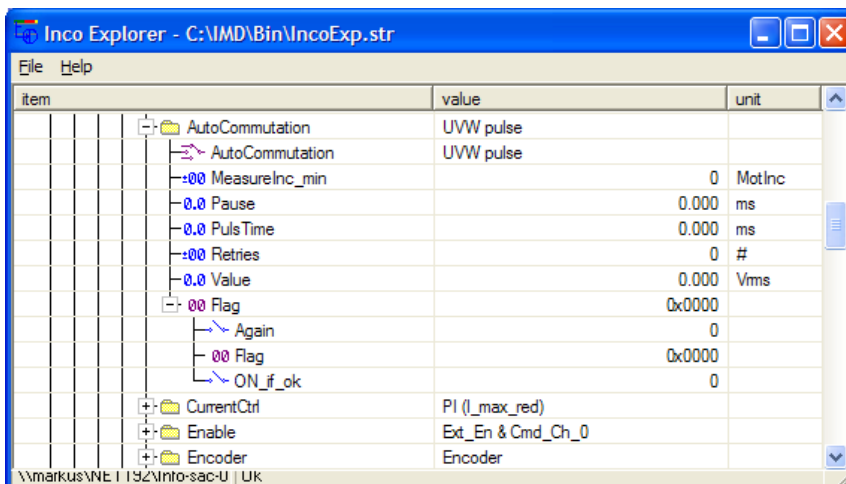


Abb 22: Auto Kommutierung UVW pulse

Das UVW Puls Verfahren erzeugt einen kurzen Spannungsimpuls auf jede Phase. Aus der entstandenen Bewegung wird der Feldoffset errechnet.

Bei dieser Kommutierungs-Methode bewegt sich der nur sehr wenig. Diese Methode ist geeignet für dynamische Motoren mit geringer Masse und geringer Reibung.

MeasureInc_min	20 ... 30	[MotInc]	Die Summe der drei Bewegungen von U, V, W muss grösser als MeasureInc_min sein, damit die Kommutierung erfolgreich ist.
Pause	20 ... 100	[ms]	Pause zwischen den einzelnen Impulsen
PulsTime	1 ... 5	[ms]	Puls-Dauer der drei Impulse
Retries	0 ... 3	[#]	Retries bei nicht erfolgreichen Kommutierungs-Versuchen.
Value=1 ... 3		[Vrms]	Spannungswert für den Impuls

3.8.3 Auto-Kommutierung mit Two-Phase Stepper Methode

Diese Kommutierungsart ist für Schrittmotoren mit Feedback sowie für 3-phasige Motoren mit hängender Last geeignet. Der Motor bewegt sich ruckartig um bis zu 60 Feldgrade.

Für Motoren mit Inkrementalgeber mit geringer Auflösung (500 Striche pro Umdrehung) ist diese Methode ebenfalls geeignet.

Der Spannungs-Impuls wird für die Zeit `PulsTime` in den Motor eingepreßt. Dadurch wird der Rotor in eine definierte Position gezogen.

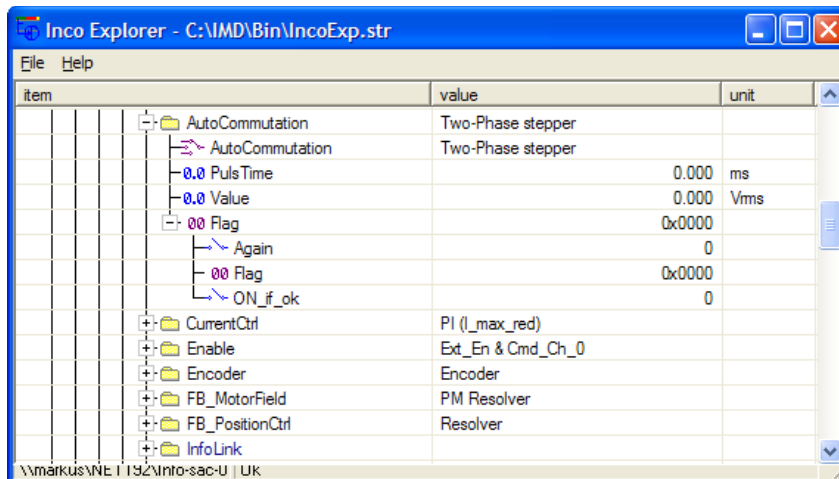


Abb 23: Auto Kommutierung Two-Phase Stepper

PulsTime 1000 [ms]
Puls-Dauer für den Spannungs-Impulse

Value 0.5 ... 3 [Vrms]
Spannungswert für den Impuls. Die Spannung so wählen, dass $I_{MAX}/2 \dots I_{MAX}$ fließt.

3.8.4 Auto-Kommutierung mit Absolut Encodern

Bei Achsen die sich vor dem Aktivieren nicht bewegen dürfen, oder bei Achsen die mit einer Festhaltebremse gehalten werden, kann die Auto-Kommutierung mit einem Absolut-Encoder durchgeführt werden.

- Hiperface
- EnDat
- SSI Synchrones Serielles Interface

3.8.5 Auto-Kommutierung mit 360° Felddrehung

Diese Kommutierungsart ist die präziseste von allen und sollte wenn immer möglich eingesetzt werden. Der Motor bewegt sich um 360 Feldgrade. Bei Motoren mit einem Polpaar entspricht dies eine Motor-Umdrehung, bei Motoren mit 10 Polpaaren entspricht dies 0.1 Umdrehungen.

Diese Methode ist für Achsen mit grosser Last und grosser Reibung aber auch für hochdynamische Achsen geeignet.

Flag.Direction	0, 1 []	Mit dem Direction Flag kann bestimmt werden, in welche Richtung das Feld gedreht werden soll (positive oder negative Richtung)
Flag.not_unwind	0, 1 []	Nach der Kommutierung kann die Achse für eine Zeit von ca. 380ms stromlos geschaltet werden. Dadurch wird verhindert, dass der Motor überlastet wird, falls die Kommutierung gegen eine mechanische Begrenzung ausgeführt wird. Dieses Flag NICHT setzen damit die Entlastung (unwinding) durchgeführt wird. Siehe Kapitel 3.8.6 Flag „Not Undwind“ (360° Kommutierung).
Max_Delta	10 ... 20 [deg]	Nach der Auswertung der Positions-Informationen darf die Differenz aller Offsetwinkel der ausgemessenen Segmente den Max_Delta Wert nicht übersteigen.
TurnTime	1000 ... 4000 [ms]	Zeit für die Felddrehung, es werden 360° in positive und 360° in negative Richtung abgefahren. Die Zeit sollte den Polpaaren des Motors angepasst werden. Motoren mit nur einem Polpaar mit mindestens 2000ms drehen lassen. Bei Motoren mit grosser Last ebenfalls längere Zeiten konfigurieren.
Value	1 ... 3 [Vrms]	Spannungswert für die Felddrehung. Die Spannung so wählen, dass I_{NENN} nicht überschritten wird.

3.8.6 Flag „Not Undwind“ (360° Kommutierung)

Mit dem Flag `not_unwind` kann das Verhalten unmittelbar nach der 360° Kommutierung beeinflusst werden.

Ist das Flag nicht gesetzt wird nach der 360° Felddrehung eine Pause von 380ms eingelegt, während dieser Zeit wird der Wirkstrom auf Null gesetzt. Nach der Pause wird die Positions-Regelung aktiviert und es stellt sich ein bestimmter Haltestrom ein.

Für hängende Z-Achsen ist dieser Mode (`not_unwind=0`) nicht geeignet, da die Achse während der Pause nicht gehalten werden kann.

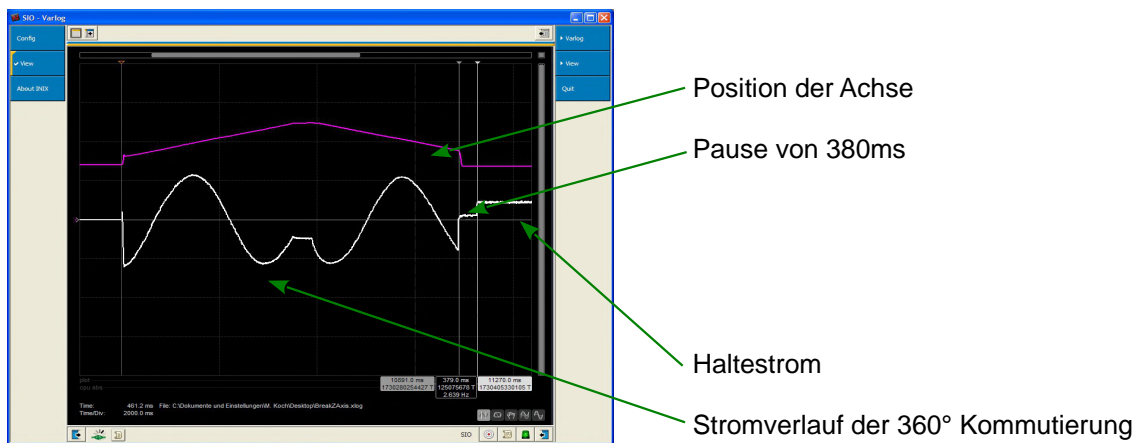


Abb 24: Flag `not_Unwind` nicht gesetzt

Ist das Flag gesetzt, entfällt die Pause für die Entlastung der Achse. Falls die Achse nach dem Kommutieren gegen eine mechanische Begrenzung drückt, wird unter Umständen nach der Kommutierung ein erhöhter Strom in den Motor eingepreßt. Siehe Abb 25.

Der Mode mit gesetztem `not_unwind` Flag setzt voraus, dass die I²t-Regelung optimal eingestellt ist. Ansonsten besteht die Gefahr, dass der Motor überlastet wird.

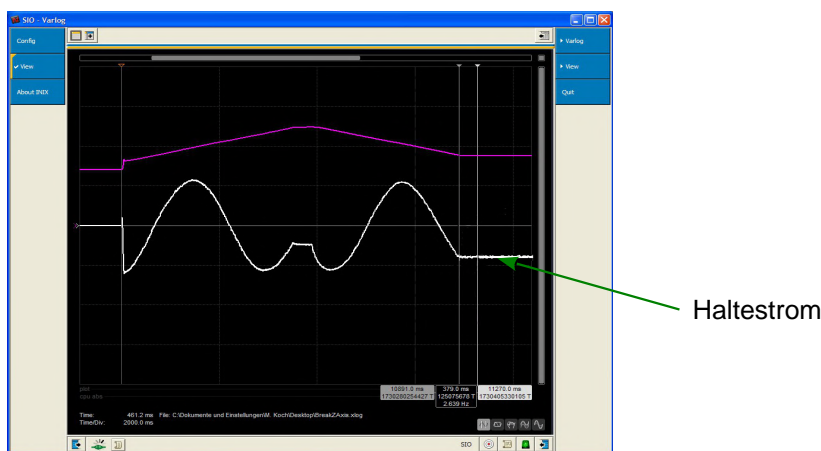


Abb 25: Flag `not_Unwind` gesetzt

3.8.7 Auto-Kommutierung mit Hall-Sensoren für Maxon Motoren

HallInp_Seq Erwartete Abfolge der Hall-Sensor Eingangssignale

HallInp_0 Bit-Nummer des ersten Hall-Sensor Einganges. Die drei Hall-Sensoren müssen hintereinander angeschlossen werden:
z.B. Eingang 0, 1, 2

- Hall-Sensor 1 auf Eingang Inp Nr verdrahten
- Hall-Sensor 2 auf Eingang Inp Nr + 1 verdrahten
- Hall-Sensor 3 auf Eingang Inp Nr + 2 verdrahten

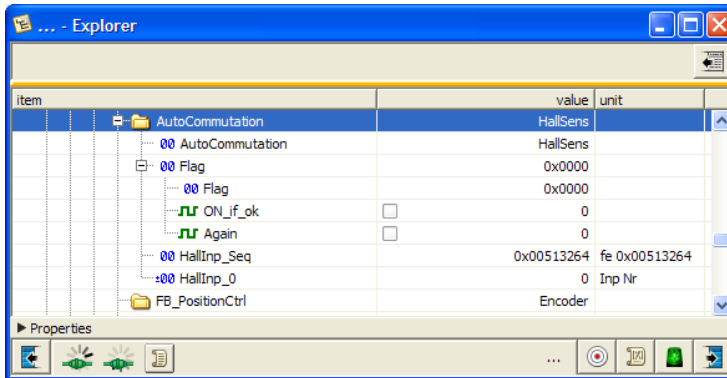
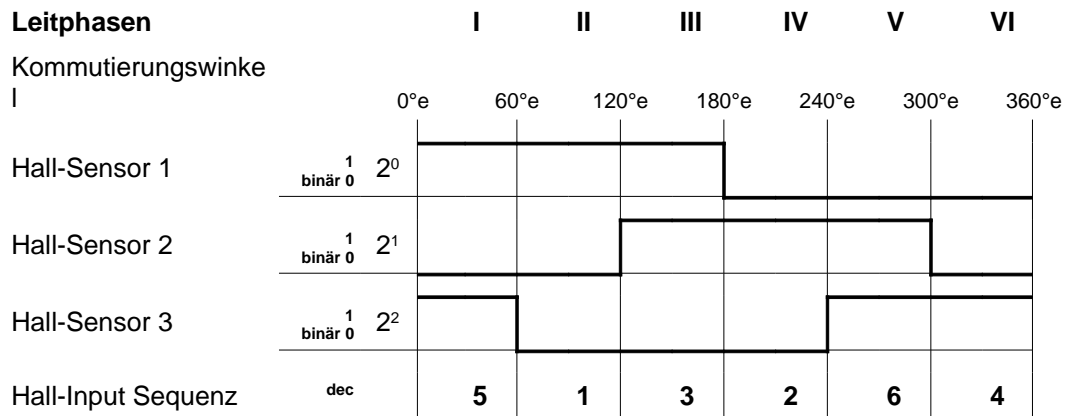


Abb 26: Hall-Sensor Kommutierung

Block-Kommutierung



Der Kommutierungswinkel wird in Feldgraden (°e) angegeben. Achtung: bei Motoren mit mehr als einem Polpaar stimmen Grad an der Motor-Welle nicht mit den Feldgraden überein.

Beispiel:

Bei einem Motor mit 7 Polpaaren dreht das Motor-Feld 7 x 360°e = 2520°e bei einer Umdrehung an der Welle.

Konfiguration der Block-Kommutierung

Diese Boards bieten die Block-Kommutierung mit Hallsensoren an: MAX2, MAX4, AX-4x2, AX-4x4.

Mit Hilfe der Hallsensoren kann eine erste grobe Kommutierung gemacht werden (ca $\pm 30^\circ$ genau).

Zusätzlich wird ein genauer Kommutierungs-Wert bei der Encoder-Ref Marke abgelegt. Beim ersten Überfahren der Marke wird dann automatisch die endgültige Kommutierung ermittelt. Es müssen also zwei Kommutierungen vorbereitet werden.

Vorbereitung

Nehmen Sie den Motor vollständig in Betrieb. Folgende Punkte müssen stimmen bevor die Block-Kommutierung konfiguriert werden kann:

- Der Drehsinn des Motors muss stimmen:
Motorwicklung 1 auf U, Motorwicklung 2 auf V, Motorwicklung 3 auf W
- Die Drehrichtung von Motor und Gebersystem muss stimmen, d.h. die Direction Flags vom Motor und Encoder in der Motor-Konfiguration müssen stimmen
- Hallsensor-Eingänge invertieren wenn keine Open-Kollektor Ausgänge am Hall-Sensor vorhanden sind:

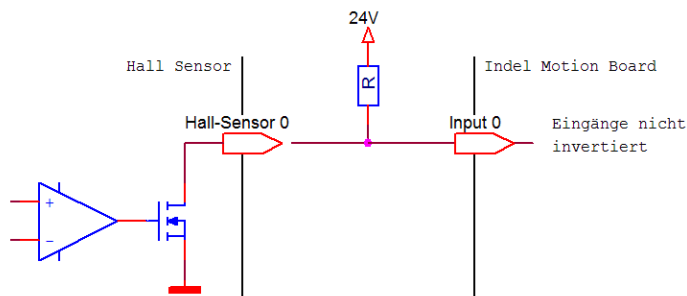


Abb 27: Beschaltung mit nicht invertierten Eingängen

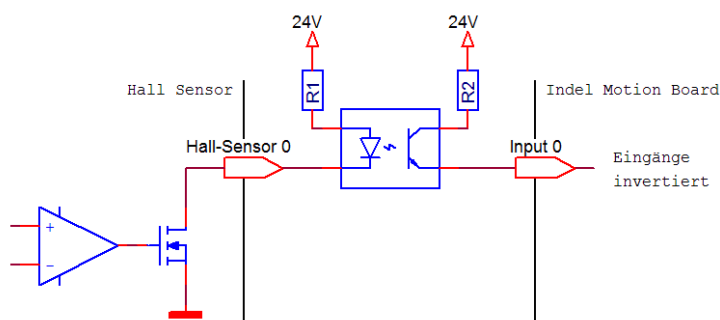


Abb 28: Beschaltung mit invertierten Eingängen

Die Eingänge können in der dt2-Konfiguration invertiert werden, dazu das „Inverted“ Bit setzen.

Vorgehen

1. Präzise Kommutierung bei Synch-Marke (Nullimpuls)

Dazu den Feld Offset für den Motor möglichst genau bestimmen:

z.B. mit dem Mode 360deg FieldRotation und anschliessendem Feintuning im Strom- oder Spannungsmode gem. Manual) Siehe Kapitel: 10.16.3 Resolver Offset von Hand abgleichen

- Autokommutierung z.B. mit 360° Methode, Feintuning von Hand

2. Feld Offset bei Referenzmarke bestimmen

- Setzte `Ctrl.Actual.FB_MotorField.SynchDone = 0`
- Setzte `Ctrl.MotorConfig.FB_MotorField.FieldOffset_at_Ref = -1`
- Setzte Flag `Ctrl.MotorConfig.FB_MotorField.Flag.FieldSynchWithRefInp = 1`

- den Motor über die Synch-Marke (Nullimpuls) bewegen, entweder von Hand oder den Motor langsam drehen lassen.

-> das Flag `Ctrl.Actual.FB_MotorField.SynchDone` wird 1

-> in `Ctrl.MotorConfig.FB_MotorField.FieldOffset_at_Ref`

wird der genaue Fieldoffset abgelegt und kann so gebrannt werden.

3. Hallensoren Sequenz auslesen

Je nachdem wie der Motor angeschlossen ist, bzw. wie die Drehrichtung konfiguriert ist, ändert die Abfolge der Hallensoren.

- den Kommutierungsmodus `Ctrl.MotorConfig.AutoCommutation = HallSens` wählen
- Hallensoren-Eingänge konfigurieren: `Ctrl.MotorConfig.AutoCommutation.HallInp_0 = Inp_Nr`

Die drei Hallensoren müssen direkt hintereinander angeschlossen werden.

- Den Motor im Test-Mode `Field_Rotaion` laufen lassen gleichzeitig mit dem Varlog folgende Parameter aufzeichnen:
 - `Ctrl.Actual.FB_MotorField`
 - `Ctrl.Actual.AutoCommutation.HallSens_1`
 - `Ctrl.Actual.AutoCommutation.HallSens_2`
 - `Ctrl.Actual.AutoCommutation.HallSens_3`
- Ab -2048 Inc e (Feldwinkel) die Sequenz der Eingänge von links nach rechts auslesen: innerhalb einer Feldumdrehung sollten, je nach Motor Direction Flag, eine der beiden folgenden Sequenzen erscheinen:
 - 0x0062'3154
 - 0x0051'3264

In der Hall-Input Sequenz muss für die Beschaltung U, V, W – Motorwicklung 1, 2, 3 immer folgender Wert eingetragen werden:

`Ctrl.MotorConfig.AutoCommutation.HallInp_Seq = 0x0051'3264`

- Den Parameter `Ctrl.MotorConfig.AutoCommutation.FieldOffset = 0` setzen.
- Setzte `Ctrl.Actual.FB_MotorField.SynchDone = 0`

Hallsensor Sequenz

Motor-Direction Flag: 1
 Motor-Anschlüsse: U, V, W – Motorwicklung 1, 2, 3

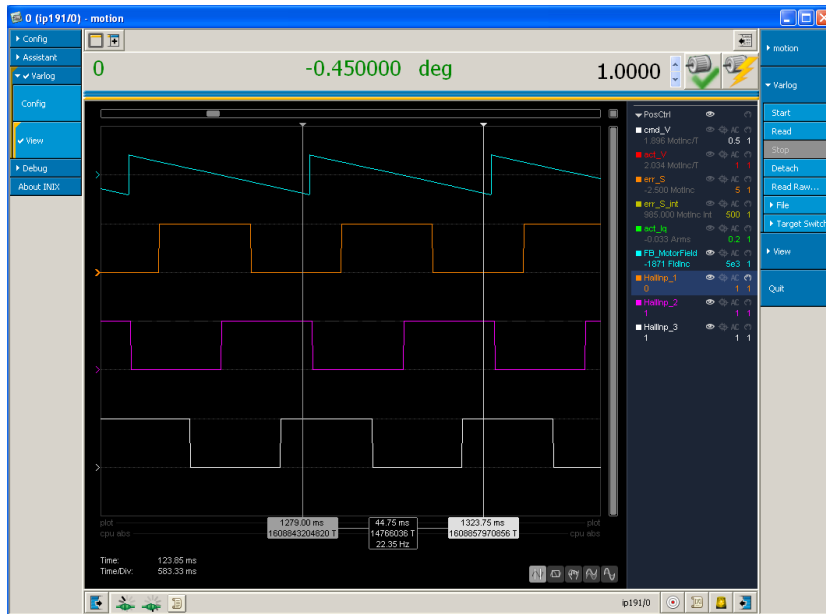


Abb 29: Hallsensor Sequenz, Normdrehrichtung CCW

Motor-Direction Flag: 0
 Motor-Anschlüsse: U, V, W – Wicklung 1, 2, 3

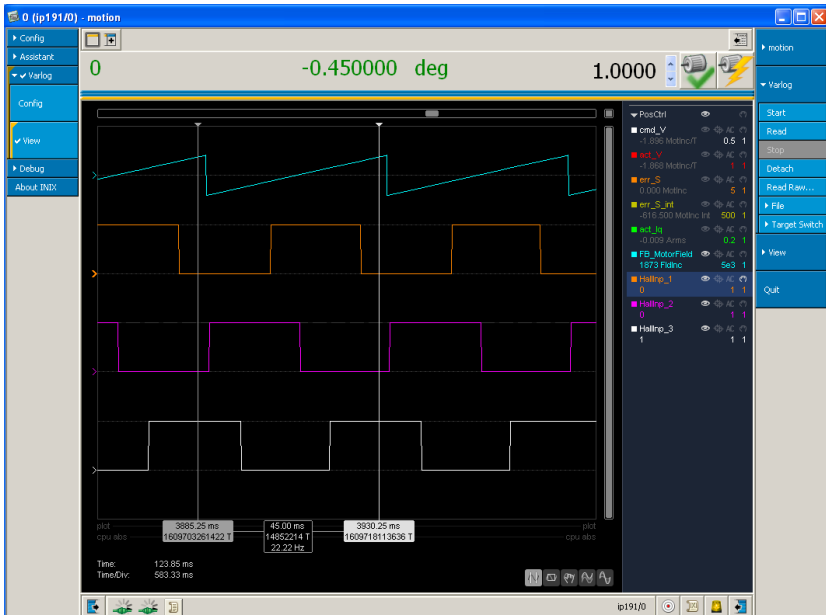


Abb 30: Hallsensor Sequenz, Normdrehrichtung CW

- | | | |
|------------|----------|---------------------------|
| Channel 1: | hellblau | Feldwinkel in Inkrementen |
| Channel 2: | orange | Hallsensor Eingang 1 |
| Channel 3: | pink | Hallsensor Eingang 2 |
| Channel 4: | weiss | Hallsensor Eingang 3 |

4. Testen der Block-Kommutierung

- Um die Block-Kommutierung zu testen muss das Flag vorübergehend auf Null gesetzt werden:
`Ctrl.MotorConfig.FB_MotorField.Flag.FieldSynchWithRefInp = 0`
- Motor-Parameter ins Flash-Prom brennen
- Schalten sie mehre Male und mit verschiedenen Startpositionen Aus/Ein
Die Achse muss im Stande sein, damit jeweils mindestens die Synch-Marke zu erreichen. Evtl. muss mit reduzierten PID-Einstellungen gefahren werden.

Vorsicht: Wenn die Kommutierung nicht stimmt, fließen hohe Ströme im Motor und der Motor dreht evtl. in die falsch Richtung! Um den Motor zu schützen kann der I_{MAX} reduziert werden. Der Wert für I^2t -down kann auf 0.98 eingestellt werden.

- Wenn alles funktioniert, nicht vergessen, das Flag wieder richtig zu stellen und brennen.
`Ctrl.MotorConfig.FB_MotorField.Flag.FieldSynchWithRefInp = 1`

Mögliche Probleme

Wenn die Hallsensoren nicht genau 60°e Sektoren anzeigen, verschlechtert sich die Genauigkeit der Kommutierung sehr stark. Dadurch kann die Strecke bis zur Synch-Marke evtl. nicht zurückgelegt werden, ohne dass der Motor in Resonanz gerät.

Um diesem Problem entgegen zu wirken kann ein angepasster PID-Parameter-Satz verwendet werden um die erste Synch-Marte zu erreichen:

- kP reduzieren auf 50 ... 80%
- Pos_Int_Max reduzieren auf 1000 ... 5000

Um die Gleichmässigkeit der Segmente zu überprüfen sollte immer ein Log aufgezeichnet werden mit den einzelnen Hallsensor-Eingängen (HallInp_0, 1, 2) und dem Feldwinkel. Die Achse dabei mit konstanter Geschwindigkeit verfahren.

Alle Signal-Flanken müssen etwa den selben Abstand haben. In Abb. 29 und 30 sind die Segmente sehr unregelmässig.

3.8.8 Ist-Werte Auto-Kommutierung

Inco-Pfad zu den Ist-Werten der Auto-Kommutierung: `Ctrl.Actual.AutoCommutation`
 Abhängig von der Auto-Kommutierungs Methode werden die Ist-Werte wie folgt eingeblendet:

UVW-Methode

item	value	unit
AutoCommutation	180.000	FldDeg
+00 Status	0	
+00 Ok	0	0/1
0.0 S_U	0.000	MotInc
0.0 S_V	0.000	MotInc
0.0 S_W	0.000	MotInc
0.0 S_total	0.000	MotInc
+00 LoopCnt	0	

Abb 31: Ist-Werte Auto-Kommutierung UVW

- Status** Status der Auto-Kommutierung
- OK** wenn die Kommutierung erfolgreich war, ist dieser Wert 1, ansonsten 0
- S_U, S_V, S_W** Weg der die einzelnen U,V,W Impulse erzeugen
- S_Total** gesamte Strecke die bei der Kommutierung zurückgelegt worden ist. Wenn das `S_Total` den Wert von `MeasureInc_min` übersteigt, ist die Kommutierung erfolgreich.
- LoopCount** Anhand vom Loop Counter kann festgestellt werden, wie viele Versuche die Kommutierung benötigt hat bis sie erfolgreich war. Siehe `Retries`.

Two-Phase Stepper und Absolut-Geber Methode

Diese beiden Verfahren enden immer erfolgreich.

360° Feldrotation Methode

item	value	unit
AutoCommutation	181.934	FldDeg
+00 Status	0	
+00 Ok	1	0/1
+00 best_sector_fw	1	
+00 best_sector_bw	5	
0.0 delta_fw	19.308	
0.0 delta_bw	19.512	
0.0 act_field	6233.694	
+00 uLoopCnt	0	

Abb 32: Ist-Werte 360° Kommutierung

Der Feldwinkel unter `AutoCommutation` wird bei erfolgreicher Kommutierung für die Feldregelung übernommen.

3.9 Stromregler: Current Control

3.9.1 Stromregler Varianten

Der Sollwert für den Stromregler kommt immer aus dem vorgelagerten PID-Regler (Geschwindigkeit- bzw. Positions-Regler). Dieser Sollwert kann mit folgenden Varianten angepasst werden.

Ausgang 1 entspricht dem Bit `I_Red` (Stromreduzierung). Das Bit `I_Red` wird vom Feldbus-Controller über den Feldbus gesetzt.

none

PID # Cmdlq Ausgang 1 = 0: Sollwert für Strom-Regler ist Ausgang von PID Regler Ausgang 1 = 1: Sollwert ist Ausgang von PID Regler, unterlagert ist die Drehmomentkurve `Cmdlq`. Mit dieser Funktion kann eine beliebige Drehmoment-Kurve gefahren werden.

PI Sollwert für Strom-Regler ist Ausgang von PID Regler (Postions-Regler), Ausgang 1 wird nicht berücksichtigt

PI (I_max_red) Ausgang 1 = 0: Stromregler begrenzt auf `I_max` Ausgang 1 = 1: Stromregler regelt auf `I_red` -> konstantes Drehmoment **Default Einstellung**

PID (Cmdlred) Ausgang 1 = 0: Sollwert für Strom-Regler ist Ausgang von PID Regler Ausgang 1 = 1: Stromregler begrenzt auf Drehmomentkurve `I_red`, Betriebsart "Ired"

PID + Cmdlq Dem Sollstrom wird die Kurve `Cmdlq` überlagert (Vorsteuerung).

Cmdlq Stromkurve einprägen, ohne Positionsregelung

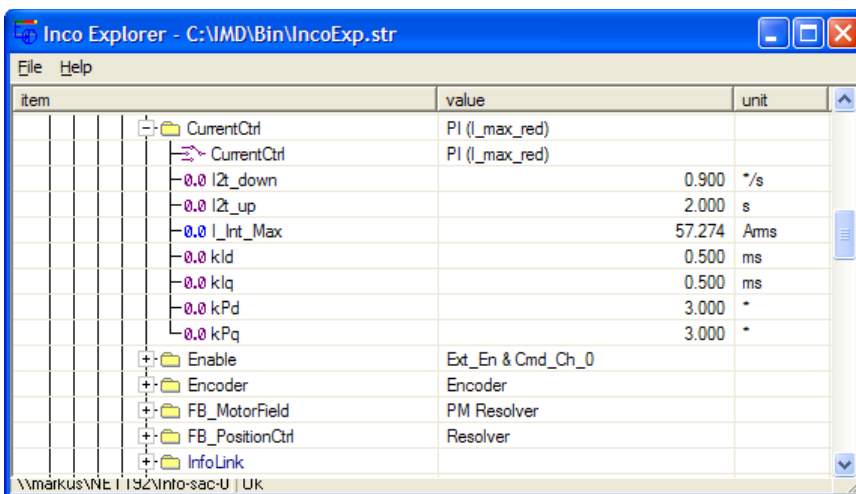


Abb 33: Stromregler

3.9.2 Stromregler Parameter

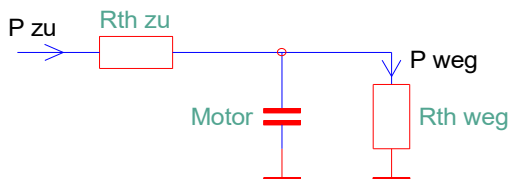
I_Int_Max [Arms]	Begrenzung des I-Anteiles im Strom-Regler. Dimensionierung: $I_Int_Max = 3 \text{ mal } I_{MAX}$
Überlastschutz I2t	
I2t [%]	Gibt an, wie viel der Verlustleistung, die der Motor aufnimmt, innerhalb der vorgegebenen Zeitkonstante (I2t_down) wieder abgegeben werden kann.
I2t_up_run: [s@I_max]	Zeitkonstante für die aufgenommene Verlustleistung des Motors während der Motor in Bewegung ist.
I2t_up_halt: [s@I_nom]	Zeitkonstante für die aufgenommene Verlustleistung des Motors während der Motor stillsteht.
I2t_down: [*/s]	Entladeverhalten für die gespeicherte Wärme im Motor (Wärmeabgabe über Gehäuse, Kühlung, usw).
kId [ms]	I-Anteil von Blindstrom-Regelung
kPd [*]	P-Anteil von Blindstrom-Regelung
kIq [ms]	I-Anteil von Wirkstrom-Regelung
kPq [*]	P-Anteil von Wirkstrom-Regelung

3.9.3 I²t Regelung

Um den Motor vor Überlast zu schützen, wird die Verlustleistung, die der Motor aufnimmt, integriert. Von dieser Summe wird stetig die Verlustleistung abgetragen, die der Motor über das Gehäuse und über allfällige Lüftungen abgibt. Der verbleibende Wert darf eine gewisse Schwelle nicht überschreiten.

Wenn kein Temperatur-Sensor in der Motorwicklung vorhanden ist, ist die I²t Regelung der einzige Schutz gegen thermische Überlast für den Motor!

Ersatzschaltbild für P_t



Defaultwerte

I ² t_up_run	[s]	0.5 ... 2	für Motoren mit geringer Überlast Fähigkeit
I ² t_up_run	[s]	2 ... 4	für Motoren mit grosser Überlast Fähigkeit
I ² t_down	[*]	0.90 ... 0.95	rotative Motoren
I ² t_down	[*]	0.95 ... 0.98	Linearmotoren, Motoren mit schlechter Wärme Abgabe

Ladeverhalten

Wird für die Zeit I²t_up_run der Strom I_max in den Motor eingepreßt, ohne dass dieser gekühlt wird, beträgt der Wert für I²t = 100%. Dieser Wert gilt während sich der Motor bewegt.

Wird für die Zeit I²t_up_halt der Strom I_nom in den Motor eingepreßt, ohne dass dieser gekühlt wird, beträgt der Wert für I²t = 100%. Dieser Wert gilt während der Motor stillsteht.

Entladeverhalten

Der Wert für I²t_down beschreibt die Temperatur-Abnahme im Motor:

Bei einem Wert von I²t_down = 0.9 wird der Wert für I²t jede Sekunde 10% kleiner

Bei einem Wert von I²t_down = 0.98 wird der Wert für I²t jede Sekunde 2% kleiner

Berechnungsgrundlagen

Für die Berechnung der I²t Regelung wird der I_max vom Motor verwendet.

Je grösser I_max ist, desto langsamer steigt der I²t Wert an.

Ist der I_max sehr viel grösser als I_nenn (I_max / I_nenn > 5) kann dies zu einer Verfälschung der Berechnung des I²t Wertes führen, weil der Wert für I²t evtl. zu langsam ansteigt.

Linearmotoren

Bei Linearmotoren muss der Kühlung besondere Beachtung geschenkt werden. Da die Spulen häufig in Epoxy vergossen sind, ist die Abgabe der Wärme nicht optimal. Dadurch kann sich leicht ein Wärmestau in der Motorwicklung bilden, besonders wenn der Motor nicht oder nur sehr langsam bewegt wird. (Stillstand, hängende Lasten, ...)

Bei Linearmotoren, die nur sehr kleine Bewegungen ausführen, ist darauf zu achten, dass für jede Wicklung ein separater Temperatursensor vorhanden ist!

Die Werte für die I²t Regelung müssen gegebenenfalls empirisch bestimmt werden und auf die physikalischen Gegebenheiten wie Motor-Daten, Kühlung und Fahrprofile angepasst werden.

Genaue Daten für Überlastbetrieb sind immer beim Motor-Hersteller anzufragen. Bei Dauerbetrieb des Motors mit I²t = 100% darf die maximale Motor Temperatur nicht überschritten werden.

3.10 Extern Enable

3.10.1 Konfiguration Extern Enable Eingang

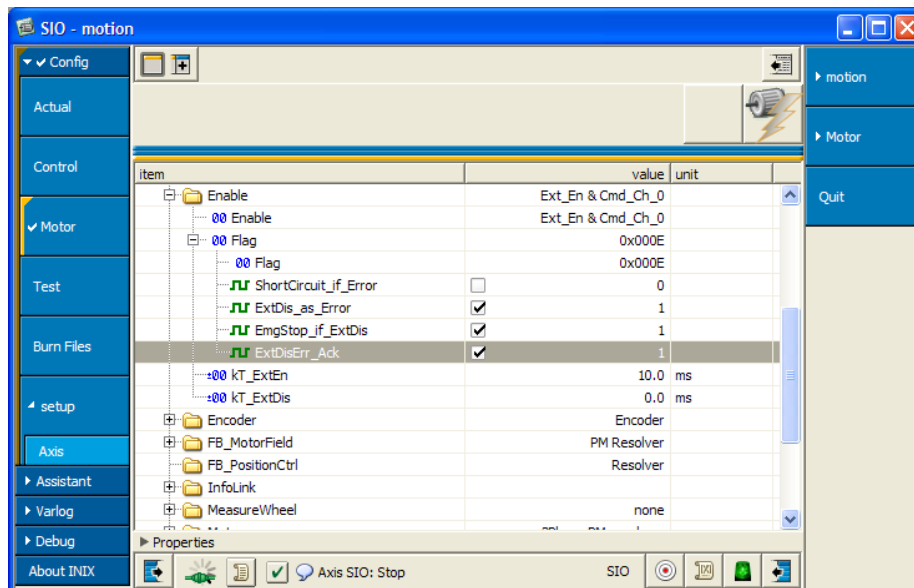


Abb 34: Extern Enable

Ext + Ch_0

Regler wird aktiv wenn Externe Freigabe vorhanden und "aktiv" über Sollwert-Kanal 0 anliegt

Ext + Ch_1

Regler wird aktiv wenn Externe Freigabe vorhanden und "aktiv" über Sollwert-Kanal 1 anliegt

Ext + (Ch_0#Ch_1)

Regler wird aktiv wenn Externe Freigabe vorhanden und "aktiv" über Sollwert-Kanal 0 oder Kanal 1 anliegt

kT_ExtDis

Sicherheitsfunktion um bei Abfallen der externen Freigabe den Motor noch abbremsen zu können. Die externe Freigabe wird um die Zeit kT_ExtDis verzögert weggenommen. Während dieser Zeit kann der Motor aus der Anwender-Software noch abgebremst werden. Damit kann die Ext-Freigabe in den Not-Aus Kreis aufgenommen werden.

kT_ExtEn

Software-Filter für die Externe-Freigabe (10ms) um Störungen durch Spikes zu eliminieren.

**Für STO (safe torque off) Kategorie 3 nach EN ISO 13849-1, müssen die beiden Safety Eingänge auf X100 verwendet werden.
Der Extern Enable ist ein nicht sicherer Eingang.**

Für die Konfiguration der Sicheren Impulssperre siehe Kapitel: 4 Safety Konfiguration.

3.10.2 Konfiguration der Not-Stop Brems-Rampe

Flags Mit diesen Flags wird bestimmt, wie sich die Achse abgeschaltet werden soll wenn die Regelung auf Fehler geht.

Für das Bremsen bei Not-Stop muss evtl. ein Bremswiderstand verwendet werden. Beim Bremsen speist der Motor die kinetische Energie in den Zwischenkreis zurück!

ShortCircuit_if_Error = 1

Der Regler schaltet die Achse wieder ein, nachdem die Regelung auf Fehler geht. Die Achse wird mit I_{max} gebremst bis zum Stillstand.

Wenn das Flag ShortCircuit_if_Error = 0 (nicht gesetzt) ist wird bei abfallen des Ext-Enable mit einer Not-Stop-Rampe abgebremst. Die Not-Stop-Rampe wird in der Achsenkonfiguration festgelegt.

Dangerous_OvrOn

Dieses Bit muss null sein.

ExtDis_as_Error

Wenn der Extern Enable ausgeschaltet wird, entsteht ein Fehler. Dieses Bit muss gesetzt sein für Betrieb mit STO!
Beim **GinLink** ist das Bit **immer** gesetzt und in der Konfiguration nicht mehr sichtbar

EmgStop_if_ExtDis

Not-Stop Bremsrampe wird gestartet, nachdem der Externe Enable ausgeschaltet ist. Dieses Bit muss gesetzt sein für Betrieb mit STO!
Beim **GinLink** ist das Bit **immer** gesetzt und in der Konfiguration nicht mehr sichtbar

ExtDisErr_Ack

Dieses Bit muss gesetzt werden um einen ungewollten Wiederanlauf zu verhindern. Dieses Bit ist gilt als nicht sicher!
Beim **GinLink** ist das Bit **immer** gesetzt und in der Konfiguration nicht mehr sichtbar

Maximaler Brems-Strom

Der maximal zulässige Strom für Kurzschlussbremsrampe berechnet sich wie folgt:

$$I_{MAX Drive} > 0.7 * \left(\frac{U_{CC}}{R_{PP}} \right)$$

I _{MAX Drive}	Maximal zulässiger Spitzenstrom des Drives	A
U _{CC}	Zwischenkreis-Spannung	V
R _{PP}	Wicklungswiderstand Phase-Phase	Ohm

Mit dem Flag ShorCircuit_if_Error kann der Drive zerstört werden! Es wird mit dem max. zulässigen Kurzschluss-Strom abgebremst. Die IGBTs/FETs können nach 1 ... 10 Brems-Rampen zerstört werden.

Dieses Flag sollte nur verwendet werden, wenn der Schutz der Mechanik über dem Schutz des Drives steht!

Diese Betriebsart ist von Garantie-Leistungen ausgeschlossen!

3.10.3 Ist-Werte Extern Enable

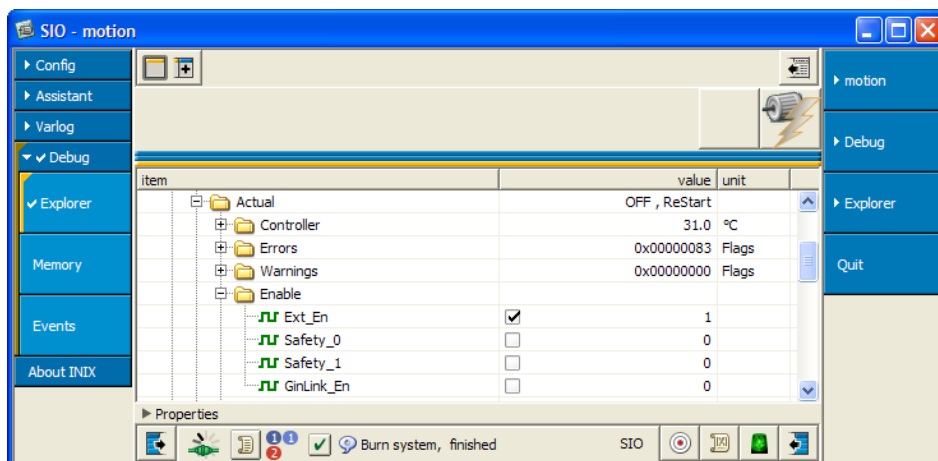


Abb 35: Ist-Werte Extern Enable

Ext_EN	Zustand Eingang +En, -En auf Stecker X15
Safety_0, Safety_1	Zustand Eingang 24V_R1, 24V_R2 auf Stecker X100
GinLink_EN	Zustand Software-Enable von GinLink Feldbus

3.11 Feedback Motor Field

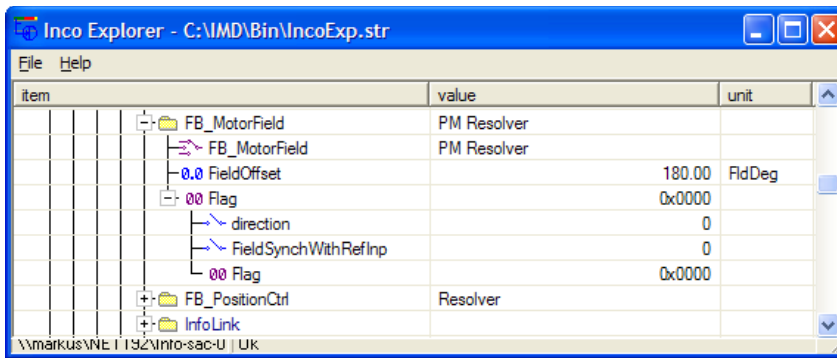


Abb 36: Motor Field Feedback

Feedback Motor Field

PM Resolver Messsystem ist Resolver an Permanent Magnet Synchron oder DC-Motor

AC I-model Encoder an AC-Asynchron Motor

PM SynCos Sinus Cosinus Interface

PM SynCos+Enc Sinus Cosinus Interface mit zusätzlichem Encoder zur Kompensation von überlagerten Relativ-Bewegungen.

PM Encoder Inkrementalgeber

Stepper without FB Feedbackloser Betrieb für Schrittmotoren

Flag.Direction Flag = 0: CW
 Flag = 1: CCW, Invertierte Zählrichtung
 Dieses Flag darf nie 1 sein!!

Für DC-Motoren ist keine Feldregelung nötig.

Für das Feld-Feedback wird ein Messsystem benötigt, das starr mit der Motor-Welle verbunden ist. Ein Massstab nach einer langen Spindel mit kleiner Steigung oder ein Messsystem nach einem Getriebe ist für die Feldregelung ungeeignet. In diesem Fall ist ein zweites Messsystem direkt auf der Motorwelle nötig.

Der Fehler des Feldoffsets darf nicht grösser als $\pm 10 \dots 15^\circ$ sein!

Rechen-Beispiel mit einer 30cm langen Stahlspindel mit 5mm Steigung:
 $a = 12\mu\text{m/K/m}$ Längenausdehnungskoeffizient von Stahl
 $dT = 10\text{K}$ Temperaturdifferenz
 $l = 0.3\text{m}$ Spindellänge
 $pp = 5$ Polpaare

$5\text{mm} \triangleq 360 \text{ Motor }^\circ \triangleq 360 * 5 \text{ Feld }^\circ$
 $5\text{mm} \triangleq 1800 \text{ Feld }^\circ$

Längenänderung bei $dT=10\text{K}$: $36\mu\text{m}$

$36\mu\text{m} \triangleq 12.96 \text{ Feld }^\circ$

→ Die $36\mu\text{m}$ Längenänderung der Spindel bei einer Temperatur-Differenz von 10K entspricht einem Feldwinkel von 13° . Damit ist die gesamte Toleranz für den Feldoffset bereits vergeben.

Hinzu kommen noch Abweichungen der Autokommutierungs-Messmethode: $\pm 5..10^\circ$,
 Ungenauigkeit der Spindel (Länge): $\pm 5..10\mu\text{m}$,
 Ungenauigkeit der Spindel (Steigung): $\pm 5..10\mu\text{m}$,

Der gesamte Fehler beträgt damit im schlimmsten Fall über 30 Feld ° . Bei dieser Abweichung kann die Achse nicht mehr optimal betreiben werden. Es fließen Blindströme, der Motor erwärmt sich und die Dynamik nimmt ab.

Bei noch grösseren Fehlern kann die Achse sich von selbst in Bewegung setzen!

3.12 Feedback Position Control



Abb 37: Position Control

Feedback für den Positions Relger im Servo-Drive:

Resolver

SinCos

Encoder

Stepper without FB

3.13 GinLink

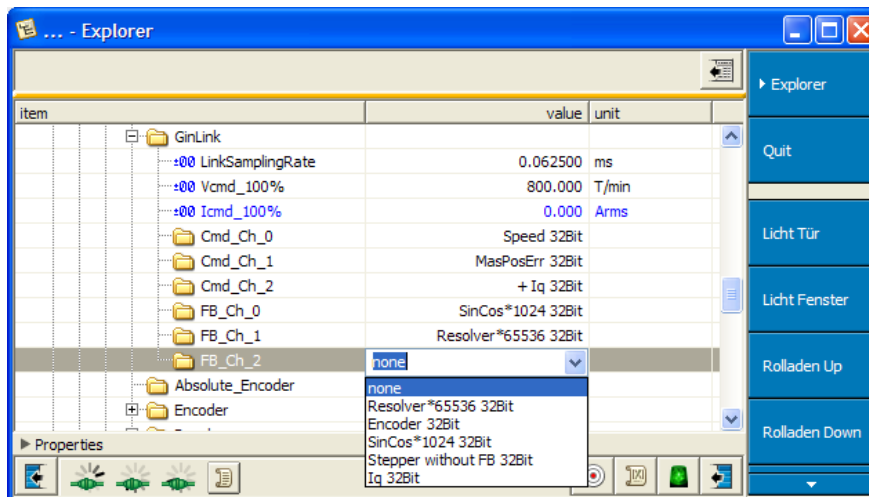


Abb 38: GinLink

Vcmd_100% Nenndrehzahl der Regelung. Normierungsfaktor

LinkSamplingRate Abtastrate des übergeordneten Reglers oder Bahnsteuerung im Feldbusmaster. Dieser Wert wird benötigt, um die Interpolation zwischen zwei Sollwerten vom Master auf die Abtastrate des SAC-Reglers abzustimmen typisch 8 bis 32kHz. Über den Feldbus sind folgende Abtastraten vorgesehen:

16 kHz	0.0625 ms
8 kHz	0.125 ms
4 kHz	0.25
2 kHz	0.5
1 kHz	1
0.5kHz	2 ms
0.25 kHz	4 ms

Icmd_100% Nennstrom bei Stromregelung. Normierungsfaktor

Sollwert-Kanäle

Insgesamt sind drei Soll-Wert Kanäle vorhanden: Cmd_Ch_0, Cmd_Ch_1, Cmd_Ch_2

Speed 32Bit Sollwert ist Geschwindigkeit (Für Positionsregelung wird das v im Regler integriert.)

Iq Limit 32Bit Limit Konstanter Sollstrom

Iq 32Bit Kurvenform für Sollstrom

+ Iq 32Bit Kurvenform für Strom-Vorhaltewert

MasPosErr 32Bit Schleppfehler aus übergeordneter Positionsregelung

Ist-Wert Kanäle

Insgesamt sind drei Ist-Wert Kanäle vorhanden:

FB_Ch_0: Standardbelegung für Motoren Feedback, Positionserfassung auf dem SAM

FB_Ch_1:

FB_Ch_2: Standardbelegung für Wirkstrom Iq

Auf allen Kanälen 32 Bit Werte übertragen werden.

Resolver*65536 32Bit Resolver mit 65536 Inkrementen pro Motorumdrehung als 32 Bit breiter Wert

Encoder 32Bit Encoder als 12 Bit breiter Wert

SinCos*1024 32Bit Anzahl SinCos Perioden pro Motor-Umdrehung * 1024 als 32 Bit breiter Wert

Stepper without FB Gerechnetes Feedback für feedbacklosen Betrieb von Schrittmotoren. Es werden 4096 Inkremente pro Motor-Umdrehung zurückgegeben.

Iq Strom-Istwert

Bei den analogen Messsystemen wird jeweils die Auflösung des Messsystems mit in der Auswahl angegeben (Resolver*65536 32Bit):

- Resolver: Die Auflösung beträgt 16Bit, es werden pro Resolver-Umdrehung 65'536 Inkremente erzeugt.
- SinCos: Die Auflösung beträgt 10Bit; Anzahl SinCos Perioden pro Umdrehung * 1024 ergibt die Anzahl Inkremente pro Umdrehung, die übertragen werden.

Folgende Parameter müssen mit der Konfiguration im IMD übereinstimmen:

Link Sampling Rate	LinkSamplingRate
Inkremente pro Motor-Umdrehung	IncsPerTurn
Nennndrehzahl	TrunsPerMin
Siehe dt2-Files	

3.14 Motor

3.14.1 Motor Konfiguration

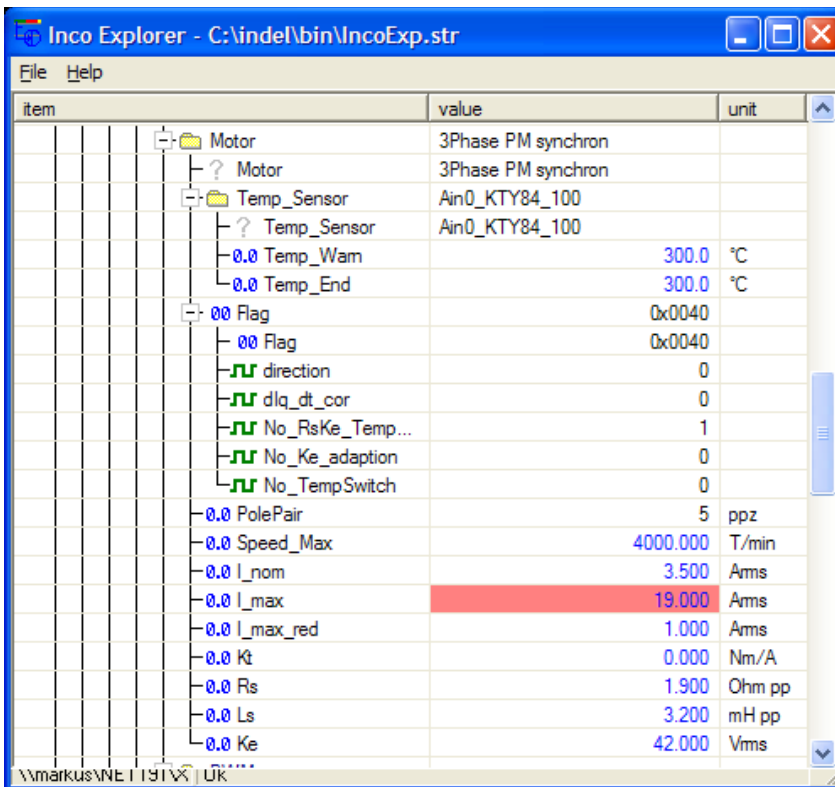


Abb 39: Motor

Werte aus Motor-Datenblatt einsetzen. Evtl. müssen die Werte durch Messung verifiziert werden. Bitte Normierung der Werte beachten: Ls, Rs werden entweder als Phase-Phase oder als Strang-Widerstand bzw. Induktivität angegeben. Das Ke wird ebenfalls häufig unterschiedlich angegeben.

Falls unklar ist, ob die Werte Rs, Ls, Ke korrekt sind, können diese gemäss Kapitel 10.21 ausgemessen werden.

Motor-Flags

Flag	description
direction	Drehrichtung des Motorfeldes
dlq_dt_cor	Vorhaltewert für Strom berechnet aus Induktivität des Motors Benötigter Strom für kommenden PWM Puls wird bereits mitberechnet. Bringt Phasen-Reserve ab ca. 200Hz, kann aber zusätzlichen Lärm verursachen.
No_Ke_Adaption	Keine automatische Korrektur von Ke während Fahrt mit hoher Drehzahl. Bit Standardmässig = 1
No_RsKe_TempComp	Keine Temperatur-Kompensation von R und Ke Diese drei Flags werden gesetzt, wenn kein Temperatur-Fühler verwendet wird, der einen Messwert in °C anzeigt. Bit Standardmässig = 1

Physikalische Motor-Parameter

Motor	3 Phase AC asynchron Motor 3 Phase PM synchron Motor, rotativ oder linear (Permanent-Magnet Motor) 2 Phase Stepper Motor: Schrottmotor mit oder ohne Feedback DC-Motor mit Feedback DC-Motor ohne Feedback
I_{MAX} [A_{RMS}]	Maximal Strom des Motors, dieser Strom wird vom Servo-Drive begrenzt.
I_{NOM} [A_{RMS}]	Nennstrom des Motors
I_{RED} [A_{RMS}]	Stromwert für die Betriebsart <i>Stromreduzierung</i> . In diesem Mode begrenzt der Regler den maximalen Strom auf I_{RED} . Die Betriebsart wird vom Feldbusmaster vorgegeben.
K_e [V]	Spannungskonstante des Motors (Gegen-EMK) [$V_{rms}/1000Umin$]
L_s [mH]	Induktivität der Motorwicklung (Phase-Phase)
PolePair	Pol-Paar Zahl des Motors. (Pol-Paar Zahl = Pol Zahl / 2)
R_s [Ω]	Ohmscher Widerstand der Motorwicklung (Phase-Phase)
Speed_Max [U/min]	Maximal zulässige mechanische Drehzahl des Motors

Falls Unklarheiten bestehen sollten R_s und L_s ausgemessen werden.

Achtung: Messfehler der eingesetzten Messinstrumente beachten!

Temperatur-Sensoren

Bei der Konfiguration ist darauf zu achten an welchem Geber-Stecker der Temp-Sensor angeschlossen ist: der Eingang Ain0 ist dem Resolver zugeordnet, der Eingang Ain1 dem SinCos Geber.

Die Kennlinie der KTY-Sensoren wird linearisiert:

KTY-84-100:

Widerstand bei 0°	475 Ohm
Widerstands-Verhältnis 100°/20°	0.190476
Widerstand bei 100°	1000 Ohm
Widerstand bei 20°	580 Ohm

KTY-84-110:

Widerstand bei 0°	778 Ohm
Widerstands-Verhältnis 100°/20°	0.1667
Widerstand bei 100°	1667 Ohm
Widerstand bei 25°	1000 Ohm

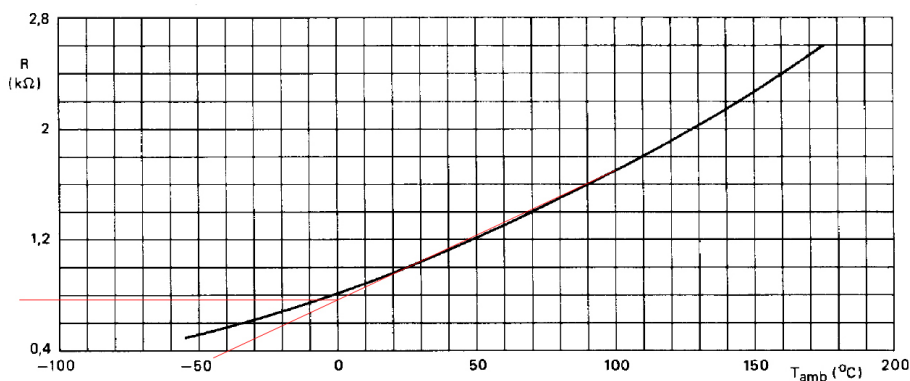


Abbildung 3.1: Linearisierung KTY-110

Temp_Sensor	Ain0_KTY84_100	KTY-100 an Analog Eingang 0 Resolver Stecker
	Ain0_KTY84_110	KTY-110 an Analog Eingang 0 Resolver Stecker
	Ain1_KTY84_100	KTY-100 an Analog Eingang 1 SinCos Stecker
	Ain1_KTY84_110	KTY-110 an Analog Eingang 1 SinCos Stecker
	Ain0_PTC	PTC an Analog Eingang 0 Resolver Stecker
	Ain1_PTC	PTC an Analog Eingang 0 Resolver Stecker
	Ain0_10kGT2	
	Ain1_10kGT2	

KTY-84-130 Sensoren können auch angeschlossen werden. Dazu einen KTY-84-100 Sensor konfigurieren und parallel zum Sensor einen 27 kOhm Widerstand verdrahten.

Temp_Warm Bei dieser Temperatur (100°C) gibt der Servo-Regler die Warnung Motor Temp warm aus.

Temp_End Bei dieser Temperatur (120°C) gibt der Servo-Regler die Fehlermeldung Motor Temp max aus.

3.14.2 Ke umrechnen für Linearmotoren

$$K_{e \text{ Rotativ}} = \frac{K_{e \text{ Translativ}} * \text{Magnetabstand} * 1000}{60}$$

$K_{e \text{ translativ}}$	V/m/s
$K_{e \text{ rotativ}}$	v/1000U/min
Magnetabstand	m

3.14.3 Ke umrechnen für Maxon Motoren

$$K_{e \text{ Rotativ}} = \frac{1000}{K_{e \text{ Maxon}}}$$

$K_{e \text{ Rotativ}}$	v/1000U/min
$K_{e \text{ Maxon}}$	min ⁻¹ V ⁻¹

3.15 PWM-Einstellungen


 PWM	12.000	kHz (--> Reset)
+000 PWM	12.000	kHz (--> Reset)
000 PWMfreq_multiplifier	x1	*
+000 DeadTime	909	ns
000 DeadTime_correction	none	

Abbildung 3.2: PWM-Einstellungen

PWM

Abtast- und PWM-Frequenz des Reglers: **8kHz, 12kHz, 16kHz, 32kHz**

Achtung: Eine hohe Frequenz bedeutet auch eine höhere Verlustleistung und somit eine höhere Wärmeentwicklung der Endstufe. Indel übernimmt keine Garantie bei defekten aufgrund von falschen Konfigurationen.

Nach dem Ändern der PWM-Frequenz ist ein Hardware-Reset des Drives nötig damit die Änderungen übernommen werden!

PWMfreq_multiplifier

x1, x2, x3, x4

Motoren mit kleiner Induktivität, die an MAX-Boards, mit geringen Abtastraten (8kHz) betreiben werden, kann der PWM zur Verbesserung des Stromverhaltens erhöht werden.

Durch die höhere PWM-Frequenz wird ein gleichmässigerer Stromfluss in den Windungen erreicht.

Höhere Schaltverluste erzeugen aber auch mehr Verlustleistung und Abwärme.

Es wird lediglich die PWM-Frequenz verändert, der Positions-Loop bleibt gleich.

x0.5

Die Einstellung x0.5 ist für grosse SAC3, SAC3x3 Drives (I_{NENN}: 24A) vorgesehen. Mit PWM x 0.5 regelt der Drive z.B. mit 16kHz, der PWM läuft aber nur mit 8kHz. Die Ausgabewerte werden in jeder Flanke geändert. Damit werden die Regler wesentlich weniger warm.

DeadTime

Totzeit der IGBTs. Dieser Parameter ist Regler abhängig.

DeadTime_correction

none

Bei Linearmotoren mit einer hohen Stromverstärkung (kPq, kPd) muss die Totzeit-Kompensation ausgeschaltet werden (none). Damit lassen sich störende Geräusche eliminieren.

U_half, U_full, U_double

Für Antriebe mit grossen Motoren, bzw. grossen Strömen kann das Regelverhalten im Stillstand verbessert werden, wenn die Totzeit-Kompensation eingeschaltet ist.

3.16 Positions Regler

Die Indel Servo-Regler sind mit einem modifizierbaren PID-Regler, bestehend aus drei verschiedenen Parametersätzen ausgestattet. Modifizierbarkeit meint Vorsteuerung oder sog. Booster. D.h. dem Sollwert können geschwindigkeits- und beschleunigungsabhängige Vorhaltewerte zuaddiert werden.

Die Vorhaltewerte sind reine Sollwerte und greifen nicht in den Regelalgorithmus ein, d.h. die Stabilität der Regelstrecke wird nicht tangiert.

Die drei verschiedenen PID-Parametersätze können gleichzeitig und völlig unabhängig voneinander verwendet werden. Standardmässig sind die drei Parametersätze für Vorwärts, Rückwärts und Stand-by vorgesehen.

Vorwärts- und Rückwärtsfahren

Damit kann z.B. auf Lastwechsel bei Handlingaufgaben gezielt eingegangen werden.

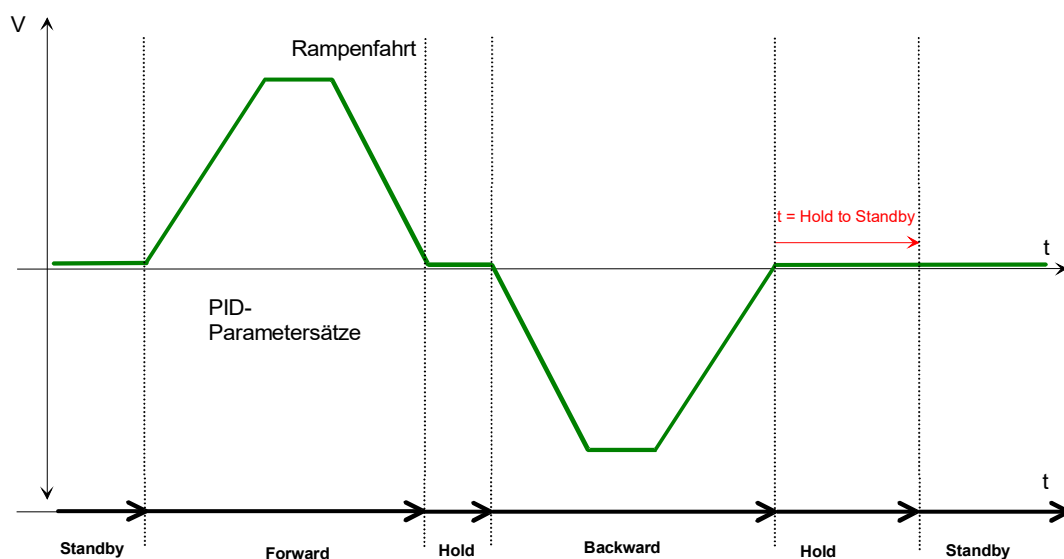


Abb 40: Trapezregler

Stand-by

Nach einer konfigurierbaren Zeit schaltet die Regelung auf Stand-by. Im Stand-by Mode kann der Motor z.B. in einem stromsparenden Mode betrieben werden.

PID-Parameter

Veränderte Motor-Parameter werden nur im RAM des Reglers aktualisiert. Um sie dauerhaft zu erhalten müssen sie mit `Burn Values to Target` ins Flash-Prom gebrannt werden. Werden die Parameter nicht gespeichert oder gebrannt sind sie nach dem Ausschalten für immer verloren!

PID-Werte können nur auch im aktiven Zustand des Reglers aktualisiert werden.

3.16.1 Konfiguration Positions-Regler

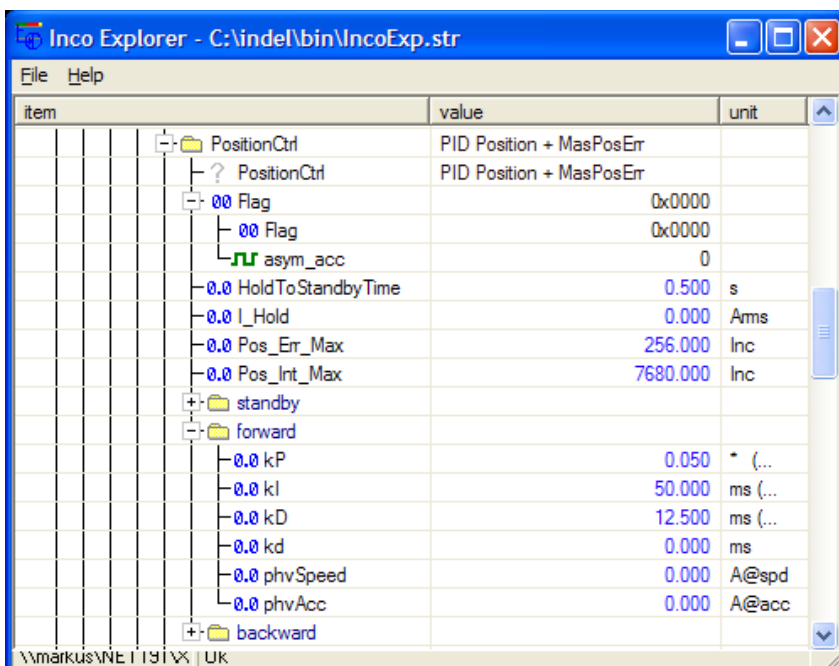


Abb 41: Positions Regler

PositionCtrl

Motoren mit Feedback

- PID Speed
- PID Position
- PID Position+MasPosErr

Geschwindigkeits Regler
 Positions Regler mit MasPos Error Korrektur
 Positions Regler mit MasPos Error Korrektur
 (MasPos Error wird immer in die Regelung mit einbezogen.)

Motoren ohne Feedback

- 3Phase Stepper withoutFB
- 2Phase Stepper withoutFB
- DC-Motor withoutFB

Positions Regler für 3-phasge PM Motoren ohne Feedback
 Positions Regler für 2-phasge PM Motoren ohne Feedback
 Positions Regler für DC-Motoren ohne Feedback

HoldToStand-by [s] Am Ende einer Rampenfahrt schaltet der Regler in den "Hold-Mode". Nach der einstellbaren Zeit "HoldToStandby" wechselt der Regler vom Hold-Mode in den Standby-Mode. Für den Standby-Mode existiert ein separater PID-Parametersatz. Diese Funktion erlaubt es, den Antrieb während seiner Stillstandzeit in einem Stromspar-Modus zu betreiben.

I_Hold Vorhaltewert: Für "hängende Lasten" kann mit "I_Hold" ein konstanter Wert, bzw. eine konstante Kraft vorgegeben werden.

Pos_Err_Max Begrenzung des maximalen Schleppfehlers. Typisch: 256 MotInc

Pos_Int_Max Begrenzung des I-Anteils der PID-Regelung (Positionsregler) Typisch: 7680 MotInc

MasPosErr_kl Geschwindigkeit für MasPosErr Korrektur. Für Achsen mit zwei Gebern (Resolver auf dem Motor, SinCos nach Getriebe und Spindel) wird lokal auf dem Resolver geregelt, der SinCos jedoch dem übergeordneten Regler zurück gegeben. Damit wird der wirkliche Positionsfehler als MasPosErr mit berücksichtigt. Wenn dazwischen das Getriebe etwas Spiel hat kann der Parameter zum Beispiel auf
 MasPosErr_kl = ca 5...10ms gesetzt werden. Dadurch erfolgt das Einziehen der Achse auf den Endwert zwar etwas langsamer, jedoch treten keine Schwingungen mehr auf.

Entspricht der Positionsregler auf dem Drive bereits dem MasPosErr Kanal so gilt:
 Für Hochauflösende Geber (SinCos): MasPosErr_kl = 0ms
 Für tiefer Auflösende Geber (Encoder / Resolver): MasPosErr_kl ca. 1ms

Flag Acc_Filter Beschleunigungs Filter

 asym_acc Asymetrischer Beschleunigungs-Vorhaltewert
 1: Beschleunigungs-Vorhaltewert in Forward gilt für Beschleunigung
 1: Beschleunigungs-Vorhaltewert in Backward gilt für Bremsen
 0: Beschleunigungs-Vorhaltewert in Forward gilt für pos. Richtung
 0: Beschleunigungs-Vorhaltewert in Backward gilt für neg. Richtung

3 PID Parameter-Sätze für Forward, Backward und Standby stehen zur Verfügung:

Wert		Normierung	Beschreibung
kP	*	A/Inc Regeldifferenz	Proportionalwert Positions-Fehler
kI	ms		Integralanteil Positions-Fehler Integrator
kD	ms		Differenzialanteil Geschwindigkeits Fehler
kd	ms		Beschleunigungs-Fehler
phvSpeed	*	A/v	Vorhaltewert Geschwindigkeit
phvAcc	ms		Vorhaltewert Beschleunigung

A: *Ampere*
 Inc: Inkrement
 v: Geschwindigkeit

Der Parameter **kd** kann nur mit hochauflösenden Gebersystemen verwendet werden.

Umschaltung der PID Parameter

Bei einem Schleppfehler von mehr als 10 MotInc wird der Parametersatz von standby auf forward bzw. Backward umgeschaltet.

$$10 \text{ MotInc} = \frac{1 \text{ Turn}}{N_{inc}} = \frac{360^\circ}{4096} = 0.879^\circ \quad \text{Siehe auch Kapitel 2.2.}$$

Achtung

Wenn für Standby andere PID-Werte als für Forward oder Backward verwendet werden, kann es sein, dass die Achse von Zeit zu Zeit einen kleinen „Ruck“ ausführt um einen aufsummierten Wegfehler zu kompensieren. Dies geschieht insbesondere, wenn das kP im Standby wesentlich kleiner ist als für Forward/Backward.

3.17 Power Supply

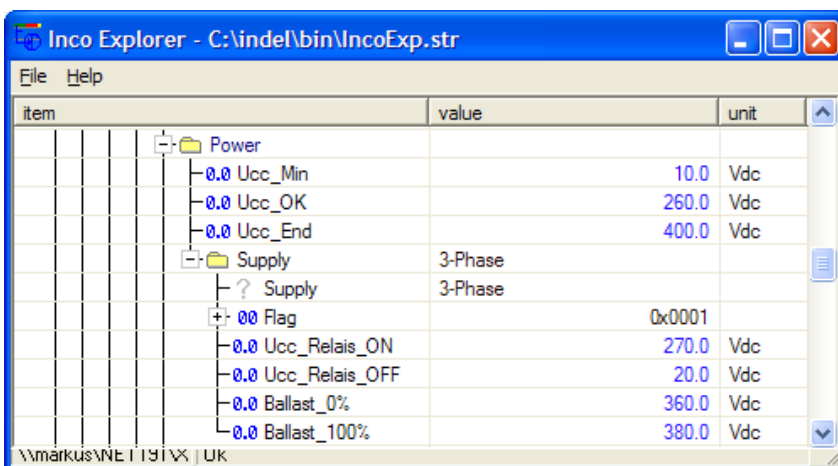


Abb 42: Power Supply

- Supply** 1-Phasige oder 3-Phasige Einspeisung, bei MAX- und AX-Boards *Brake only*
- Ucc_End** Übersteigt die Zwischenkreisspannung diesen Wert geht die Regelung auf Fehler.
- Ucc_Min** Unterhalb von diesem Grenzwert geht die Regelung auf Fehler. Zwischen Ucc_Ok
und Ucc_Min gibt die Regelung eine Warnung aus: "Ucc low"
- Ucc_OK** Zwischen Ucc_End und Ucc_OK liegt der normale Betrieb.
- Ballast_0%** Bei dieser Zwischenkreisspannung ist der PWM des Ballast-IGBT 0%
- Ballast_100%** Bei dieser Zwischenkreisspannung ist der PWM des Ballast-IGBT 100%
- Ucc_Relais_ON** Beim Einschalten des Reglers werden die Zwischenkreis-Kondensatoren über
einen Widerstand geladen. Sobald die Zwischenkreisspannung die Schwelle
überschreitet, werden die Ladewiderstände überbrückt.
- Ucc_Relais_OFF** Schwelle für Abschalten der Relais bei unterbrochener Versorgungsspannung
- Flags** No_PhaseFailure Die Phasenfehler Auswertung ist deaktiviert
Dieses Flag muss eins sein bei 1-Phasiger Einspeisung
und bei DC-Einspeisung über den Zwischenkreis.

Empfohlene Werte für die Konfiguration der Spannungs-Versorgung

Um die Zwischenkreis-Kondensatoren vor zu schneller Alterung zu schützen darf der Wert für Ucc Min nicht unterschritten werden.

Einspeisung		3-Phasig 400V	1-Phasig 230V	1-Phasig 120V	48V DC	24V DC
UCC_End	V	800	400	220	56	30
UCC_Min	V	450	260	125	40	20
UCC_Ok	V	500	280	135	44	22
Ballast_0%	V	760	360	180	50	25
Ballast_100%	V	780	380	200	52	28
UCC_Relais_ON	V	480	270	110	(42)	(22)
UCC_Relais_OFF	V	470	260	100	(40)	(20)

Die Werte für Ucc_Relais_ON/OFF bei MAX- und AX-Board sind nicht relevant, da auf diesen Boards keine Ladeschaltung vorhanden ist.

3.18 Speed Filter

Inco-Pfad zu den Speed-Filtern: `Ctrl.MotorConfig.SpeedFilter`

Siehe auch Kapitel: 11.4 Vorgehen beim Optimieren der Regelstrecke.

3.18.1 Average Speed-Filter

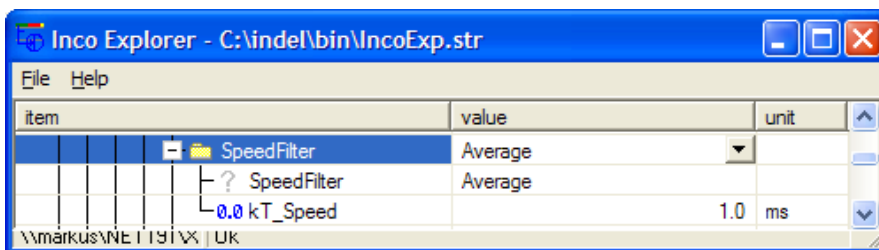


Abb 43: Average Filter

3.18.2 Speed Observer

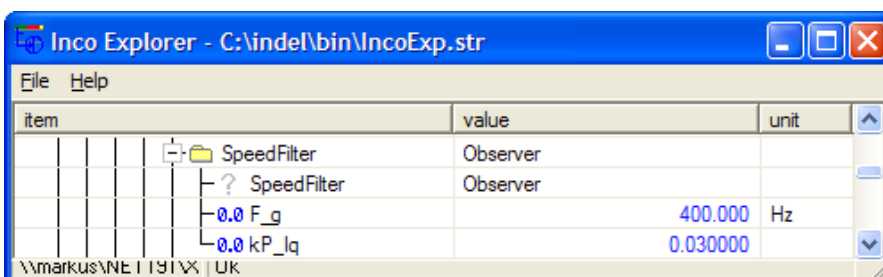


Abb 44: Observer

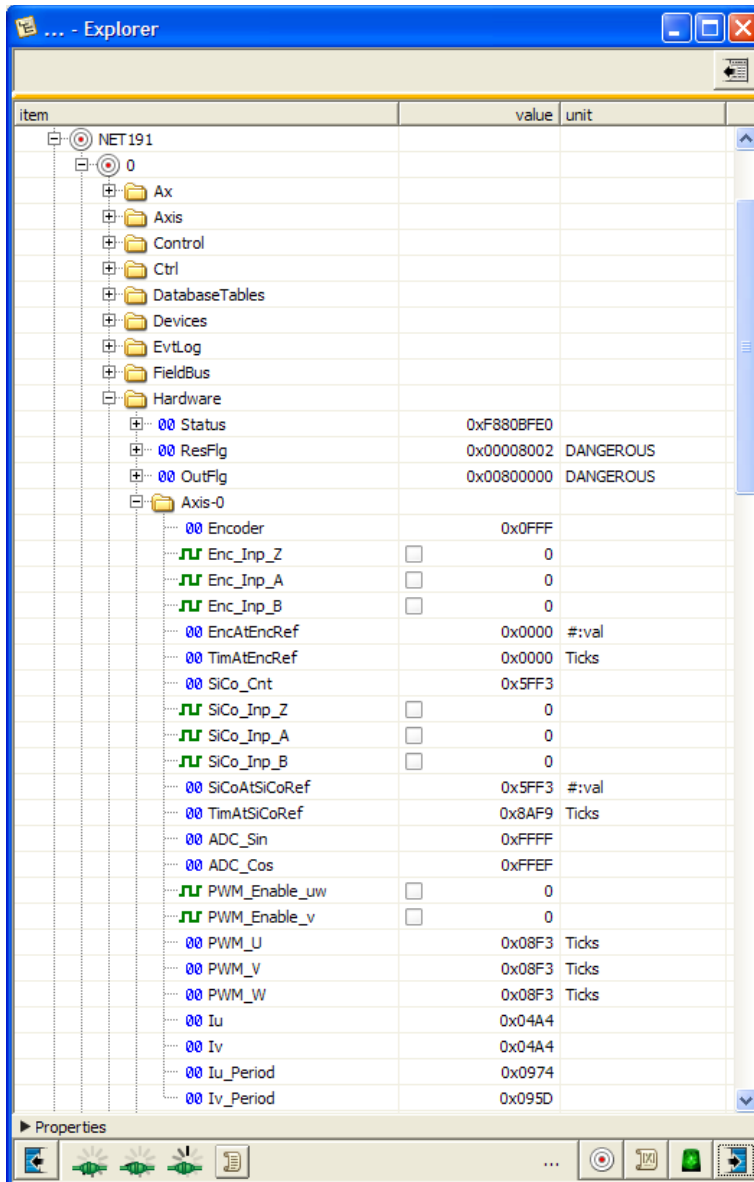
3.19 Hardware-Istwerte

Einige Ist-Werte wie z.B. die A-B-Spuren der Inkrementalgeber können direkt in der Hardware nachgeschaut werden.

Inco-Pfad zu den Hardware-Istwerten: Hardware

Bei älteren Firmware-Versionen:

Inco-Pfad zu den Hardware-Istwerten: `Ctrl.Hardware.Status`



item	value	unit
NET191		
0		
Ax		
Axis		
Control		
Ctrl		
DatabaseTables		
Devices		
EvtLog		
FieldBus		
Hardware		
00 Status	0xF880BFEO	
00 ResFlg	0x00008002	DANGEROUS
00 OutFlg	0x00800000	DANGEROUS
Axis-0		
00 Encoder	0x0FFF	
Enc_Inp_Z	<input type="checkbox"/>	0
Enc_Inp_A	<input type="checkbox"/>	0
Enc_Inp_B	<input type="checkbox"/>	0
EncAtEncRef	0x0000	#:val
TimAtEncRef	0x0000	Ticks
SiCo_Cnt	0x5FF3	
SiCo_Inp_Z	<input type="checkbox"/>	0
SiCo_Inp_A	<input type="checkbox"/>	0
SiCo_Inp_B	<input type="checkbox"/>	0
SiCoAtSiCoRef	0x5FF3	#:val
TimAtSiCoRef	0x8AF9	Ticks
ADC_Sin	0xFFFF	
ADC_Cos	0xFFEF	
PWM_Enable_uw	<input type="checkbox"/>	0
PWM_Enable_v	<input type="checkbox"/>	0
PWM_U	0x08F3	Ticks
PWM_V	0x08F3	Ticks
PWM_W	0x08F3	Ticks
Iu	0x04A4	
Iv	0x04A4	
Iu_Period	0x0974	
Iv_Period	0x095D	

Abb 45: Hardware Istwerte

3.20 Info-Link Motor-Konfig Files auf GinLink portieren

Um ältere Motor-Config Files auf den neusten Stand zu portieren können die fehlenden Parameter in ein separates File gespeichert werden. Dieses partielle Config-File kann dann in den Drive geladen werden.

Es empfiehlt sich in den partiellen Config-Files keine Achsen spezifischen Parameter wie GinLink Konfiguration, usw. zu speichern.

Beispiel für 3x400V Drive

```
Ctrl.MotorConfig.Version                3.199997
Ctrl.MotorConfig.PWM.DeadTime            909.088135
Ctrl.MotorConfig.PWM.DeadTime_correction 0.000000
Ctrl.MotorConfig.PWM.PWMfreq_multiplier  1.000000
Ctrl.MotorConfig.Power.Ucc_End           800.000000
Ctrl.MotorConfig.Power.Ucc_Min           0.000000
Ctrl.MotorConfig.Power.Ucc_OK            490.000000
Ctrl.MotorConfig.Power.Supply.Ballast_0% 700.000000
Ctrl.MotorConfig.Power.Supply.Ballast_100% 760.000000
Ctrl.MotorConfig.Power.Supply.Supply      2.000000
Ctrl.MotorConfig.Power.Supply.Ucc_Relais_OFF 470.000000
Ctrl.MotorConfig.Power.Supply.Ucc_Relais_ON 480.000000
Ctrl.MotorConfig.Power.Supply.Flag.Flag   0.000000
Ctrl.MotorConfig.Power.Supply.Flag.No_PhaseFailure 0.000000
```

4 Safety Konfiguration

4.1 Betrieb mit Safe Torque OFF (STO)

Voraussetzungen

- Die beiden Safety-Relais müssen von digitalen 24V Ausgängen aus der Steuerung angesteuert werden. (Regler-Freigabe Testen)
- Der Hauptschütz muss von einem 24V Ausgang aus der Steuerung angesteuert werden.

Konfiguration Extern Enable im Servo-Drive

Siehe Kapitel: 3.10 Extern Enable

Schutz des Drives vor Überlast

Wichtig: zum Schutz des Drives siehe auch Kapitel:
3.9.3 I2t Regelung,
10.1 Motor vor Überlast schützen

Konfiguration Ballast-Widerstand im Servo-Drive

Ballast-Widerstand konfigurieren gemäss Kapitel 3.17 Power Supply

Konfiguration im IMD-Konfiguration

Folgende Parameter in der dt2-Konfiguration müssen richtig gesetzt werden:

X.Axis.dt2

Emergency Type

Wenn der Extern Enable im Drive abgeschaltet wird, wird die Not-Stop Bremsrampe aktiviert:
 0: einfacher Stop ohne Not-Stop Bremsrampe
 1: Stop mit Not-Stop Bremsrampe, Regler wird deaktiviert
 2: Regler wird deaktiviert

Emergency Delay

Delay zwischen Stop und Inactivate bei Not-Stop Bremsrampe in ms

X.PosCtrl.dt2

emgB

Verzögerung der Not-Stop Bremsrampe; diese Verzögerung muss auf die mechanischen Gegebenheiten angepasst werden.

Überwachung der Hilfskontakte

Die Hilfskontakte der beiden Sicherheitsrelais (Schliesser) werden im INCO-Tree angezeigt. Der Teil der Applikation, der die Sicherheitsrelais einschaltet, muss den Zustand beider Hilfskontakte überprüfen. Der Vergleich der Zustände darf nicht länger als 50ms dauern.

Im Fehlerfall muss ein Not-Stop ausgelöst werden.

Regler-Freigabe Testen

Die Regler Freigabe muss zyklisch getestet werden. Dazu werden beide Sicherheitsrelais sowie der Extern-Enable ausgeschaltet und es wird versucht die Regelung zu aktivieren und die Achse zu verfahren.

Dieses Verfahren beinhaltet die folgenden Teilschritte:

Extern Enable	Relais 1	Relais 2
off	off	off
off	on	on
on	on	off
on	off	on
on	on	on

Ist die Testung der Regler-Freigabe nicht erfolgreich, darf der Drive nicht in Betrieb gesetzt werden!

4.2 Safe Torque Off anfordern

Verdrahtungs-Beispiele

Einige Verdrahtungs-Beispiele finden Sie in diesem Manual:
Indel-Safety-Manual.pdf (siehe Kapitel Anschlussbeispiele).

Konfiguration, Verdrahtung

Beispiel für eine Achse, die von der höchsten Geschwindigkeit in 200ms auf 0 bremsen kann:

- Schutztüre auf SAC Extern Enable verdrahten
- Schutztüre über sicheres Zeit-Relais (z.B. 300ms) zwei-kanalig auf SAC-STO verdrahten
- `MotorConfig.Enable.kT_ExtDis` auf z.B. 250ms programmieren
- im IMD-Projekt den SAC Extern Enable als Emergency-Eingang konfigurieren
- im IMD-Projekt die Emergency-Bremsrampe so wählen, dass von höchsten Geschwindigkeit in z.B. 200ms gebremst werden kann

Ablauf im SAC und Feldbus-Master

- Schutztüre geht auf, Extern Enable am SAC-Drive fällt ab
- Im SAC startet der Disable-Timmer, die Achse bleibt Aktiv im Positions-Betrieb und regelt weiter
- Im Master (SAM oder PCI-Karte) wird automatisch die Emergency-Bremsrampe eingeleitet
- je nach Geschwindigkeit wird nach spätestens 200ms der Speed=0 erreicht, der Master deaktiviert die Achse (rote Aktive LED löscht, falls der Aktive-Status auf einen digitalen Ausgang konfiguriert ist, fällt dieser Ausgang ab.)
- nach 250ms schaltet der SAC die Achse aus, egal ob Speed=0 schon erreicht wurde
- nach 300ms schaltet das Sicherheits-Zeit-Relais über den STO Safety-Torque-Off die Hardware-Endstufen aus.
Ab diesem Zeitpunkt ist die sichere Impulssperre aktiv und der Drive kann den Motor nicht mehr kontrollieren.

Überprüfen der Bremsrampe

Der ganzen Ablauf und die gewählte Emergency-Bremsrampe muss getestet und mit Log-Files überprüft werden:

- alle Signale müssen wie vorgesehen erscheinen:
`Ctrl.Actual.Enable.Ext_En`
`Ctrl.Actual.Enable.Safety_0`
`Ctrl.Actual.Enable.Safety_1`
`Ctrl.Actual.Enable.GinLink_En`
`Ctrl.Actual.GinLink.FB_Status.Axis_Active`
`Ctrl.Actual.PositionCtrl.cmd_V`
`Ctrl.Actual.PositionCtrl.act_V`
`Ctrl.Actual.PositionCtrl.cmd_A`
`Ctrl.Actual.PositionCtrl.act_A`
- der Drive darf nicht wegen Überstrom frühzeitig ausschalten und austrudeln
- den STO bei höchster Geschwindigkeit anfordern um sicherzustellen, dass die Geschwindigkeit in der vorgegebenen Zeit auf Null reduziert werden kann.

Hinweise

Der Extern Enable muss nicht noch mal auf einen digitalen Eingang geführt werden, dieser ist im Prozessabbild unter dem entsprechenden Positions-Kanal ersichtlich.

Bei Bedarf können die Extern Enable auch über digitale Ausgänge der Steuerung geführt werden. Dies ist nicht zwingend nötig.

Bei allen Indel-Drives und Motion-Boards ist immer nur 1 Extern Enable und 1 STO Eingang verfügbar und wirkt immer auf alle Achsen gleichzeitig.

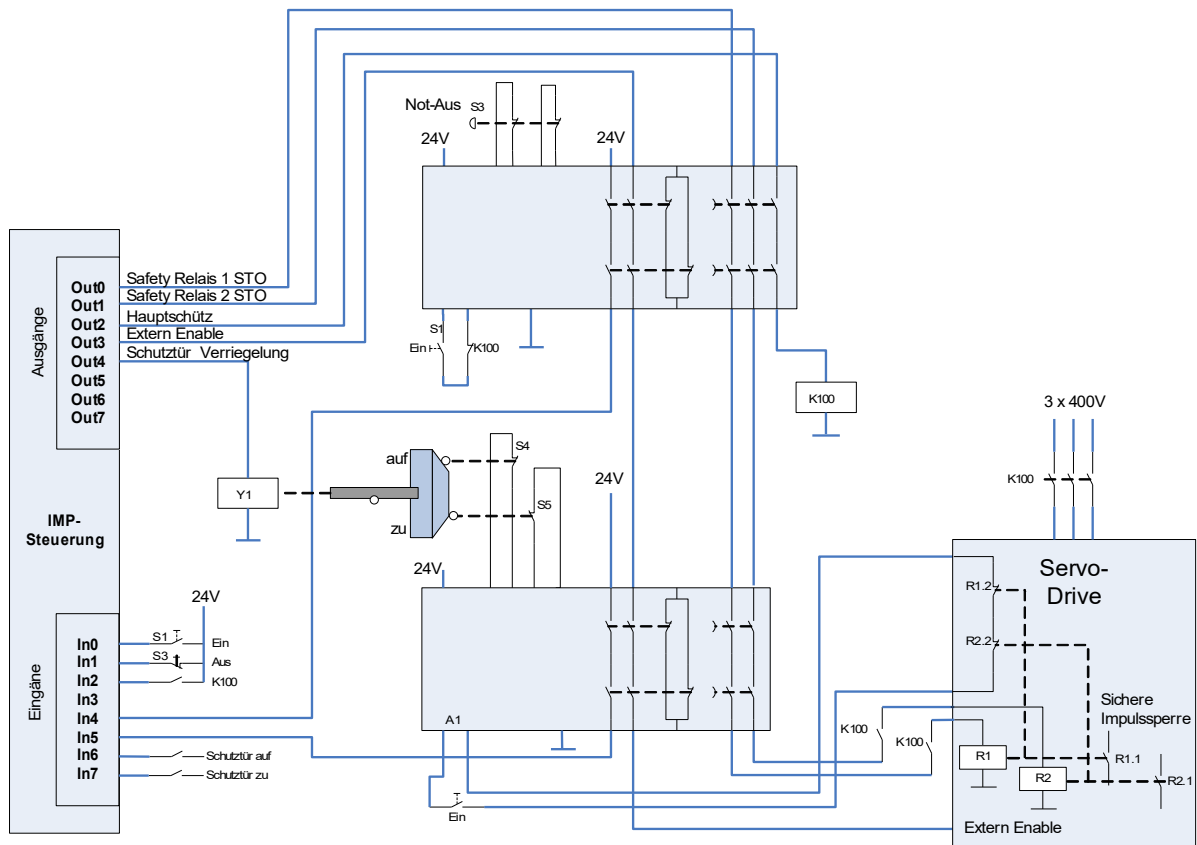


Abb 46: Beispiel: STO, verriegelte Schutztüre

Die Beispiele sind unverbindlich. Die verbindliche Auslegung der Sicherheitsfunktionen (SF) liegt in der Verantwortung des Anwenders. Er hat dabei den Stand der Technik in den entsprechenden europäischen Normen wie z.B. EN ISO 13849-1/-2, EN 62061, EN 1088, etc. zu folgen.

5 IMD-Konfiguration

5.1 GinLink Konfiguration in IMD

Beschreibung der GinLink Konfiguration im IMD mit dem GinLink.dt2 File.
 Die konfigurierte Zykluszeit auf dem GinLink jeder Achse muss mit der konfigurierten Zykluszeit im MotorenConfig File unter Ctrl.MotorConfig.GinLink.LinkSamplingRate übereinstimmen.

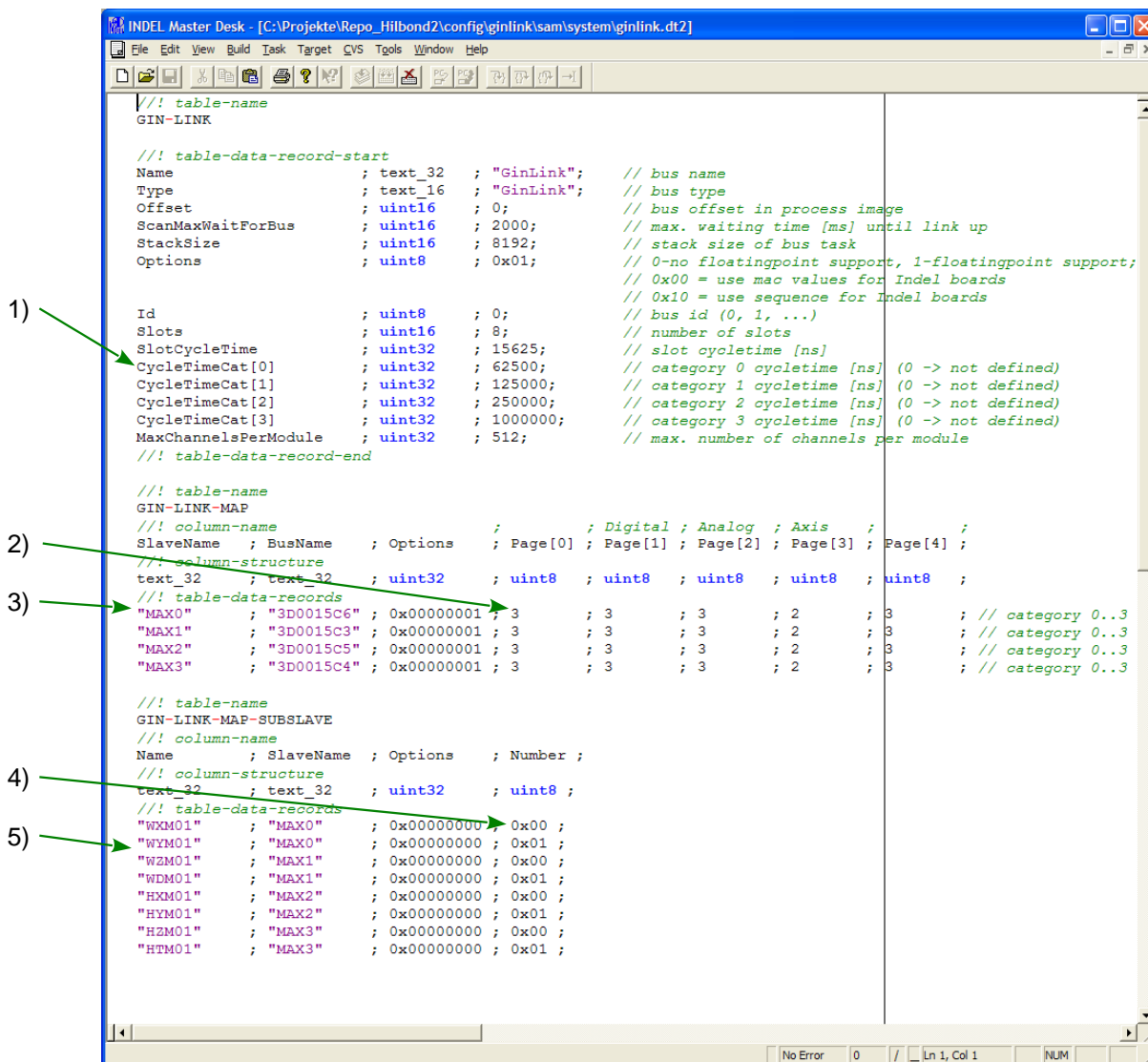


Abb 47: GinLink Konfiguration

Index aller GinLink Zykluszeiten 1)

Standardmässig sind folgende Zykluszeiten über den GinLink möglich:

Zykluszeit	Taktrate	Index
• 1ms	1kHz	3
• 0.25ms	4kHz	2
• 0.125	8kHz	1
• 0.0625	16kHz	0

Zykluszeiten der verschiedenen GinLink Pages 2)

Die gesamte Peripherie, inkl. Kommunikation ist auf verschiedene Daten-Pages verteilt. Jede dieser Pages kann mit einer konfigurierbaren Zykluszeit über den GinLink übertragen werden. Folgende Pages sind vorhanden:

- Page[0]
- Page[1] Digitale Ein- Ausgänge
- Page[2] Analoge Ein- Ausgänge
- Page[3] Achsen
- Page[4]

Zuordnung Servo-Drive zu MAC-Nummer 3)

Jeder GinLink Teilnehmer besitzt eine universelle MAC-Nummer. In der Konfiguration muss die Zuordnung zwischen dem Namen des GinLink Teilnehmers und dessen MAC-Nummer angegeben werden.

Bei Austausch eines GinLink Teilnehmers im Feld, muss hier die MAC-Nummer angepasst werden!

Index der Achsen bei Drives mit mehreren Endstufen 4) 5)

Drives mit mehreren Endstufen wie AX4, MAX2/4 oder SAC3x3 benötigen diesen Index für die Zuordnung der Namen der Achsen.

5.2 Achsen Konfiguration in IMD

5.2.1 Motor.dt2

Dieses File beinhaltet Informationen über den Motor bzw. Der Achse und werden vom SAM für die Umrechnung von Motoren-Inkrementen auf die Position benötigt.

TurnsPerMin	Normierungsfaktor für den Speed auf dem GinLink. Muss mit <code>Ctrl.Motorconfig.GinLink.Vcmd_100%</code> übereinstimmen
IncsPerTurn	Anzahl Inkremente auf dem in FB_Ch_0 Kanal konfigurierten Feedback. Muss mit der Achsen spezifischen Anzahl Inkremente Pro Motorenumdrehung aus dem MotorenConfig File übereinstimmen.
FeedPerTurn	Anzahl konfigurierter "Einheiten" pro Motorenumdrehung
GearRatio	Eingesetztes Getriebe. Wenn kein Getriebe vorhanden, dann Übersetzung 1.0

5.2.2 PosCtrl.dt2

Im PosCtrl.dt2 File muss die Variable DeadTime richtig gesetzt werden. Diese Totzeit [ms] wird vom Sam benötigt und muss auf die aktuelle Abtastfrequenz der entsprechenden Achse eingestellt werden.

Beispiel:

Regelfrequenz auf 16kHz:

$$DeatTime = 2 \frac{1}{Regelfrequenz} = 2 \frac{1}{16 \text{ kHz}} = 0.125 \text{ ms}$$

6 Regler Konfiguration

Im Verzeichnis `Ctrl.CtrlConfig` können sämtliche Parameter, die mit dem Regler im Zusammenhang stehen, verändert werden.

- Abgleichdaten der Strommessung
- Messbereiche von Strom, Spannung, Temperatur
- Maximaler IGBT-Strom
- Abgleich Resolver

Sämtliche Abgleiche werden werksintern bei INDEL AG vorgenommen. Es dürfen niemals Werte ohne Rücksprache mit Indel abgeändert werden. Werden die Regler-Parameter überschrieben (Reglerparameter-File *.chf ins Flash-PROM brennen), gehen Abgleichdaten unwiderruflich verloren und der Regler muss zur Reparatur eingeschickt werden.

Um die Regler-Konfiguration in den Drive laden zu können, wird ein Eintrag in der Windows-Registry benötigt. Dieser Eintrag kann bei Indel angefordert werden.

Dies ist nötig um die Abtastfrequenz des Positions-Reglers im Drive anzupassen.

7 Fehlermeldung vom Servo-Drive

7.1 Fehlermeldungen

Stop	0x0000'0001
Ucc kleiner Ucc min	0x0000'0002
Ucc grösser Ucc max	0x0000'0004
I2t überschritten > 120%	0x0000'0008
Endstufe überhitzt (80°C)	0x0000'0010
Motor-Temp überschritten	0x0000'0020
Motor Kurzschluss	0x0000'0040
Resolver- SinCos-Fehler	0x0000'0080
Maximale Drehzahl überschritten	0x0000'0100
Safety Relais nicht eingeschaltet	0x0000'0200
Auto-Kommutierungs Fehler	0x0000'0400
Strom-Endanschlag erreicht	0x0000'0800
Phasen-Fehler	0x0000'1000
PWM Watchdog: Interrupt Overrun	0x0000'2000
missing External Enable	0x0000'4000
missing (Motor) configuration	0x0000'8000
Feldbus Watchdog	0x0001'0000

7.2 Warnungen

Ucc ist kleiner Ucc ok	0x0000'0001
Ucc ist angelegt und OK	0x0000'0002
Warnung Iq erreicht	0x0000'0004 1)
Warnung Endstufe heiss (75°C)	0x0000'0010
Warnung I2t überschritten	0x0000'0020
Motor-Temp überschritten	0x0000'0040
100% Modulation überschritten	0x0000'0080
Warnung Entlade-Zeit überschritten	0x0000'0100

1) Diese Warnung tritt auch auf, wenn das Safety-Relais nicht eingeschaltet ist und die Impulssperre aktiviert ist.

Weitere Details zu den Fehlermeldungen finden Sie im Handbuch Hardware-Manual-SAC3.pdf.

8 Indel Positions Regler

8.1 Move Kommandos

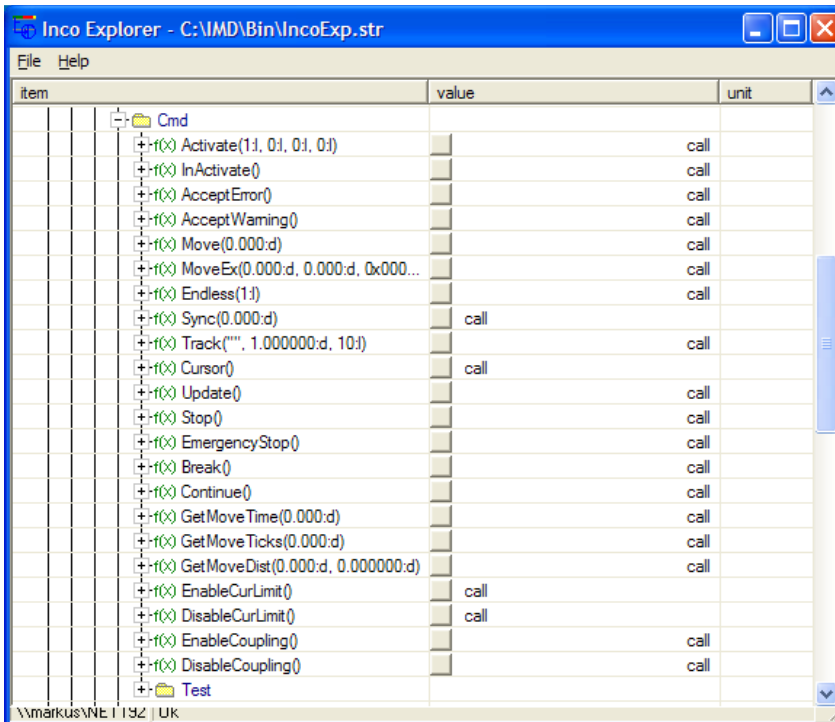


Abb 48: Move Kommandos

- Activate()** Aktivierung der Achse
 CheckPos: Der maximal zulässige Schleppfehler wird berücksichtigt, wird "maxSerr" überschritten geht die Regelung auf "Schleppfehler"
- SimulatePos: Der Ist-Wert, der vom Servo-Regler kommt, wird simuliert, es entsteht kein "errS".
- SimulateOut: Der Sollwert wird simuliert, der "errS" wird berechnet aber nicht ausgewertet.
- SimulateAct: Es wird kein Sollwert an den Servo-Regler gesandt. Die Rampe wird nur Softwaremässig gerechnet.
- Activate(0,0,1,0) entspricht "Simulations-Mode" im alten System

InActivate() Deaktivierung der Achse

Move() Positionen anfahren in °, m, mm

Endless() Endlos drehen in positive oder negative Richtung. Das Kurven-Profil wird unter "Ramp/Cmd" konfiguriert.

Break() Unterbricht den aktuellen Fahrbefehl

Continue() Führt den letzten Fahrbefehl weiter

Synch() Synch-Fahrt

Position: Innerhalb dieser Position wird die Synchmarke (Nullimpuls) erwartet. Wird der Nullimpuls nicht gefunden geht die Regelung auf Fehler:

SynchPosition: Synchronisationsposition, d.h. nach einer Synchronisation wird die IST-Position des Synch-Punktes auf "SynchPosition" gesetzt.

StopAtSynch: Entweder wird beim Überfahren der SynchPosition angehalten oder die Rampe wird zu Ende gefahren.

EmergencyStop()

NotStop
Stop: Die Achse stoppt mit der vorgegebenen maximalen Bremsrampe:
Axis.PhysicalAxis0.Ramp.Cmd.emgB = 360'000 °/s2

Inaktive: Die Achse wird inaktiv geschaltet

Stop/Inaktive: Die Achse stoppt mit der vorgegebenen maximalen Bremsrampe:
Axis.PhysicalAxis0.Ramp.Cmd.emgB = 360'000 °/s2
Nach der Delay Zeit wird die Achse inaktiv geschaltet

Delay: Zeit in ms

AcceptError() Quittierung von Fehlern
Diese Funktion ist für die Implementierung eines Fehlerhandlings vorgesehen. Einzelne oder sämtliche Fehler können mit dieser Funktion quittiert werden. Eine Achse kann nur aktiviert werden, wenn vorher alle Fehler quittiert worden sind.

AcceptWarning()
Quittierung von Warnungen

8.2 Fehlermeldungen vom Positions Regler im Feldbus Master

Schleppfehler Axis.PhysicalAxis0.Error = 0x0000'4000 "Control Error Pending"
Axis.PhysicalAxis0.Control.Error = 0x0000'0001
Axis.PhysicalAxis0.Ramp.Error = 0x0000'0000

Synchfehler Axis.PhysicalAxis0.Error = 0x0000'0002 "Synch Error"
Axis.PhysicalAxis0.Control.Error = 0x0000'0000
Axis.PhysicalAxis0.Ramp.Error = 0x0000'0000

Regler Fehler: Ucc < Uccmin

Axis.PhysicalAxis0.Error = 0x0000'4000 "Control Error Pending"
Axis.PhysicalAxis0.Control.Error = 0x0000'4000 "External Controller Error"
Axis.PhysicalAxis0.Ramp.Error = 0x0000'0000
\\..\PhysicalAxis0\Ctrl.Actual.Errors = 0x01010001 "Warn: Ucc < Ucc_OK"
"Warning: Power-OFF
DisCharge failure"
\\..\PhysicalAxis0\Ctrl.Actual.Errors = 0x0000'0003 "Ucc < Ucc_Min"

8.2.1 Achsen bewegen

Routine "Lösche Fehler, Start Achse"

- InActive() Lageregler und Servo-Regler inkativ schalten
- AcceptError() Fehler quittieren
- Active() Lageregler und Servo-Regler aktivieren

- Sleep(3000) Beim ersten Einschalten mit einem SinCos-Geber wird in dieser Zeit die Autokommütierung vorgenommen. Nach erfolgreicher Auto-Kommütierung wird der Regler aktiv (Siehe auch Konfiguration). Die Sleep-Zeit ist abhängig von der Konfiguration der Auto-Kommütierung!

- Fehlerhandling

Routine "Synchfahrt"

- Routine "Lösche Fehler, Start Achse"
- Steht die Achse auf der Nullmarke? Falls ja wegfahren
- Sync(Pos, SyncPos, StopAtSync)
- Fehlerhandling

Routine "Move"

- Move(Pos, Synchron/Asynchron)
- Fehlerhandling

9 Trapezregler

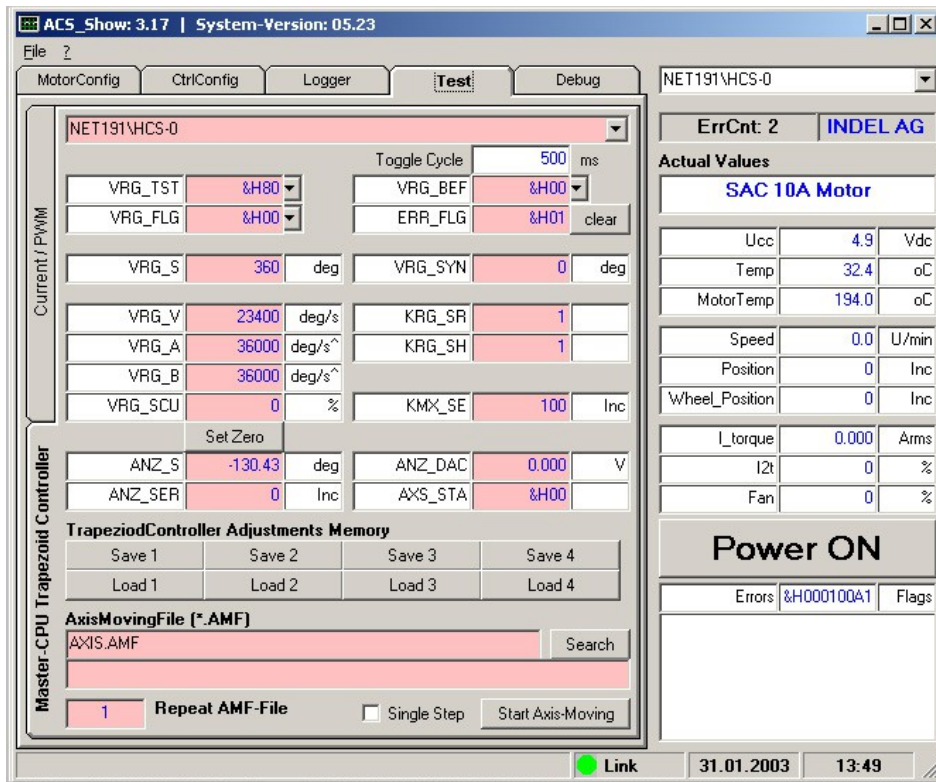


Abb 49: ACS-Show

9.1.1 ACS-Show

Mit dem ACS-Show können alle Indel Achs-Karten: Servo-Regler, Schrittmotor-Indexer und Endstufe, DC-Motor Endstufe, usw. in Betrieb gesetzt werden.

9.1.2 Vorgaben für Trapez- und S-Profil

Diese Werte können auch während des Betriebs noch geändert werden.

vrg_s	zu fahrende Strecke.	[Grad, m, mm]
vrg_v	zu fahrende Geschwindigkeit	[Grad, m, mm/sec]
vrg_a	Beschleunigung	[Grad, m, mm/sec ²]
vrg_b	Verzögerung	[Grad, m, mm/sec ²]
vrg_scu	S-Kurven Anteil in %	[%]
vrg_syn	Synchronisationsposition, d.h. nach einer Synchronisation wird die IST- Position des Synch-Punktes auf vrg_syn gesetzt.	

9.1.3 Regelkonstanten

Die Regelgewichte können für S (Weg) und V (Geschwindigkeit) sowie auch für RUN (Achse läuft) und HALT (Achse steht) separat angegeben werden. Der Wert 1.0 bedeutet, dass 1 INC Fehler 1 DAC-Bit Korrektur verursacht. Diese Werte können erst während der Inbetriebnahme endgültig festgelegt werden.

<code>krv_vr</code>	Regelgewicht für die Geschwindigkeit bei laufender Achse. Werden Indel Servo-Regler verwendet, muss dieser Wert auf Null gesetzt werden!
<code>krv_vh</code>	Regelgewicht für die Geschwindigkeit bei stehender Achse. Werden Indel Servo-Regler verwendet, muss dieser Wert auf Null gesetzt werden!
<code>krv_sr</code>	Regelgewicht für den Weg bei laufender Achse. Typisch <code>krv_sr</code> = 1 für die Servo-Regler.
<code>krv_sh</code>	Regelgewicht für den Weg bei stehender Achse. Typisch <code>krv_sh</code> = 1 oder 2 für die Servo-Regler.
<code>kmx_se</code>	Hierbei handelt es sich um den maximal erlaubten Wegfehler in INC (Differenz zwischen IST- und SOLL-Position). Wird dieser Wert während des Fahrens überschritten, so geht der Regler automatisch auf NOT-AUS.

9.1.4 Ist-Werte

Anzeige der aktuellen Position in der von Ihnen in der Konfiguration festgelegten Masseinheit (Grad, Meter oder Millimeter).

<code>S_ERR</code>	Aktuelle Positionsabweichung in INC (Differenz zwischen SOLL- und IST-Position). Wird diese Zahl während des Fahrens grösser als
<code>kmx_se</code>	so geht die Regelung auf NOT-AUS und Bit 0 im Error-Flag " <code>err_flag</code> " wird gesetzt.
<code>V_ERR</code>	(nur in Simulationsmode) Aktuelle Geschwindigkeitsabweichung in INC/Abstufung (Differenz zwischen Soll- und Ist-Geschwindigkeiten).
<code>DAC</code>	Aktueller DAC-Wert in Volt.

9.1.5 Achsen Status

Achsen Status

<code>axs_sta</code>	Bit 0: Syncheingang
----------------------	---------------------

9.1.6 Fahrbefehle, VRG_BEF

Mit diesem Aktionsschalter können verschiedene Fahrbefehle ausgelöst werden.

<code>Start</code>	<code>0x81h</code>	Fahre unter Berücksichtigung von <code>vrg_v</code> , <code>vrg_a</code> und <code>vrg_b</code> auf die Position <code>vrg_s</code> (mit F7 ist diese Funktion direkt anwählbar).
<code>Neues V/A</code>	<code>0x82h</code>	Beschleunige / Verzögere auf neue Geschwindigkeit <code>vrg_v</code>
<code>Synch</code>	<code>0x83h</code>	Fahre unter Berücksichtigung von <code>vrg_v</code> , <code>vrg_a</code> und <code>vrg_b</code> auf die Position <code>vrg_s</code> . Prüfe während des Fahrens das Auftreten eines Synchimpulses. Tritt ein solcher auf, so setze die IST-Position gleich <code>vrg_syn</code> und bremse normal ab, ansonsten setze das Fehler-Flag auf Synchfehler.
<code>Start Pull</code>	<code>0x84h</code>	Folge der Vorgabeposition <code>vrg_s</code> . Da die Regelung immer sofort (ohne eine

Rampe zu fahren) versucht **vrg.s** zu erreichen, dürfen die Schritte nicht zu gross gewählt werden. (Fahrfehler, wenn der Wegfehler zu gross wird!) Es sollte auch die max. Beschleunigung des Motors berücksichtigt werden. Dieser Modus kann z.B. zum Kurven fahren verwendet werden.

Stop	0x88h	Bremse die fahrende Achse unter Berücksichtigung von vrg_b auf Null ab (mit F8 ist diese Funktion direkt anwählbar).
Not Stop	0x89h	Bremse die fahrende Achse mit sofortiger Wirkung (ohne Bremsrampe) auf Null ab.
POS=0	0x8Ah	Normiere Winkel auf 0 ... 359°, gemäss Kapitel Normierung
POS=0	0x8Bh	setze IST-Position auf 0.
POS=VRG_SYN	0x8Ch	setze IST-Position auf VRG_SYN
Toggle		Fahre unter Berücksichtigung von vrg_v , vrg_a und vrg_b auf die Position vrg_s , warte 'Delay' ms, setze vrg_s = - vrg_s und beginne wieder von vorn. Wiederhole dies, bis abgebrochen wird.
PToggle		Fahre unter Berücksichtigung von vrg_v , vrg_a und vrg_b auf die Position vrg_s , warte 'Delay' ms und fahre erneut dieselbe Strecke. Wiederhole dies, bis abgebrochen wird (mit F9 ist diese Funktion direkt anwählbar).
Delay		Mit diesem Punkt kann die Wartezeit in ms zwischen zwei Fahrbefehlen bei 'Toggle' und 'PToggle' eingestellt werden.

9.1.7 Normierung, VRG_FLG

Der Master normiert die IST-Position immer automatisch, sobald HALT erreicht wird. Durch die Normierung gehen keine Inkremente verloren (keine Summenfehler, auch wenn nach jeder Fahrt auf 0.0 normiert wird).

Kein	0x00h	Die Position bleibt nach jeder Fahrt wie sie ist.
Winkel	0x01h	Der Winkel wird auf 0..359° normiert.
Nullen	0x02h	Die aktuelle Position wird nach jeder Fahrt auf 0.0 gesetzt.
Endlos	0x03h	Ab Start-Befehl wird bis zum nächsten Stop mit VRG_V gefahren, unabhängig davon wie gross vrg_s ist. (Hinweis: vrg_s muss für die Beschleunigungsrampe ausreichen, da sonst das Endlos-Fahren nicht aktiv wird.)
Runden	0x04h	Winkel auf 365° runden

9.1.8 Betriebsart, VRG_TST

Mit der Betriebsart entscheiden Sie, wie sich die Regelung gegenüber der Aussenwelt verhalten soll.

Inaktiv	0x00h	Die Regelung ist für diese Achse ausgeschaltet.
Aktiv	0x80h	Der volle Umfang der Regelung (inklusive Fehlerüberwachung usw.) ist aktiviert.
A. o. F.	0x81h	Die Regelung ist aktiv, geht jedoch nicht auf NOT-AUS bei Fahrfehler.
Simulation	0x82h	Der Motor wird simuliert. Es handelt sich hier um eine reine Sollwertausgabe.

DAC/INC **0x83h** Die Regelung für diese Achse ist ausgeschaltet. Man kann nun den DAC bzw. den Inkrementalgeberingang für eigene Zwecke brauchen.

9.1.9 Normfaktoren

knf_dac Dieser Faktor dient der internen Umrechnung von Sollgeschwindigkeit nach DAC-Bits. Er wird durch die Konfiguration berechnet.

knf_s Dieser Faktor dient der Umrechnung Ihrer Vorgaben V, S, A, B, nach Inc. Er wird ebenfalls durch die Konfiguration berechnet.

knf_e

knf_tim Abtastrate: 0 = 4 ms
 1 = 2 ms
 2 = 1 ms
 3 = 0.5 ms

9.1.10 Fehler Meldungen

Achsenfehler. Solange dieses Byte nicht gelöscht wird, werden keine neuen Befehle angenommen.

err_flg 00h ok

Bit 0=1 →Schlepp-Fehler. **anz_ser** wurde grösser als **kmx_se** -> AUS

Bit 1=1 →Es konnte nicht synchronisiert werden

Bit 2=1 →Die Achs-Karte meldet sich nicht korrekt

Bit 3=1 →Externer Regler Fehler

10 Inbetriebnahme Schritt für Schritt

Folgendes Kapitel zeigt Schritt für Schritt die Inbetriebnahme eines Motors bzw. einer Achse. Vor Beginn der Inbetriebnahme muss das gesamte Inbetriebnahme Manual und Hardware Manual durchgelesen werden.

Für die erste Inbetriebnahme sollte der Motor ohne Last betrieben werden. Der Bediener benötigt Sichtkontakt auf die Motorwelle.

Arbeiten Sie jeden Schritt exakt durch, werden einzelne Schritte ausgelassen, läuft der Motor unter Umständen nicht richtig oder zeigt ein Verhalten, das schwer erklärbar ist.

10.1 Motor vor Überlast schützen

Um den Motor während der Inbetriebnahme vor Überlast zu schützen sind folgende Punkte zu beachten:

- Aus Motor-Datenblatt: Maximale Zeit während der maximale Strom fließen darf
- Temperatur-Sensor von Motorwicklung anschliessen und in Betrieb nehmen
Wenn kein Temperatur-Sensor vorhanden ist, ist die I2t Regelung der einzige Schutz um den Motor vor Überlast zu schützen. Siehe Kapitel: 3.9.3 I2t Regelung
- I2t Regelung abgleichen
- Maximale Geschwindigkeit und Beschleunigungs-Rampe nicht übersteigen

10.2 Motorparameter eintragen

Werte aus Motor Datenblatt eingeben. Die Angaben beziehen sich auf PM-Synchronmotoren in Stern-Schaltung. Weitere Informationen sind in Kapitel 3.14 zu finden.

Motor Einspeisung ausschalten: L1, L2, L3 aus!

1. Inco-Pfad zur Motor Konfiguration `Ctrl.MotorConfig.Motor`
2. **Rs** Wird als R Phase-Phase eingegeben: $Rs = 2 * Rstr$
3. **Ls** Wird als L Phase-Phase eingegeben: $Ls = 2 * Lstr$
(Siemens-Angabe $Ld =$ Drehfeld-Induktivität -> L Phase-Phase)
4. **Ke** Wird als Ke Phase-Phase eingegeben -> da sind alle gleich richtig
in $V_{RMS}/1000U/min$

Falls **Unklarheiten** bestehen über die Korrektheit von Rs, Ls, Ke bzw. deren Einheiten nicht klar sind, sollten diese Werte gemäss Kapitel 10.21 Feinabgleich von Ke, Rs und Ls verifiziert werden.
5. **Inom** Wird in Arms angegeben
Imax Wird in Arms angegeben
Ired Wird in Arms angegeben $Ired = Inom$ (default)
6. **Speed_Max** Maximale mechanische Drehzahl
7. **BurnMotorCfg, File -> Save**

10.3 Temperatur-Schalter

1. Motor und Encoder (Resolver, Inkrementalgeber bzw. SinCos) an Regler anschliessen.
2. 24V für Regler-Speisung einschalten, Motoreinspeisung (3 x 400V) **nicht** einschalten!

10.3.1 Temperaturfühler im Resolver/SinCos Kabel

Befindet sich ein Temperaturfühler (Widerstand) in der Resolver/SinCos Kabel, muss die Motor Temperatur unter `Ctrl.Actual.Motor` einen sinnvollen Wert zwischen 20 ... 30° anzeigen.

Konfiguration Temperaturfühler

Inco-Pfad zur Motor Konfiguration: `Ctrl.MotorConfig.Motor`

3. **Temp_Warn** Ab dieser Motortemperatur wird eine Warnung angezeigt Temp_Warn = 100°C
4. **Temp_End** Ab dieser Motortemperatur wird ein Fehler angezeigt und der Regler wird inaktiv geschaltet. Temp_End = 120°C
5. **Flag**

<code>No_Ke_adaption</code>	= 0	Temperaturabhängige Ke Kompensation EIN
<code>No_RsKe_TempComp</code>	= 0	Temperaturabhängige Rs Kompensation EIN
<code>No_TempSwitch</code>	= 1	Kein Bi-Metall Switch in der Motor-Leitung

10.3.2 Temperatur-Endschalter im Resolver/SinCos Kabel

Befindet sich ein Endschalter für Motorübertemperatur in der Resolver- SinCos-Leitung, erscheint unter Actual Value für den Wert `Ctrl.Actual.Motor` die Temperatur -90° wenn der Schalter geschlossen ist und +190°C wenn der Schalter offen ist.

6. **Temp_Warn** Temperatur Warnung = 100°C
7. **Temp_End** Temperatur Error = 100°C
8. **Flag**

<code>No_Ke_adaption</code>	= 1	Temperaturabhängige Ke Kompensation AUS
<code>No_RsKe_TempComp</code>	= 1	Temperaturabhängige Rs Kompensation AUS
<code>No_TempSwitch</code>	= 1	Kein Bi-Metall Switch in der Motor-Leitung

10.3.3 Endschalter in den Motor-Leitungen

Befindet sich ein Endschalter für Motorübertemperatur in der Motorleitung erscheint unter Actual Value für den Wert `Ctrl.Actual.Motor` die Temperatur +190°C

9. **Temp_Warn** Temperatur Warnung = 300°C
10. **Temp_End** Temperatur Error = 300°C
11. **Flag**

<code>No_Ke_adaption</code>	= 1	Temperaturabhängige Ke Kompensation AUS
<code>No_RsKe_TempComp</code>	= 1	Temperaturabhängige Rs Kompensation AUS
<code>No_TempSwitch</code>	= 0	Bi-Metall Switch in der Motor-Leitung

10.3.4 Temperaturfühler in den Motor Leitungen

Temperatur-Sensoren, die in der Motor-Leitung verlegt sind, dürfen nicht an Signal-Stecker verdrahtet werden!

Isolationsklasse beachten! An den Sensor-Steckern gilt 50V als maximale Bemessungsspannung.

10.4 Feedback Konfigurieren

Informationen zu den einzelnen Feedbacksystemen sind in den Kapiteln 3.3, 3.4, 3.5 und 3.6 zu finden.

Es können mehrere Feedback für eine Achse verwendet werden. Der typische Anwendungsfall beinhaltet einen Geber der direkt auf der Motorwelle sitzt und einen zusätzlichen Geber z.B. von einem Massstab, der nach einem Getriebe oder Spindel platziert ist.

Bei Applikationen mit mehreren Messsystemen wird der Geber, der auf der Motorwelle sitzt für die Feldregelung verwendet.

Das zweite Messsystem wird als Feedback für den Positions-Regler im Drive verwendet. Beide Positions-Feedback können über den Feldbus an den Master gesendet werden.

Falls nur Ein Feedback auf der Motorwelle vorhanden ist, so wird dieses für die Feldregelung und für die Positionsregelung verwendet.

`Ctrl.MotorConfig.FB_MotorField` Feedback für Feldregelung

`Ctrl.MotorConfig.FB_PositionCtrl` Feedback für Positionsregelung

Beispiel 1

Ein Messsystem, Inkrementalgeber:

Feldregelung	<code>Ctrl.MotorConfig.FB_MotorField</code>	PM Encoder
Positions-Regelung	<code>Ctrl.MotorConfig.FB_PositionCtrl</code>	Encoder
GinLink Feedback	<code>Ctrl.MotorConfig.GinLink.FB_Ch_0</code>	Encoder 32Bit
GinLink Sollwert	<code>Ctrl.MotorConfig.GinLink.Cmd_Ch_0</code>	Speed

Beispiel 2

Resolver auf der Motorwelle, SinCos von Massstab, GinLink Feedback Kanal 0 (32 Bit breiter Positions-Wert)

Feldregelung	<code>Ctrl.MotorConfig.FB_MotorField</code>	PM Resolver
Positions-Regelung	<code>Ctrl.MotorConfig.FB_PositionCtrl</code>	Resolver
GinLink Feedback	<code>Ctrl.MotorConfig.GinLink.FB_Ch_0</code>	SinCos *1024 32Bit
GinLink Sollwert	<code>Ctrl.MotorConfig.GinLink.Cmd_Ch_0</code>	Speed

Beispiel 3

Resolver auf der Motorwelle, SinCos von Massstab, GinLink Feedback Kanal 1 (32 Bit breiter Positions-Wert)

Feldregelung	<code>Ctrl.MotorConfig.FB_MotorField</code>	PM Resolver
Positions-Regelung	<code>Ctrl.MotorConfig.FB_PositionCtrl</code>	SinCos
GinLink Feedback	<code>Ctrl.MotorConfig.GinLink.FB_Ch_0</code>	Resolver *65536 32Bit
GinLink Feedback	<code>Ctrl.MotorConfig.GinLink.FB_Ch_1</code>	SinCos *1024 32Bit
GinLink Sollwert	<code>Ctrl.MotorConfig.GinLink.Cmd_Ch_1</code>	Speed

Auflösung der analogen Messsysteme

Resolver werden mit 16Bit, 65'536 Werten aufgelöst, SinCos-Geber werden mit 10Bit, 1'024 Werten aufgelöst.

10.5 Feldbus-Kommunikation auf dem Regler konfigurieren

Siehe auch Kapitel 3.13 für weitere Informationen.

Inco-Pfad zur GinLink Konfiguration `Ctrl.MotorConfig.GinLink`

1. `Vcmd_100%` Maximal vorkommende Drehzahl der Achse. Dies ist ein Normierungsfaktor Muss mit dem Wert in der IMD Projekt-Konfiguration übereinstimmen!
2. `LinkSamplingRate` Abtastrate der übergeordneten Positionsregelung vom Feldbusmaster. Muss mit dem Wert in der IMD Projekt-Konfiguration übereinstimmen!
3. `Cmd_Ch_0` Standard: Speed 32Bit
4. `Cmd_Ch_1` Standard: MasPosErr 32Bit
5. `Cmd_Ch_2` Standard: +Iq 32Bit
6. `FB_Ch_0` Entsprechendes Feedback über den GinLink auswählen. Diese Auswahl bestimmt den Wert für `IncsPerTurn` in der Motor-Konfiguration in der IMP Projekt-Konfiguration
7. `FB_Ch_1` Standard: Zusätzliches Feedback, entsprechendes Feedback auswählen.
8. `FB_Ch_2` Standard: Wirkdstrom des Reglers Iq 32Bit.

10.5.1 Konfigurations Beispiel

SinCos Geber 512 Strichen
 Drehzahl 3000 U/min
 Abtastrate 2 ms
 FB-Kanal Kanal 0

Anzahl Inkremente pro Umdrehung: $512 \text{ Striche} * 1024$ = 524'288 Inc/T

Anzahl Inkremente pro Abtastrate: $\frac{524'288 \text{ Inc/s} * 3000 * 2\text{ms}}{60\text{s}}$ = 52428.8 Inc/2ms

Anzahl Bits um +- 52428.8 Inkremente zu übertragen: = 17 Bit (32 Bit max.)

`FB_MotorField` PM SinCos
`FB_PositionCtrl` SinCos

`FB_Ch_0` SinCos*1024 32 Bit
`Cmd_Ch_0` Speed

`IncPerTurn` 524'288 512 Striche * 1024 (in IMD Projekt-Konfiguration eingeben)

Eine minimale Auflösung von 4096 Inc pro Umdrehung sollte nicht unterschritten werden.

Beispiel 3

SinCos Geber	40u Teilung	Massstab nach Spindel
Resolver	16 Bit Auflösung	auf Motorwelle
Drehzahl	4500 U/min	
Spindel	5mm/U	
Abtastezeit	0.5 ms	
FB-Kanal	Kanal 0	SinCos
FB-Kanal	Kanal 1	Resolver

$$\text{Anzahl Inkremente pro Umdrehung: } \frac{5\text{mm} * 1024}{40\mu\text{m}} = 128'000 \text{ Inc/MotorTurn}$$

(von SinCos)

$$\text{Anzahl Inkremente pro Abtastezeit: } \frac{128'000 \text{ Inc/T} * 4500 * 0.5\text{ms}}{60} = 4'800 \text{ Inc/0.5ms}$$

$$\text{Anzahl Bits um } \pm 4800 \text{ Inkremente zu übertragen: } = 14 \text{ Bit (32 Bit max.)}$$

FB_MotorField	PM Resolver
FB_PositionCtrl	SinCos

FB_Ch_0	SinCos*1024 32 Bit
FB_Ch_1	Resolver*65536 32 Bit
Cmd_Ch_0	Speed

IncPerTurn	128'000	125 * 1024 (in IMD Projekt-Konfiguration eingeben)
------------	---------	--

10.6 Konfiguration der Feldbus-Kommunikation In der Software

Jede Achse benötigt auch eine entsprechende Konfiguration in der Software. Dies ist nötig damit der SAM die Daten, welche er vom Regler über den Feldbus erhält, richtig umrechnen und interpretieren kann. Auch bestimmte Normierungen sind für ein einwandfreies funktionieren der Achse auf beiden Seiten (SAM und Regler) nötig.

Weitere Informationen dazu in Kapitel 5.

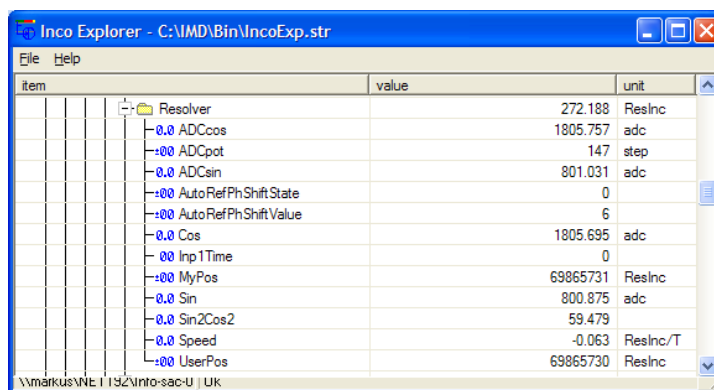
10.7 Messsystem in Betrieb nehmen

In der Konfiguration vom Messsystem (Inkrementalgeber, Resolver oder SinCos) muss die Anzahl Inkremente pro Umdrehung angegeben werden.

Konfiguration des Feedbacks siehe Kapitel 10.4.

10.7.1 Drehrichtung überprüfen

1. Motorwelle, bzw. Linear-Motor in positive Richtung bewegen, die Position muss aufwärts zählen.
2. Der Encoder-Wert muss aufwärts zählen.
 Inkrementalgeber Ctrl.Actual.Encoder
 Resolver Ctrl.Actual.Resolver
 SinCos Ctrl.Actual.SinCos



item	value	unit
Resolver	272.188	ResInc
0.0 ADCcos	1805.757	adc
-0.0 ADCpot	147	step
0.0 ADCsin	801.031	adc
-0.0 AutoRefPhShift State	0	
-0.0 AutoRefPhShift Value	6	
0.0 Cos	1805.695	adc
0.0 Inp1Time	0	
-0.0 MyPos	69865731	ResInc
0.0 Sin	800.875	adc
0.0 Sin2Cos2	59.479	
0.0 Speed	-0.063	ResInc/T
-0.0 UserPos	69865730	ResInc

Abb 50: Ist-Werte Resolver

Bei Resolver und SinCos Geber zusätzlich überprüfen, ob der Wert für Sin2Cos2 innerhalb der Limiten Sin2Cos2_Max. Sin2Cos2_Min liegt. Ist das Sin2Cos2 zu tief, die Verdrahtung und die Montage der Geber überprüfen.

3. Falls die Drehrichtung nicht stimmt, das Flag für die Drehrichtung invertieren:
 ... \\Ctrl.MotorConfig.Encoder.Flag.direction
 ... \\Ctrl.MotorConfig.Resolver.Flag.direction
 ... \\Ctrl.MotorConfig.SinCos.Flag.direction

Nachdem das Drehrichtungs-Flag verändert worden ist muss die Achse neu kommutieren! Dazu die Parameter ins Flash-Promm brennen und den Drive stromlos schalten. Oder die Kommutierung über den Test-Mode von Hand wiederholen.

10.7.2 Normdrehrichtung

Drehrichtung vorwärts = Von hinten auf den Motor schauen, der Motor muss im Uhrzeigersinn drehen.

10.7.3 Auflösung der Geber überprüfen

1. Die User-Pos des entsprechenden Gebers auf 0 setzen:
`Ctrl.Actual.Encoder.UserPos = 0`
`Ctrl.Actual.Resolver.UserPos = 0`
`Ctrl.Actual.SinCos.UserPos = 0`
2. Die Motorwelle möglichst genau um 360° in positive Richtung drehen, den Linearmotor möglichst genau um einen Magnet-Abstand in positive Richtung bewegen.
3. Die User-Pos muss jetzt den Wert (positiv) anzeigen, der konfiguriert worden ist. Die Drehung bzw. die Verschiebung der Achse sollte so genau durchgeführt werden, dass es möglich ist den Unterschied zwischen Gebern mit 1024 und 1000 Strichen zu erkennen.

10.8 Ist-Position im Feldbusmaster überprüfen

Nachdem die Konfiguration im IMD Projekt und im Servoregler eingetragen ist, kann diese überprüft werden:

Nachdem die IMD-Konfiguration angepasst wurde, neu Trasen!

1. Dazu im Feldbusmaster unter Axis die entsprechende Achse anwählen:

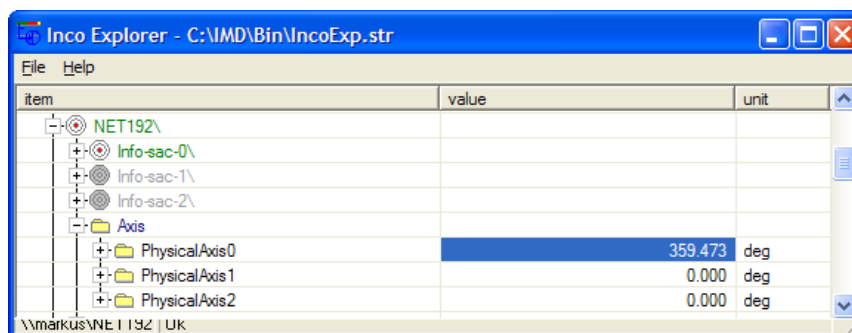


Abb 51: Position der Achse

2. Die Position von Hand auf Null setzen
3. An der Motor-Welle um genau 360° drehen
Den Linearmotor um genau einen Magnetabstand verschieben
4. Das Vorzeichen der Position sowie der Wert der Position muss der Bewegung entsprechen.

10.9 Externe Reglerfreigabe

1. Die Externe Reglerfreigabe ExtEn muss auf 24V verdrahtet sein.
2. Die beiden Safety 24V Eingänge müssen eingeschaltet sein. (Stecker X100)
3. Die grüne LED auf der Frontplatte des Reglers muss leuchten.
4. Die Konfiguration unter `Ctrl.MotorConfig.Enable` muss entsprechend den Anforderungen der Applikation eingestellt werden.

Weitere Informationen zur Konfiguration des Extern Enable in Kapitel 3.10

10.10 PWM

Unter `Ctrl.Motorconfig.PWM.PWM` kann neu die Abtastfrequenz der Endstufe in dem MotorenConfig File konfiguriert werden. Dies erlaubt dem Benutzer eine individuelle Anpassung an die Anforderungen ohne das eine Änderung im Controler File (siehe Kapitel 6) nötig ist. Bei einem Mehrfachregler (bspw. SAC3x3) ist diese Konfiguration nur auf der Achse 0 möglich und gilt dann auch für alle weiteren Achsen des Reglers.

Informationen zu den PWM Einstellungen sind in Kapitel 3.15 zu finden.

10.11 Power

Unter `Ctrl.MotorConfig.Power` wird das Power Netzteil des Reglers konfiguriert. Bei einem Mehrfachregler (bspw. SAC3x3) ist die Konfiguration des Netzteils nur auf der Achse 0 möglich und gilt dann für den gesamten Regler.

Informationen zu den Netzteil Einstellungen sind in Kapitel 3.17 zu finden.

10.12 Positionsregler

Unter `Ctrl.MotorConfig.PositionCtrl` wird der Typ der Regelung der Achse spezifiziert. Bei Achsen ohne Positionsfeedback, bspw. Open-Loop Schrittmotoren wird ein virtuelles Feedback von 4096 IncPerMotTurn generiert.

Weitere Infos zu Positionsregler sind in Kapitel 3.16 zu finden.

10.13 Polpaarzahl finden, verifizieren

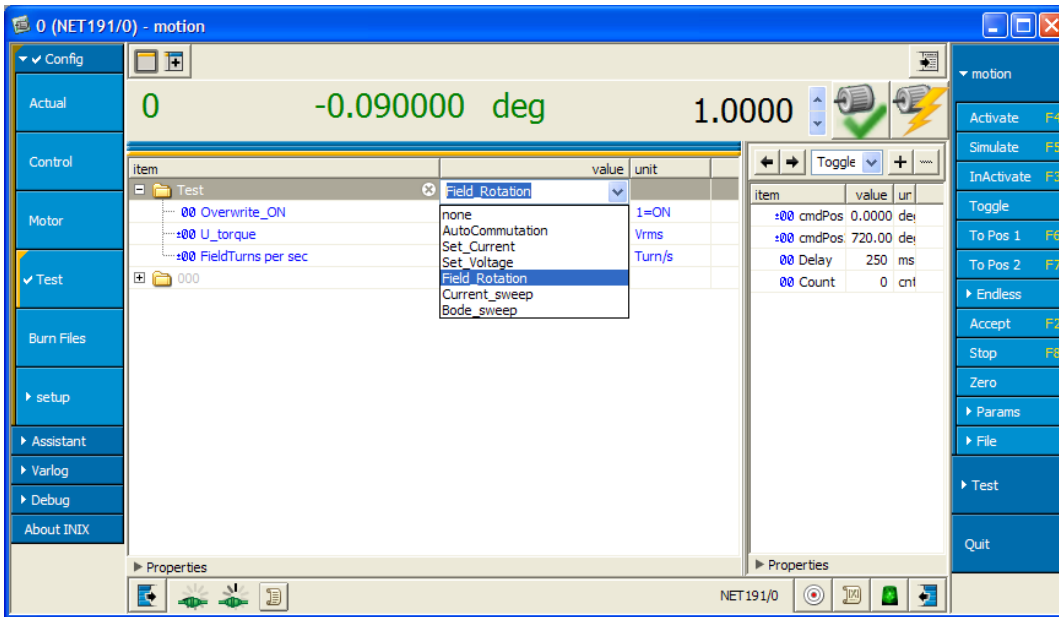


Abb 52: Feld-Mode

U_torque Motor-Spannung
 FieldTurns per sec Anzahl Feld-Umdrehungen pro Sekunde

1. Den Motor ohne Last betreiben! Unter Test den Field_Rotation Mode anwählen
2. Spannung U_torque wählen, mit 1V beginnen, Spannungs-Wert langsam erhöhen. Anzahl Feldumdrehungen FieldTurns_Per_sec wählen, Defaultwert = 1 Elektrisches Drehfeld dreht um 360° pro Sekunde
3. Regler in den Simulationsmode schalten. Dazu F3, F5 betätigen. Dreht der Motor bei 10V immer noch nicht, muss die Verdrahtung kontrolliert werden.

Pool Paar Zahl	U/s
1	1
2	1/2
3	1/3
4	1/4

Für eine genaue Bestimmung der Polpaarzahl ist es ratsam einen Log aufzuzeichnen.

10.14 Drehrichtung verifizieren (vor Kommutierung)

Dieser Test kann auch ohne Kommutierung durchgeführt werden.

1. Den Motor ohne Last betreiben! Unter `Test` den `Field_Rotation Mode` anwählen
2. Spannung `U_torque` wählen, mit 1V beginnen, Spannungs-Wert langsam erhöhen.
Anzahl Feldumdrehungen `FieldTurns_Per_sec` wählen, Defaultwert = 1
→ Elektrisches Drehfeld dreht um 360° pro
3. Regler in den Simulationsmode schalten. Dazu F3, F5 betätigen.
Der Motor muss nun vorwärts drehen!

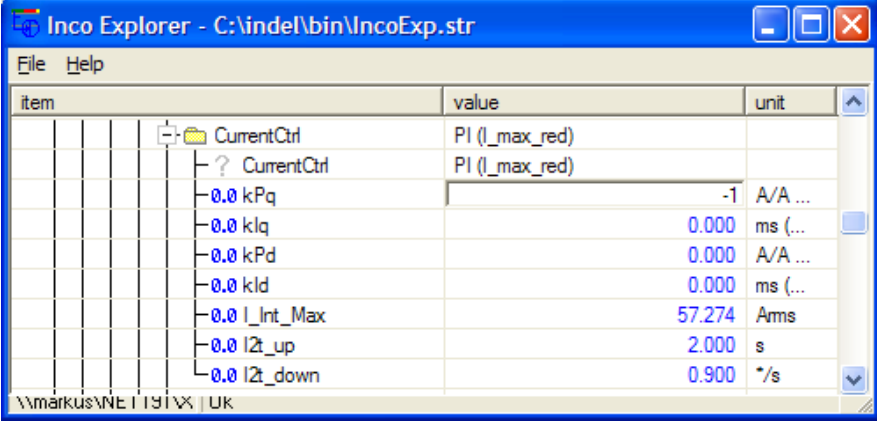
Falls die Drehrichtung nicht stimmt, das Flag für die Drehrichtung in der Motor-Konfiguration invertieren: `Ctrl.MotorConfig.Motor.Flag.direction`

Nachdem das Drehrichtungs-Flag verändert worden ist muss die Achse neu kommutieren! Dazu die Parameter ins Flash-Promm brennen und den Drive stromlos schalten. Oder die Kommutierung über den Test-Mode von Hand durchführen.

10.15 Stromregler abgleichen

Verschiedene Stromregler-Varianten und Parameter sind in Kapitel 3.9 beschrieben.

Die Parameter für den Stromregler können vom SAC-Drive automatisch berechnet werden.



item	value	unit
CurrentCtrl	PI (I_max_red)	
? CurrentCtrl	PI (I_max_red)	
-0.0 kPq	-1	A/A ...
-0.0 kIq	0.000	ms (...)
-0.0 kPd	0.000	A/A ...
-0.0 kId	0.000	ms (...)
-0.0 I_Int_Max	57.274	Ams
-0.0 I2t_up	2.000	s
-0.0 I2t_down	0.900	*/s

Abb 53: Stromregler berechnen

Dazu beim Parameter `Ctrl.MotorConfig.CurrentCtrl.kPq` (p-Anteil von Wirkstrom-Regler) `-1` eingeben.

Danach werden die Parameter für Wirk- und Blindstrom-Regler automatisch berechnet. Das errechnete kP ist der maximal mögliche Wert für eine stabile Strom-Regelung! Üblicherweise muss der P-Anteil etwas reduziert werden um Pfeif-Geräusche zu reduzieren.

Bei Linear-Motoren werden üblicherweise sehr hohe P-Werte berechnet (ca. 50). Je nach Geräuschbildung beim aktivieren der Achse (F3, F2, F5) muss der P-Werte reduziert werden.

Bei Achsen mit markanten Resonanzstellen kann das berechnete kPq, kPd so hoch sein, dass die Achse gleich beim Aktivieren im Simulations-Mode laut pfeift. In diesem Fall sollten die kP-Werte so reduziert werden, dass das Pfeifen verschwindet.

Bei eisenlosen Linear-Motoren und hohen Abtastraten (16kHz) kann die Integrator-Zeitkonstante sehr klein werden: $< 200\mu\text{s}$

Wenn das kP vom Stromregler bei einer bereits eingestellten Achse erhöht wird, muss evtl. das kP vom Positions-Regler reduziert werden. Der Strom-Regler ist dem Positions-Regler unterlagert, deshalb besteht zwischen den beiden Reglern eine Abhängigkeit.

Blindstromregler

Der P-Anteil der Blindstromregelung muss nicht zwingend den gleichen Wert haben wie das P vom Wirkstromregler. Steigt der Blindstrom-Integrator nicht stark an, kann das Blindstrom-P stark reduziert werden. Vorallem bei Linearmotoren ist die Regelung sehr gut, so dass kPd auf ca die Hälfte von kPq reduziert werden kann.

Grundsätzlich kann bei grossen P-Werten (30 ... 50) das Wirkstrom-P reduziert werden bis die Geräusche erträglich sind. Das Blindstrom-P reduzieren bis der Integrator anzusteigen beginnt, typisch kPd = 1 ... 5.

Berechnungsgrundlagen

Der automatische Abgleich des Stromreglers basiert auf dem Widerstand und der Induktivität des Motors, weiter spielt die Abtastfrequenz des Drives eine Rolle.

I_{2t}-Regelung

I_{2tup}, I_{2tdown}
I_Int_Max

Siehe Kapitel 3.9 Stromregler: Current Control
3 mal I_{MAX}

10.16 Kommutierung

Mit der Kommutierung wird der Feldoffset (Winkel) zwischen dem elektrischen Feld im Stator und den Magneten des Rotors bestimmt.

Bei **Motoren mit Resolver** muss der Feldoffset nur ein einziges mal eingestellt werden. Üblicherweise liefern alle Motoren Hersteller ihre Motoren immer mit dem gleichen Feldoffset aus. D.h. der Resolver wird, relativ zum Rotor, immer genau gleich ausgerichtet montiert. Der Feldoffset von Resolvern wird im Motor Konfigurations-File abgespeichert.

Bei **Motoren mit einem Absolut-Messsystem**, muss der Feldoffset in der Regel jedes mal bei der Montage des Gebers auf den Motor bestimmt werden. Dies trifft auch auf den Service Fall zu, wenn ein Motor oder ein Geber ersetzt werden muss. Der Feldoffset von absolut-Messsystemen wird im Motor Konfigurations-File abgespeichert.

Motoren mit Inkrementalgeber oder Sinus-Cosinus Geber müssen bei jedem Einschalten des Drives kommutiert werden. Dazu stehen verschiedene Auto-Kommutierungs Methoden zur Verfügung.

Stimmt der Feldoffset nicht, kann der Motor nicht richtig betrieben werden. Dies äussert sich darin, dass z.B. die maximale Leistung oder die maximale Drehzahl nicht erreicht werden kann. Im schlimmsten Fall dreht der Motor rückwärts!

Ein optimal eingestellter Feld-Offset erhöht den Wirkungsgrad des Antriebes.

10.16.1 Auto-Kommutierung mit Sinus-Cosinus und Inkrementalgeber

Inkrementalgeber, Sinus-Cosinus Geber und Resolver können automatisch kommutiert werden.

1. Unter `Ctrl.MotorConfig.AutoCommutation` das Kommutierungsverfahren wählen und parametrieren. (Siehe 3.8 Auto Kommutierung)
2. Die Flags in der Autokommutierung wählen:
`ON_If_Ok=0` Wenn die Kommutierung erfolgreich war, bleibt die Achse aktiv.
`Again=1` Damit bleibt die Betriebsart immer in AutoCommutation.
3. Unter `Ctrl.Test` den `AutoCommutaion Mode` anwählen
4. Mit F3, F5 die Achse im Simulations-Mode aktivieren
5. Unter `Ctrl.Actual.AutoCommutation.Ok` überprüfen ob die Kommutierung erfolgreich war.
6. Die Kommutierung mehrmals an verschiedenen Positionen wiederholen.
Der `FieldOffset` `Ctrl.Actual.AutoCommutation` sollte über den gesamten Fahrbereich nicht mehr als $\pm 10^\circ$ abweichen.
7. Die Flags in der Autokommutierung zurücksetzen, Testmode auf `none`.
`ON_If_Ok=1`
`Again=0`

Die Achse niemals aktivieren (weder im Simulations-Mode noch im Aktive-Mode) wenn die Kommutierung nicht einwandfrei funktioniert!

10.16.2 Auto-Kommutierung mit Absolut-Encodern

1. Absolut-Geber konfigurieren, Auto-Kommutierungs Methode auf `Absolut Encoder` einstellen.
2. Den Feld-Offset von Hand bestimmen. (Kapitel 10.16.3 Resolver Offset von Hand abgleichen)
3. Unter `Ctrl.Test` den `AutoCommutaion Mode` anwählen
4. Auto-Kommutierung durchführen
5. Motor-Parameter ins Flash brennen und in ein File abspeichern.

10.16.2.1 Auto-Kommutierung mit Hiperface

Bei der ersten Inbetriebnahme eines Hiperface muss der richtige Feld-Offset gefunden und eingestellt werden. Folgende Schritte müssen bei der erstmaligen Inbetriebnahme des Motors zwingend durchgeführt werden.

1. Hiperface Konfigurieren.
2. Drehrichtung vom analogen (SinCos) sowie digitalem Feedback des Hiperface überprüfen. Die Zählrichtung beider Feedbacks müssen übereinstimmen (Zählrichtung der Inkremente).
3. Auto-Kommutierung auf Absolute Encoder setzen.
Folgendes Flag darf nicht gesetzt sein!
`ON>If_Ok=0`
4. Kommutierung mit Hiperface im Testmodus durchführen. Nun wird Position von Absolut-Geber gesynched.
5. Nun muss der Feld-Offset gefunden werden. Dazu stehen zwei Möglichkeiten zu Verfügung. Der Motor muss sich dazu frei bewegen können.
 - 1) Feld-Offset mittels Auto-Kommutierungsmodus abgleichen (siehe 3.8).
 - 2) Feld-Offset von Hand abgleichen (siehe 10.16.3)

Nun wurde der Feldoffset gefunden und unter `MotorConfig.FB_MotorField` eingetragen.

6. Nun Auto-Kommutierung wieder auf `Absolute Encoder` setzen. Mittels Testmodus Kommutierung testen. Zudem sollte der Drive ebenfalls einmal aus und wieder eingeschaltet werden. Dazu muss vorgängig das `MotorConfig` File gebrannt werden.
7. Wenn Kommutierung erfolgreich war kann folgendes Flag in Auto-Kommutierung gesetzt werden. Damit wird die Achse nach erfolgreicher Kommutierung automatisch aktiv geschaltet.

`ON>If_Ok=1`

10.16.3 Resolver Offset von Hand abgleichen

Um ein bestmögliches Resultat für den Feld-Offset von Resolvern zu erzielen, sollte der Offset von Hand abgeglichen werden.

Grundsätzlich kann eine der Auto-Kommutierungs Methoden verwendet werden, um einen ersten Wert für den Feld-Offset des Resolvers zu gewinnen. Bei Möglichkeit sollte die Auto-Kommutierung mit 360° Felddrehung eingesetzt werden (siehe Kapitel 3.8.5). Nach dem Ausführen der Auto-Kommutierung wird der Feld-Offset automatisch unter

`Ctrl.MotorConfig.FB_MotorField.Field_Offset` abgelegt.

Danach Auto-Kommutierung wieder ausschalten und nachfolgende Schritte für den genauen Feld-Offset durchführen.

1. Phasen und Resolver müssen korrekt angeschlossen sein.
Der Motor muss frei drehen können, ohne Last
2. Unter `Ctrl.Test` den `Set_Current` Mode anwählen
→ Blind- und Wirkstrom können von Hand eingegeben werden.
3. Externe Freigabe einschalten, grüne LED "Ext.En" am Servo-Regler muss leuchten
Safety-Eingänge müssen ebenfalls eingeschaltet sein.
4. ca. $1/10 I_{NENN}$ Blindstrom bei "`I_Reactive`" einstellen und Regler in den Simulationsmode schalten. Dazu F3, F2, F5 betätigen. Wenn der Motor dreht, ist der richtige Resolver-Offset noch nicht gefunden worden. Mit F3 Motor wieder ausschalten.
5. Resolver Offset `Ctrl.MotorConfig.FieldOffset` verändern bis Motor still steht.
(von Hand probieren in welche Richtung er noch leichter dreht)
6. Wenn der Motor stillsteht Blindstrom kontinuierlich auf I_{MAX} erhöhen. Nur kurzzeitig, ca. $\frac{1}{2}$ Sekunde `PowerOn/PowerOff` bzw. F3, F5, F3 Wiederholen bis der Motor bei `I_Reactive = I_{MAX}` stillsteht.
7. "`I_Reactive`" = 0V
8. Verifizieren, ob richtiger Resolver-Offset gefunden wurde:
9. Unter `Ctrl.Test` den `Set_Voltage` Mode anwählen
→ Blind- und Wirkspannung können von Hand eingegeben werden.
10. `U_torque` auf 1 ... 10V einstellen. Regler in den Simulationsmode schalten.
Dazu F3, F5 betätigen. Der Motor muss vorwärts drehen! Drehrichtung vorwärts = Von hinten auf den Motor schauen, der Motor muss im Uhrzeigersinn drehen.

Dreht der Motor nicht im Uhrzeigersinn, ist ein falscher Resolver-Offset gefunden worden.
(Es gibt mehrere).
11. `BurnToFlashProm`, `SaveToFile`, `U_torque` = 0V, `Flag` = 0

10.16.4 Spezialfall Z-Achse mit 360 deg FieldRotation

Die 360deg FieldRotation Kommutierung findet bei einer Waagerechten Achse immer den idealen Winkel, da sich der +- Last-Winkel aufhebt. Bei hängenden Z-Achsen geht das leider nicht, da die Last immer nach unten zieht. Dieser Winkel beträgt je nach Last 10...25 FldDeg.

Dieser Winkel kann Mathematisch nicht erfasst werden. Bei gleichbleibender Last sollte der Winkel jedoch immer etwa gleich sein. Der Winkel kann unter

`MotorConfig-AutoCommutation.Z_Axis_FieldOffset`

hinterlegt werden. Dieser wird nach der Kommutierung jeweils automatisch dazu gerechnet.

Den Optimalen Wert finden Sie, indem sie mit `Z_Axis_FieldOffset=0` Kommutieren und sich den gefundenen Wert unter `Ctrl.Actual.AutoCommutation` notieren. Lassen Sie dann die Achse Rauf/Runter toggeln und loggen Sie dabei `Iq` immer auf der Fahrt nach oben. Verändern Sie den Wert

`Ctrl.Actual.AutoCommutation`

in ca 5 Grad Schritten, `Iq` sollte erst immer kleiner und dann wieder grösser werden. Die Differenz vom notierten Wert zum neu gefundenen Wert mit dem kleinsten `Iq` ist dann unter `Z_Axis_FieldOffset` einzutragen.

10.17 Drehrichtung verifizieren (nach Kommutierung)

Für diesen Test muss die Kommutierung erfolgreich durchgeführt worden sein.

1. Den Motor ohne Last betreiben! Unter `Test` den `Set_Voltage` Mode anwählen
2. Spannung `U_torque` wählen, mit 1V beginnen, Spannungs-Wert langsam erhöhen.
Achtung: Der Motor dreht schneller mit 1V als beim Field-Rotation Mode!
3. Regler in den Simulationsmode schalten. Dazu F3, F2, F5 betätigen.
Der Motor muss nun vorwärts drehen und vorwärts zählen!

10.18 Gain- Offset Korrektur für Resolver und SinCos

Gain- und Offset-Fehler sowie Gain-Asymmetrie von Resolver bzw. SinCos Gebersystemen können softwaremässig ausgewertet und korrigiert werden.

Dazu kann der Assistent „SinCos“ verwendet werden. Der Assistent ist Bestandteil der Software direkt im Drive.

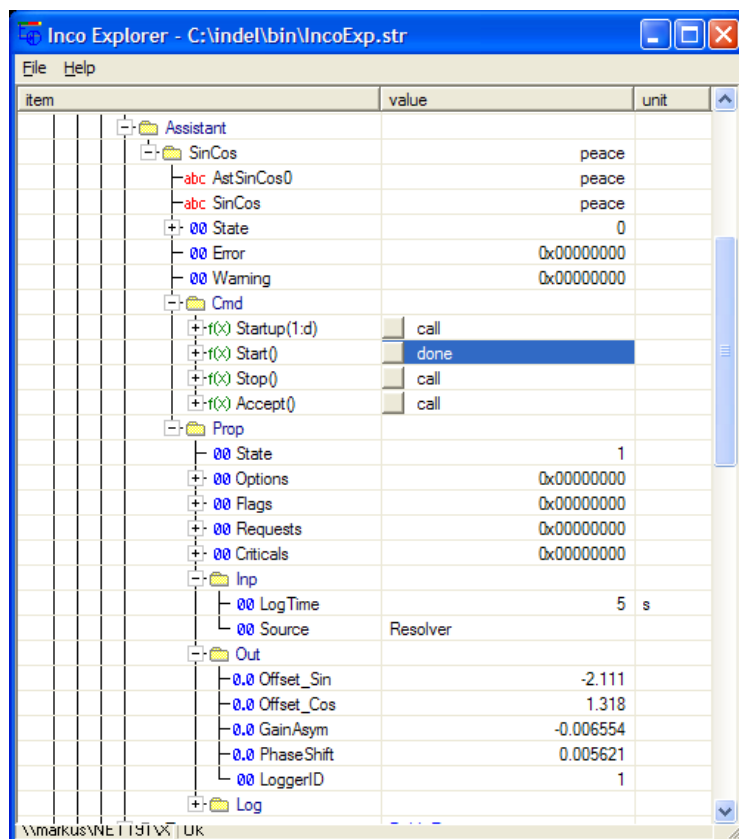


Abbildung 10.1: SinCos Abgleich

10.18.1 Abgleich Resolver

Abgleich Resolver

1. Motor mit konstant 100 U/min laufen lassen (entspricht 600°/s)
2. SinCos Assistenten starten `Start()`
3. Berechnung von Gain, Offset, usw. abwarten
4. Abgleichwerte übernehmen `Accept()`
5. Parameter brennen, speichern

`Ctrl.Assistant.SinCos.Cmd.Start()` gestartet werden. Die Logger-Zeit kann auf 5s (Defaultwert) belassen werden.

Der Zustand des Ablaufes wird im Inco Explorer angezeigt:

- Waiting for logger, please move axis
- Logging
- Calculating Correction Values
- Peace

Nach Ablauf des Tests werden unter `Ctrl.Assistant.SinCos.Prop.Out` die Ergebnisse des Abgleiches angezeigt.

Diese müssen nun mit `Ctrl.Assistant.SinCos.Cmd.Accept()` übernommen werden. Zum Schluss müssen die abgeglichenen Werte ins Flash-Prom gebrannt und abgespeichert werden.

10.18.2 Abgleich SinCos Geber

Die Achse möglichst über den gesamten Bereich des Messsystems bewegen. Bei Rotativen Messsystemen genügt mindestens eine Motor-Umdrehung, bei linearen Messsystemen den gesamten Masstab abfahren.

Die Geschwindigkeit muss so gewählt werden, dass pro Abtastperiode mindestens 40 Messwerte einer vollen Sinus-Schwingung aufgezeichnet werden können.

Beispiel

Rotativer Geber mit 1024 Sinusperioden
 Abtastrate 16kHz

$$\text{Drehzahl für SinCos Abgleich} = \frac{16000 \text{ kHz}}{1024 * 40} = 0.39 \text{ U/s (140°/s)}$$

Unter `Ctrl.Assistant.SinCos.Prop.Inp.LogTime` kann die Zeit für den Log eingegeben werden. Hier wird die Zeit für die gesamte Bewegung eingetragen.

Abgleich SinCos

1. Motor über gesamte Länge des Messsystems laufen lassen
2. Zeit für die Bewegung eingeben `LogTime`
3. SinCos Assistenten starten `Start()`
4. Berechnung von Gain, Offset, usw. abwarten
5. Abgleichwerte übernehmen `Accept()`
6. Parameter brennen, speichern

10.19 PID Parameter abgleichen

Grundsätzlich müssen die PID-Parameter für den Motor mit Last abgeglichen werden. Wenn Sie das erste mal PID-Parameter abgleichen, verwenden Sie sicherheitshalber einen Motor ohne Last! Schwingt der Motor zu stark, kann evtl. die Mechanik darunter leiden.

Die PID-Parameter können mit verschiedenen Ansätzen eruiert werden:

- Stossantwort, Sprungantwort; Optimierung nach Chien, Hrones und Reswick
- Optimierungsverfahren nach Ziegler und Nichols
- „Axis Tuner“ Tool von Indel, siehe Kapitel 11 Bode-Sweep – PID-Wizard

10.19.1 Optimierungsverfahren nach Ziegler-Nichols

Die Regelstrecke wird zunächst mit einem reinen P-Regler betrieben. Der P-Anteil wird soweit erhöht, bis die Regelstrecke in Schwingungen konstanter Amplitude gerät. Aus dem eingestellten kritischen P-Anteil (kP_{KRIT}) und der Periodendauer (T_k) der entstehenden Schwingung, kann der Regler optimal eingestellt werden.

Reglertyp	I-Anteil	D-Anteil	P-Anteil
P	–	–	$0.5 * kP_{KRIT}$
PD	–	$0.125 * T_{KRIT}$	$0.8 * kP_{KRIT}$
PI	$0.85 * T_{KRIT}$	–	$0.45 * kP_{KRIT}$
PID	$0.5 * T_{KRIT}$	$0.125 * T_{KRIT}$	$0.6 * kP_{KRIT}$

10.19.2 Vorgehen beim Abgleich der PID-Parameter

Folgende Parameter mit 1kHz loggen

Zwischenkreisspannung	U_{CC}	Ctrl.Actual.Power
Sollgeschwindigkeit	V_{CMD}	Ctrl.Actual.PositionCtrl.cmd_V
Istgeschwindigkeit	V_{ACT}	Ctrl.Actual.PositionCtrl.act_V
Schleppfehler	S_{ERR}	Ctrl.Actual.PositionCtrl.err_S
Integrator	S_{INT}	Ctrl.Actual.PositionCtrl.err_S_int
Wirkstrom	I_Q	Ctrl.Actual.CurrentCtrl.act_Iq

1. PID-Parameter im Inco-Tree: Ctrl.MotorConfig.PositionCtrl
2. Alle Filter auf none stellen: Ctrl.MotorConfig.Filter.Filter_0,1,2,3
Speed-Filter ebenfalls auf none stellen: Ctrl.MotorConfig.SpeedFilter

3. Default-Parameter:

HoldToStandbyTime	=	100ms	
PID standby	=	0	alle Parameter auf 0 setzen
PID forward	=	0	alle Parameter auf 0 setzen
PID backward	=	0	alle Parameter auf 0 setzen
phvSpeed, phvAcc	=	0	alle Parameter auf 0 setzen

4. Als Anfangswert wählen Sie für $k_P = I_{NOM} / 50$
Ctrl.MotorConfig.PositionCtrl.forward.kP

Beispiel Motor mit $I_{NENN} = 2.4 A \rightarrow k_P = 0.05$

5. Wechseln Sie zum Feldbus-Master. Geben Sie eine kleine Rampe vor:

Axis.PhysicalAxis0.Cmd.Test.cmdPos1	=	360°
Axis.PhysicalAxis0.Ramp.Cmd.cmdV	=	600 %/S (100 U/min)
Axis.PhysicalAxis0.Ramp.Cmd.cmdA	=	5000 %/S ²
Axis.PhysicalAxis0.Ramp.Cmd.cmdB	=	5000 %/S ²
Axis.PhysicalAxis0.Ramp.Cmd.cmdJ	=	0%

Es kann auch eine steilere Rampe gewählt werden z.B. 50'000 %/s². Damit wird erreicht, dass Eigenresonanzen eher angeregt werden.

Motor-Kenndaten und Last mit berücksichtigen!

Fahren Sie diese Rampe im **Simulationsmode F3, F5**. Start Rampenfahrt mit F7.
Falls der Motor extrem schwingt oder vibriert, schalten Sie sofort wieder aus mit F3 (Inaktiv) oder F8 (Stop) und verkleinern Sie den Wert von k_P .

Diese Angaben gelten für ein Getriebe von 1:1, bzw. beziehen sich auf die Motorwelle!

6. Erhöhen Sie "kP PID forward" bis die Schwingung im waagrechten Abschnitt des gefahrenen Trapezes nicht zu- oder abnimmt. -> Logger verwenden.

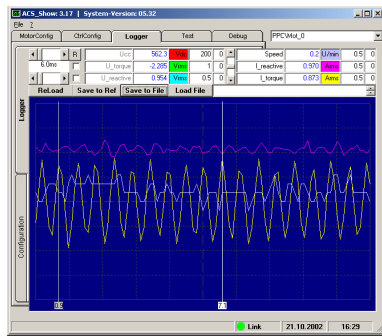


Abb 54: Kritische Verstärkung k_P krit

Aus Abb 7.9.1: dies ist die kritische Verstärkung " k_P krit", die mit dem Logger gemessene Periode der Schwingung ist die kritische Periode Zeit " T_k ".

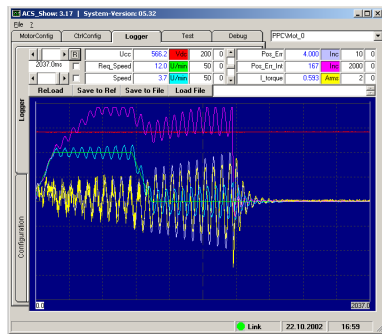


Abb 55: Kritische Verstärkung k_P krit zu hoch

7. Wenden Sie folgende Formeln an für k_P , k_I , und k_D :

$$\begin{aligned} k_P &= 0.6 * k_{Pkrit} \\ k_I &= 0.5 * T_k \\ k_D &= 0.12 * T_k \end{aligned}$$

Aus Beispiel:

$$\begin{aligned} k_P &= 0.6 * 0.13 = 0.78 \\ k_I &= 0.5 * 54 = 27 \text{ ms} \\ k_D &= 0.12 * 54 = 6.48 \text{ ms} \end{aligned}$$



Abb 56: Erste Version der PID-Parameter

8. Erhöhen Sie den Wert für k_P kontinuierlich. Damit wird der Schleppfehler verkleinert und die Kurve wird glatter. Gleichzeitig sollte auch die Rampe verändert werden:

$$\begin{aligned} \text{Axis.PhysicalAxis0.Cmd.Test.cmdPos1} &= 3'600^\circ \\ \text{Axis.PhysicalAxis0.Ramp.Cmd.cmdV} &= 6'000 \text{ }^\circ/\text{S} \text{ (100 U/min)} \\ \text{Axis.PhysicalAxis0.Ramp.Cmd.cmdA} &= 5'000 \text{ }^\circ/\text{S}^2 \\ \text{Axis.PhysicalAxis0.Ramp.Cmd.cmdB} &= 5'000 \text{ }^\circ/\text{S}^2 \\ \text{Axis.PhysicalAxis0.Ramp.Cmd.cmdJ} &= 0\% \end{aligned}$$

9. Wiederholen Sie obige Schritte 6 und 7 bis Sie eine Kurvenform mit folgenden Kriterien erreichen:
- Req_Speed und Speed sind deckungsgleich
 - Nach gefahrener Rampe stimmt der Parameter ANZ_S mit der Vorgabe VRG_S auf +- 1% überein.
 - Kein extremes Schwingverhalten bei Strom und Schleppfehler.

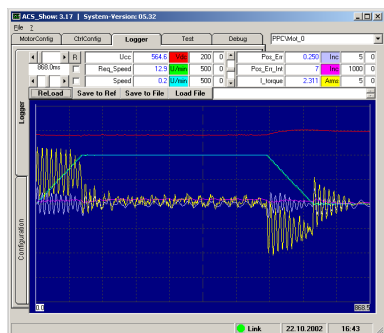


Abb 57: *kP wurde erhöht bis Strom und Schleppfehler wieder schwingen*

10. Aus Beispiel: `kP` wurde erhöht bis auf 0.6 -> bei dieser Einstellung beginnt der Strom und der Schleppfehler erneut zu schwingen.
- `kP` wieder verkleinern auf ca. 80 ... 90% vom kritischen Wert: `kP` = 0.5
11. Die PID Parameter für "PID standby" ergeben sich aus dem "Durchschnitt" von `forward` und `backward`. Für `kP standby` geben sie ca. die Hälfte des Durchschnittes ein.
12. Mit diesen Parametern sollte der Motor grundsätzlich laufen. Der noch nötige Feinabgleich bzw. der Abgleich mit Last erfordert ein hohes Mass an Erfahrung und manchmal auch Geduld. Gerne stehen wir Ihnen zur Verfügung und helfen Ihnen beim Einstellen der Regelgewichte.
13. Bis jetzt ist der Motor immer im Simulationsbetrieb gelaufen. Für den Feinabgleich wird der Motor in der Betriebsart "Aktive" (Funktionstaste F4) betrieben. Damit wird die übergeordnete Lage-Regelung eingeschaltet.
14. Die gefundenen Parameter müssen mit "Burn Values to Target" ins Flash-Prom des Reglers geladen werden, ansonsten gehen die Werte verloren bei Power-down. Zum Schluss sollten die Werte in ein File gespeichert werden mit "Save Values to File".
15. Wenden Sie das gleiche Prozedere für `PID backward` an.

10.20 Vorhaltewerte abgleichen

phvSpeed

Der Vorhaltewert für Geschwindigkeit ist eine Soll-Strom Vorgabe. Sie hebt geschwindigkeitsabhängige Verluste auf.

1. Messe Strom während möglichst hoher Konstant-Fahrt
2. Geschwindigkeitsabhängiger Vorhaltewert:

$$\mathbf{phvSpeed = Konst-Strom / Konst-v}$$

Beispiel:

Konstante Fahrt mit 3000 U/min mit 6A

$$\mathbf{phvSpeed = 6A / 3(1000)U/min = 6 / 3 = 2 A@Spd}$$

3. Der Integrator (Pos_Err_int) sollte jetzt während kontinuierlicher Fahrt konstant bleiben.
Optimal: Pos_Err_int < 1000 Inc

phvAcc

Ist ein Beschleunigungs- bzw. Verzögerungs-Vorhalte Wert, der den für die Beschleunigung / Verzögerung benötigten Strom zum Sollwert addiert.

PhvAcc ist als Stromaufnahme in A bei einer Beschleunigung / Verzögerung von 0 auf 1000U/min in einer sec definiert.

1. Messe Strom während möglichst langer Verzögerungs-Fahrt
-> ergibt zB 12A für Verzögerung in 250ms von 0 auf 3000U/min
2. $\mathbf{phvAcc = 12Arms / 3 (1000)U/min * 025sec = 12 / 3 * 0.25 = 1 A@acc}$
3. Der Integrator (Pos_Err_int) sollte jetzt während Beschleunigungs- und Brems-Rampe symetrisch sein.

10.21 Feinabgleich von Ke, Rs und Ls

Dieser Abgleich ist nicht zwingend notwendig. Wenn die Daten aus dem Datenblatt richtig normiert auf Strangwerte (Phase-Phase) eingegeben werden, muss der Motor auch einwandfrei arbeiten!

Der Feinabgleich kann auch zur Verifizierung der Werte aus dem Datenblatt verwendet werden. Damit werden Fehler, die aus evtl. alten Daten-Blättern oder falschen Angaben stammen frühzeitig eliminiert. Wenn das Ke falsch ist, hat dies eine direkte Auswirkung auf die Dynamik der Regelung.

Bei Applikationen mit grossen Lasten oder extrem schnellen Rampen lohnt sich dieser zusätzliche Feinabgleich ebenfalls.

Änderungen von Ke, Rs, Ls werden nur übernommen wenn die Achse inaktive und wieder aktive geschaltet wird.

Uq	U Torque	Wirkspannung
Ud	U Reactive	Blindspannung
Iq	I Torque	Wirkstrom
Id	I Reactive	Blindstrom
Err_Iq_Int	I Torque Integrator	Integrator Wirkstrom (Stromregler)
Err_Id_Int	I Reactive Integrator	Integrator Blindstrom (Stromregler)

Die Reihenfolge muss eingehalten werden!

1. Feldoffset (Resolver-Offset) im Stillstand und mit I_{max} einstellen
2. Ke so einstellen, dass Err_Iq_Int bei voller Drehzahl (ohne Last) möglichst 0 ist
 $U_q = I_q \cdot R_s + wKe = I_q=0 = 0 + wKe$ bleibt nur -Fehler
3. Ls so einstellen, dass Err_Id_Int während Rampe Auf/Ab möglichst 0 ist (symmetrisch)
 Wenn Err_Id_Int bei Rampe Auf neg: Ls zu klein
 Wenn Err_Id_Int bei Rampe Auf pos: Ls zu gross
4. Rs so einstellen, dass Err_Iq_Int während Rampe möglichst konstant (und nahe 0 ist).
 Wenn Err_Iq_Int bei Rampe Auf neg: Rs zu klein
 Wenn Err_Iq_Int bei Rampe Auf pos: Rs zu gross

10.22 Resonanzen beseitigen

Methoden um hochfrequente Resonanzen, hervorgerufen durch Getriebespiel oder Riemenspiel zu reduzieren:

Speed-Filter: kT_Speed

Resolver-Filter, Zeitkonstante für "Resolver-Geschwindigkeit" (1ms).

`Ctrl.MotorConfig.PositionCtrl.kT_Speed`

- kT_Speed in kleinen Schritten erhöhen.
- kT_Speed kann ca. 5 ... 10 mal kleiner als kD gewählt werden.

k_d

`Ctrl.MotorConfig.PositionCtrl.forward.kd`

- k_d wirkt am besten mit SinCos Gebern (hochauflösende Geber)
- kann auch bei Resolvern verwendet werden, evtl. mit begrenzter Wirkung
- ein zu grosses k_d erzeugt pfeifende Geräusche

kD

`Ctrl.MotorConfig.PositionCtrl.forward.kD`

- kD so klein wie möglich
- bei kleinerem kD kann kP erhöht werden

Stromregler

`Ctrl.MotorConfig.CurrentCtrl.kPq`

`Ctrl.MotorConfig.CurrentCtrl.kIq`

`Ctrl.MotorConfig.CurrentCtrl.kPd`

`Ctrl.MotorConfig.CurrentCtrl.kId`

- kPq , kPd verkleinern

Totzeitkompensation

`Ctrl.MotorConfig.PWM.DeadTime_correction`

- Totzeit-Kompensation ausschalten (none)

Methoden um niederfrequente Resonanzen, hervorgerufen durch grosse Massen, zu reduzieren:

Positions-Integrator

`Ctrl.MotorConfig.PositionCtrl.Pos_Int_Max`

- Integrator-Begrenzung (beliebig) verkleinern (Standard: 7600)

kD

`Ctrl.MotorConfig.PositionCtrl.kD`

- D-Anteil der Regelung vergrössern

Frequenz-Filter 0 ... 3

Filter mit Bode-Sweep abgleichen.

11 Bode-Sweep – PID-Wizard

Mit einem Bode-Sweep können die bestehenden PID-Parameter optimiert werden. Zusätzlich können bis zu 4 BiQuad Filter konfiguriert werden um störende Resonanzen / Dissonanzen zu eliminieren.

11.1 Motion Tool Einstellungen

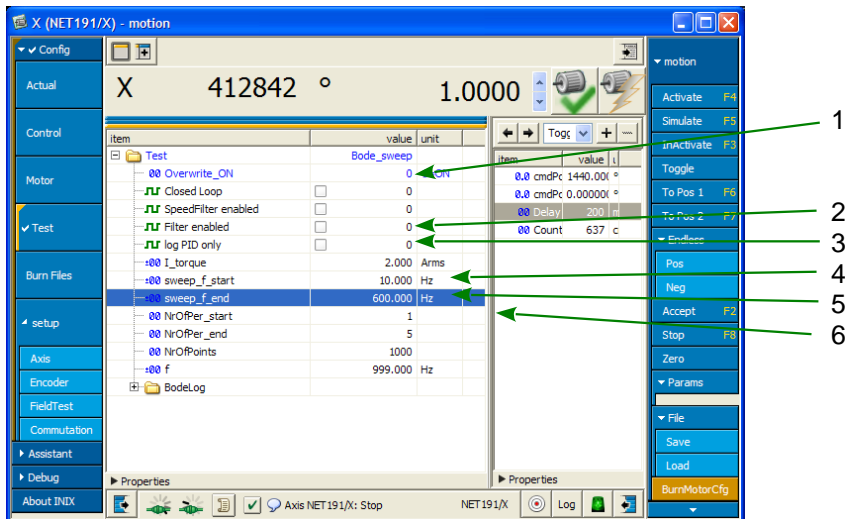


Abbildung 11.1: Sweep konfigurieren

- 1) Test-Mode: Bode-Sweep (Drop-Down Menü)
- 2) Speed Filter aktivieren für Bode-Sweep
Wenn ein Speed-Filter konfiguriert ist, muss dieses Flag beim Bode-Sweep eingeschaltet sein. Siehe auch Kapitel Fehler: Referenz nicht gefunden Fehler: Referenz nicht gefunden
- 3) Strom Filter aktivieren für Bode-Sweep
Die Strom-Filter sollten für den Bode-Sweep nur dann eingeschaltet sein, wenn derart starke Resonanzen vorhanden sind, dass kein aussagekräftiger Bode-Sweep möglich ist.
- 4) Wirkstrom für Bode-Sweep
- 5) Start-Frequenz
- 6) End-Frequenz

11.2 PID-Wizard Einstellungen

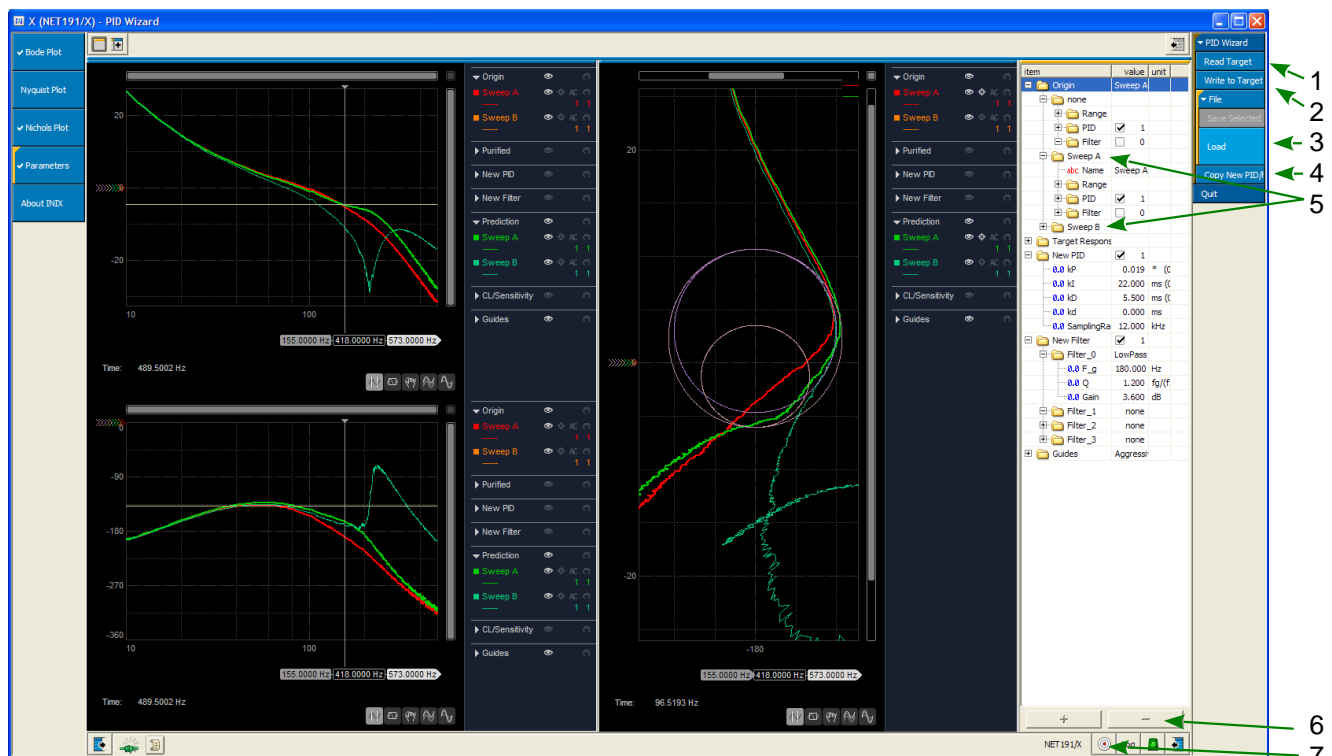


Abbildung 11.2: PID Wizard

- 1) Read Target: Sweep-Daten aus Regler laden
- 2) Write to Target: Filter-Werte und PID-Parameter aus den Verzeichnissen New Filter und New PID in den Drive kopieren
- 3) Speichern, Laden von Sweeps
Vor dem Speichern eines Sweeps muss dieser im Variablen-Baum angewählt werden. Siehe auch Punkt 5).
- 4) PID-Parameter und Filter Einstellungen aus dem Drive in den PID-Wizard kopieren (Verzeichnisse New Filter und New PID)
- 5) Auswahl eines Sweeps
- 6) Löschen von einem Filter oder von einem gesamten Sweep.
Den Sweep/Filter im Variablen-Baum anwählen, danach löschen
- 7) Target Auswahl: zuerst das Target wählen, danach den Drive innerhalb des Targets wählen

11.3 Bode Sweep aufzeichnen

Grundsätzlich wird der erste Sweep mit grob eingestellten PID-Parametern und ohne jegliche Filter ausgeführt.

Vor Beginn muss der Stromregler und der SinCos (Resolver) abgeglichen werden. Siehe dazu:

10.18 Gain- Offset Korrektur für Resolver und SinCos
10.15 Stromregler abgleichen

1. Im Motion-Tool den Test-Mode `Bode_Sweep` anwählen
2. Flags für `SpeedFilter`, `Filter enabled` löschen
3. Beim ersten Sweep `I_Torque` auf ca. 1/4 ... 1/3 von I_{NENN} einstellen, besser mit einem geringen Strom beginnen um Motor und Mechanik nicht zu überlasten.

Die I^2t Regelung sollte ebenfalls korrekt eingestellt sein um Motor-Überlast zu vermeiden. (Siehe Kap. 3.9.3 I^2t Regelung)

Der Sweep-Strom sollte schlussendlich dem Strom, der im Betrieb tatsächlich fließt, entsprechen.
4. Drive im Simulations-Mode aktivieren (F3, F2, F5)
5. → Der Frequenz-Sweep wird ausgeführt
6. Im PID-Wizard `Read Target` anklicken
7. → Der Log wird geladen und angezeigt
8. Nach dem Bode-Sweep den Test-Mode wieder auf `None` setzen.

11.4 Vorgehen beim Optimieren der Regelstrecke

Erster Sweep

In einem ersten Schritt einen Sweep mit grob eingestellten PID-Parametern und moderatem Strom erstellen.

Observer-Filter

Sollten im oberen Frequenzbereich bereits Resonanzen vorhanden sein, oder bei Gebern mit sehr geringer Auflösung empfiehlt es sich das Observer-Filter zu konfigurieren. Siehe Fehler: Referenz nicht gefunden Fehler: Referenz nicht gefunden.

Zuerst nur die Grenzfrequenz des Filters setzen, den Wert `kP_Iq = 0` eingeben.

Bei Gebern mit sehr geringer Auflösung ab 2048 Inkrementen pro Umdrehung kann die Grenzfrequenz zwischen `F_g = 180 ... 450Hz` angesetzt werden.

Zu tiefe Werte für die Grenzfrequenz beeinträchtigen die Bandbreite des Systems und verschlechtern evtl. die Regelgüte.

Für hochauflösende SinCos-Geber mit einer interpolierten Auflösung grösser 1'000'000 Inkremente pro Umdrehung beträgt die Grenzfrequenz beispielsweise `F_g = 600 ... 1000Hz`.

Bei eingeschaltetem Observer-Filter muss das Flag `SpeedFilter enabled` eingeschaltet sein. Siehe 11.2 PID-Wizard Einstellungen.

Strom erhöhen

Danach den Sweep-Strom erhöhen und so einstellen, dass er in etwa dem benötigten Strom für die Applikation entspricht.

Phasenreserve im unteren Frequenzbereich gewinnen

Um mehr Phasenreserve im unteren Frequenzbereich zu gewinnen hat es sich mittlerweile bewährt einen Tiefpass-Filter mit positivem Gain zu verwenden.

Das Gain kann bis ca. 6dB konfiguriert werden mit einem Gütefaktor (Q) zwischen 0.9 und 1.1. Siehe dazu auch Abb. 63 und Abb. 64 in Kapitel 11.7.

PID-Parameter und Filter optimieren

Zuerst geeignete PID-Parameter definieren, danach mit der Konfiguration der Filter beginnen. In der Regel werden die beste Resultate mit optimalen PID-Parametern und Notch-Filter erzielt.

11.5 Bode Sweep auswerten

Signale kleiner als -35dB sind für die Regelung und Optimierung der Filter nicht mehr relevant. Der Parameter `Ctrl.Test.sweep_f_end` kann dementsprechend angepasst werden.

In diesem Beispiel wird er von 1000 Hz (Default) auf 600 Hz reduziert.

Der PID-Wizard kann gleichzeitig nur Sweeps mit den gleichen Start-Stopp Frequenzen anzeigen!

Bevor die Start-Stopp Frequenz verändert wird sollten zuerst alle Sweeps im PID-Wizard gelöscht werden. (Siehe Sweep Löschen Seite: 106 Paragraph: 5)



Abbildung 11.3: Max. Frequenz einstellen

Die restlichen Parameter sollten nicht verändert werden (ausser dem Strom).

Zu Beginn der Optimierungsarbeit müssen die bestehenden PID-Parameter und Filterwerte noch in den PID-Wizard übernommen werden.

Dazu `CopyNewPID/Filter from Origin` anklicken. Die Parameter werden in die Verzeichnisse `New PID` und `New Filter` kopiert.

Danach können Filter und PID-Parameter in den Verzeichnissen **New PID** und **New Filter** zur Optimierung angepasst werden.

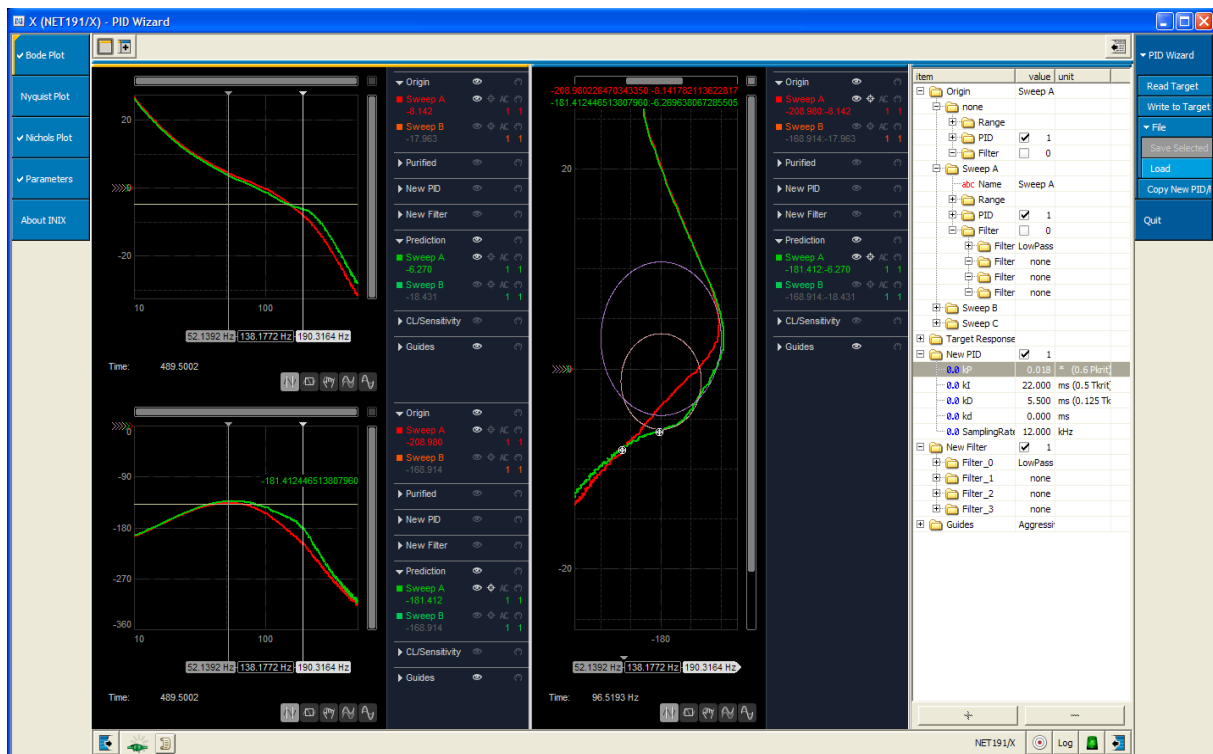


Abbildung 11.4: Filter, PID einstellen

Einstell-Hilfen

Unter **Guides** können Hilfslinien und Hilfskreise eingeblendet werden. Diese Hilfslinien beinhalten verschiedene Stabilitäts-Kriterien.

Es können drei verschiedene „Schärfe-Grade“ für die Stabilität ausgewählt werden:

- Aggressive
- Moderate
- Conservative

Wobei die Regelung meist mit **Aggressive** Grenzwerten ausgelegt wird.

- AR** Amplitudenrand, 4.6 dB bei Aggressive:
Beim -180° Durchtritt der Phase bleibt eine minimale Dämpfung von -4.6 dB im Openloop
- PR** Phasenrand
Beim 0-Druchgang des Signals bleibt eine Phasen-Reserve von 41.4°
- Mt** Stabilitäts-Grenze, 3 dB Reserve bei Aggressive
Im Closed-Loop ist die Überhöhung bei Einschwingverhalten nie grösser als 3 dB
- Ms** Sensibilitäts-Grenze, 6 dB Reserve bei Aggressive

11.6 Wirkung der PID-Parameter

kP-Anteil

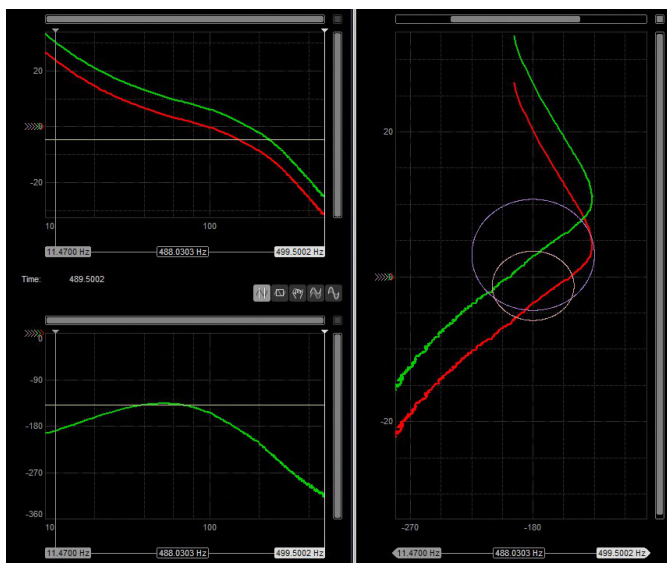


Abb 58: Wirkung des kP-Anteils

Erhöhen des kP-Anteils bewirkt im Bode- und Nichols-Diagramm die Verschiebung der gesamten Kurve nach oben.

kI-Anteil

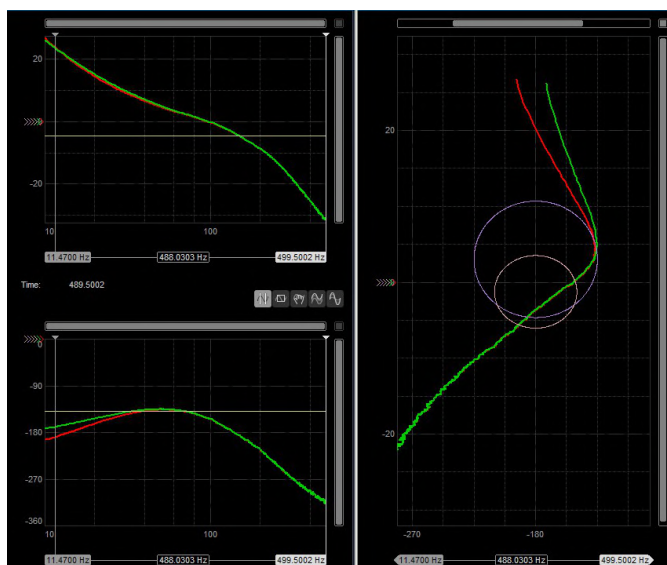
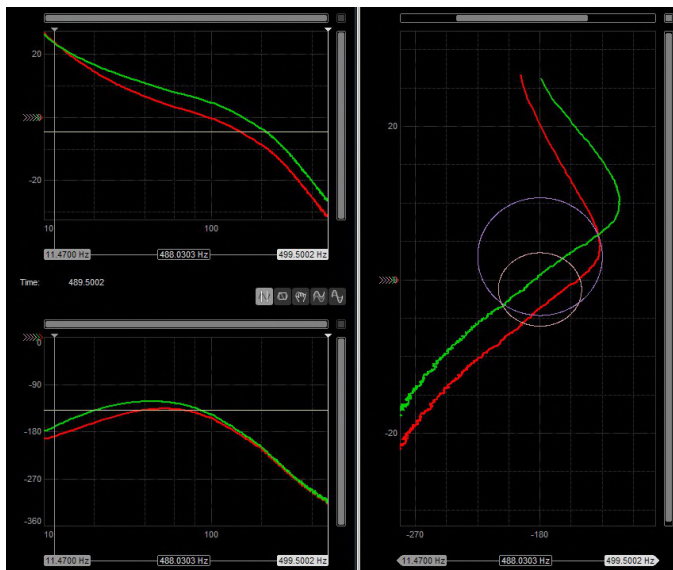


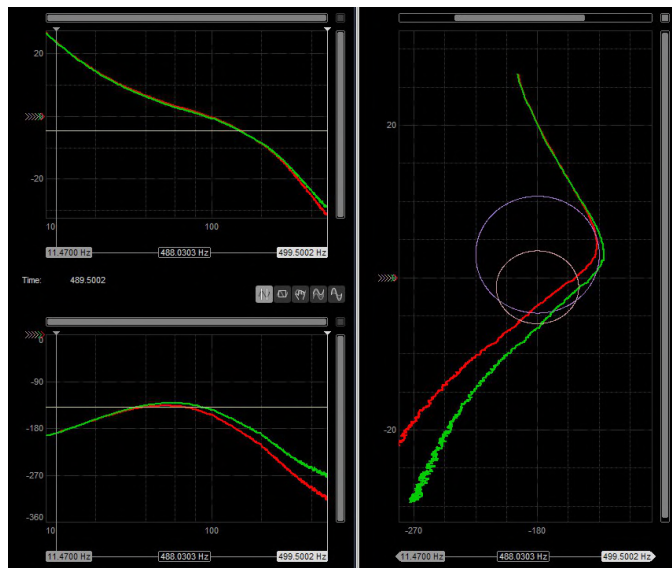
Abb 59: Wirkung des kI-Anteils

Erhöhen des kI-Anteils bewirkt eine Anhebung der Phasen-Reserve im tiefen Frequenzbereich und ein nach rechts Schwenken der Kurve im Nichols-Diagramm.

Das Verhältnis zwischen I-Anteil und D-Anteil sollte wenn möglich immer 4:1 sein. I-Anteil 4 x grösser als D-Anteil. Dies entspricht der Theorie von Ziegler-Nichols.

kD-AnteilAbb 60: Wirkung des *kD*-Anteils

Erhöhung des *kD*-Anteils verschiebt die Kurve im Nichols Diagramm nach rechts und nach oben. Nebst der Verstärkung des Pegels wächst auch die Phasen-Reserve.

***kd*-Anteil**Abb 61: Wirkung des *kd*-Anteils

Der *kd*-Anteil erhöht die Phasenreserve im oberen Frequenzbereich. Voraussetzung für die optimale Wirkung dieses Parameters ist eine hohe Auflösung des Gebersystems.

Die schnellen Änderungen im hohen Frequenzbereich müssen mit einer entsprechenden „Menge“ an Weginformation verarbeitet werden können. Am besten wirkt das *kd* mit hochauflösenden SinCos Gebern.

Das *kd* kann nicht beliebig gross gewählt werden. Die Regelstrecke reagiert mit lauten Geräuschen und Strom-Rauschen.

11.7 **Strom-Filter**

Je nach Motion-Board bzw. Servo-Regler können 1 ... 4 Strom-Filter konfiguriert werden.

Grundsätzlich können so viele Filter konfiguriert werden, wie Rechenleistung vorhanden ist. Die Anzahl Filter hängt deshalb auch von der Abtastrate des Positions-Kontrollers ab.

Bei AX-4 bzw. MAX4 Boards mit eigener Applikation kann unter Umständen keine Rechenleistung mehr vorhanden sein für Strom-Filter.

Es stehen 3 verschiedene Filter-Typen zur Verfügung: Low-Pass, Notch (Bandsperr) und Two-Load.

Filter-Parameter

Die Güte kann im Bereich von 0.5 ... 5 verändert werden.

Die Dämpfung kann zwischen -80 ... +6dB verändert werden. Verstärkungen grösser als 1dB sind in der realen Regelstrecke evtl. nicht mehr umsetzbar.

Low-Pass

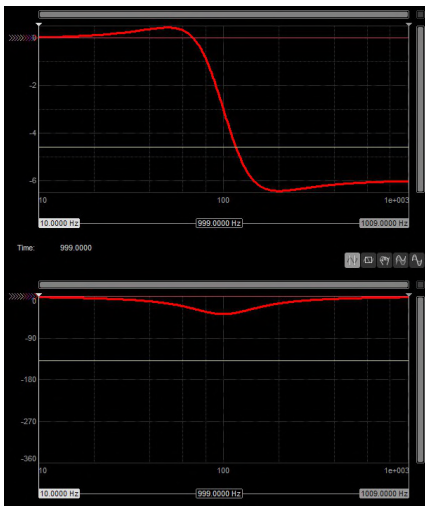


Abb 62: Low-Pass Filter -6dB Verstärkung

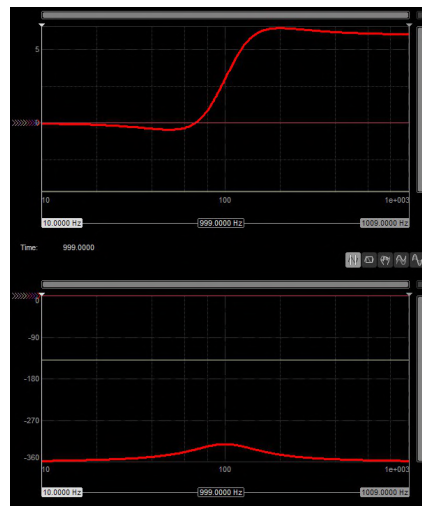


Abb 63: Low-Pass Filter 6dB Verstärkung

Low-Pass Filter mit positiver Verstärkung können verwendet werden um die Phasenreserve zu erhöhen.

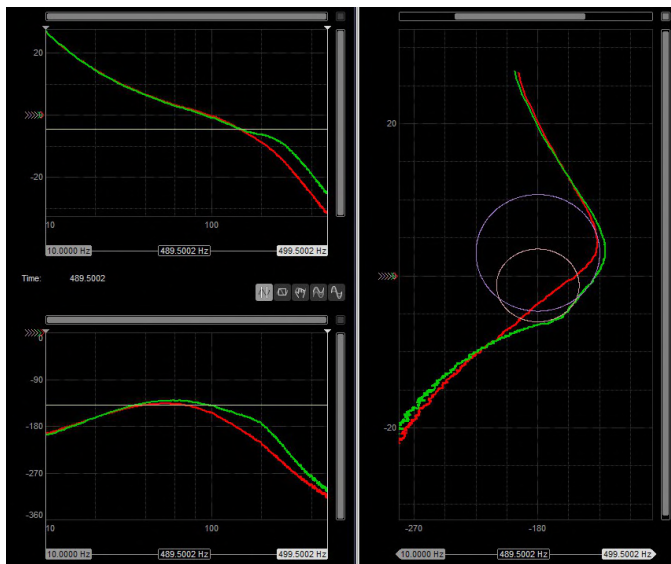


Abb 64: Beispiel: Low Pass Filter mit 6dB Verstärkung bei 210Hz

Notch (Bandpass)



Abb 65: Notch-Filter -6dB Dämpfung

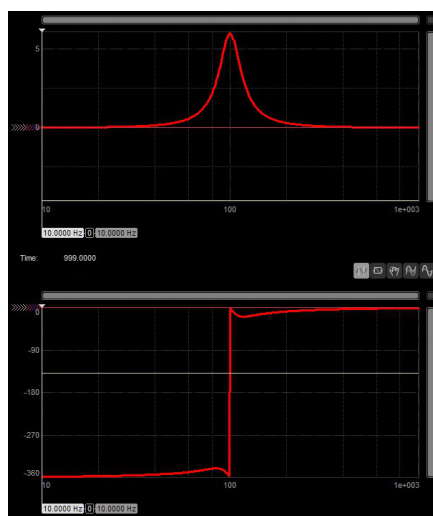


Abb 66: Notch-Filter 6dB Verstärkung

Bandpass-Filter eignen sich besonders um Resonanzüberhöhungen zu reduzieren oder gezielt die Pegel anzuheben.

Two-Load

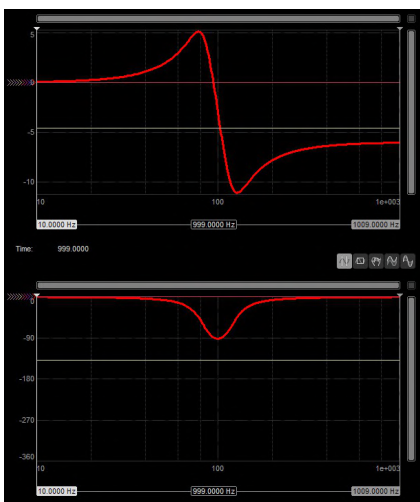


Abb 67: Two-Load Filter Güte=3

Two-Load Filter mit erhöhter Güte können verwendet werden um eine Pol-Nullstelle zu kompensieren. Dies ist jedoch mit Vorsicht anzuwenden, da bei einer geringen Verschiebung der Resonanz das Filter unter Umständen kontraproduktiv wirkt.

11.8 Optimierungsregeln

Die Ortskurve des Ziegler-Nichols Diagrammes darf die beiden Kreise `Mt Complementary Sensitivity Circle` und `Ms Sensitivity Circle` nicht berühren.

Das Gain der Filter darf positiv oder negativ sein. Speed Filter

Die Speed Filter wirken direkt auf die Ist-Geschwindigkeit. Wird eines der Speed-Filter verwendet, muss das Flag `Ctrl.Test.SpeedFilter enabled` beim Sweepen eingeschaltet werden

11.8.1 Observer Filter

Mit Hilfe des Geschwindigkeits-Beobachters können höherfrequente Resonanzen und Resonanzen hervorgerufen durch „elastischen Encoderanbau“ weitgehend beseitigt werden.

Das Observer Filter besteht einerseits aus einem Tiefpassfilter mit Grenzfrequenz `F_g`, andererseits aus dem eigentlichen Observer mit dem Regelgewicht `kP_Iq`. Die Dämpfung des Tiefpassfilters beträgt -40dB pro Dekade.

Das Regelgewicht `kP_Iq` kann auch auf 0 gesetzt werden. Damit bleibt nur das Tiefpassfilter.

`F_g` Grenzfrequenz, bei schwingungsarmer Mechanik kann die Grenzfrequenz auf 600 Hz eingestellt werden. Bei einer Mechanik mit Spiel und Riemen muss die Grenzfrequenz evtl. bis auf 200 Hz reduziert werden.

Zu tiefe Werte für die Grenzfrequenz beeinträchtigen die Bandbreite des Systems und verschlechtern evtl. die Regelgüte.

`kP_Iq` P-Anteil für Observer Filter

Der P-Anteil für die Gewichtung des Beobachters muss empirisch bestimmt werden. Dazu mit einem sehr kleinen Wert beginnen: 0.0001.

Den Wert kontinuierlich erhöhen bis die Phasenreserve ansteigt. Das Optimum ist erreicht, wenn bei maximierter Phasenreserve noch genügend Amplituden-Reserve vorhanden ist.

Die Geschwindigkeits-Regelbandbreite wird erhöht ohne Verringerung der Phasen-Reserve.

Charakteristik des Tiefpass-Filters

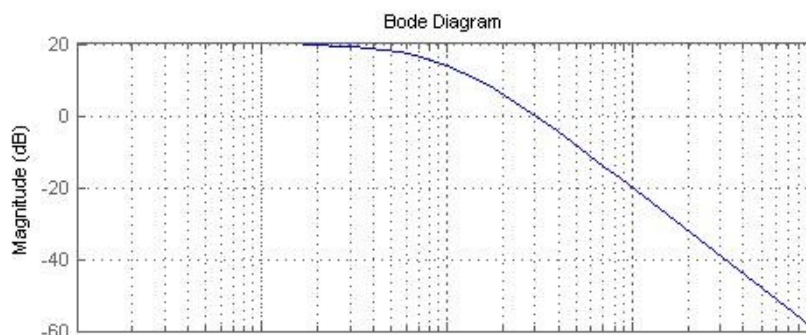


Abb 68: Tiefpassfilter 2. Ordnung

Das Observer-Filter ist dem Average-Filter aufgrund der besseren Charakteristik vorzuziehen.

11.8.2 Average Filter

kT_Speed

Der D-Anteil des PID-Reglers verursacht Lärm, der mit *kT_Speed* etwas verringert werden kann. Wenn eine grosse Masse vorhanden ist, kann *kT_Speed* bis ca 1/10 der Periodendauer T_k gesetzt werden. Immer so tief wie möglich, nur so hoch wie nötig.

Bei zwei überlagerten Schwingzeiten:

Wenn sich zwei Schwingzeiten zeigen (z.B. 6ms Riemen-Schwingzeit, 100ms Massen-Schwingzeit) kann mit dem **kd** (Beschleunigungs-Fehler bzw. D-Anteil von Geschwindigkeits-Regler) Abhilfe geschaffen werden.

Dazu wird der **kd** in sehr kleinen Schritten erhöht. Es muss ein Kompromiss gefunden werden zwischen möglichst kleinem Schwingverhalten und möglichst geringer Lärm-Entwicklung des Antriebes.

Wenn das **kd** verwendet wird, muss das Resolver-Filter *kT_Speed* auf 0.1 gesetzt werden.

Charakteristik des Average-Filters

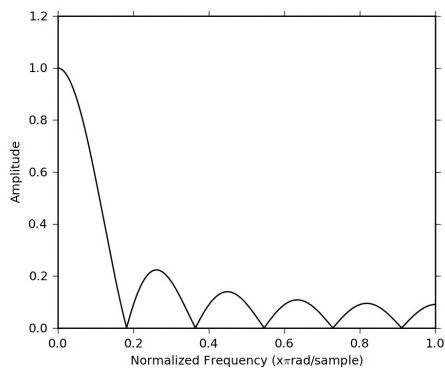


Abb 69: Average Filter

11.9 Gantries

Beim Sweepen von Doppel-Y Gantries sind folgende Punkte zu beachten:

- Die Ströme für sweepen so wählen, dass Motoren und Mechanik nicht überlastet werden.
- Die Achsen müssen entkoppelt sein für die Aufzeichnung der Bode-Sweeps
- X- und Y-Achsen jeweils in die Mitte platzieren und sweepen
- Während die eine Achse Y1 gesweept wird, muss die andere Achse Y2 im Simulations-Mode aktiviert und festgehalten werden.
- Nach dem ersten Sweep und der Optimierung der PID-Parameter /Filter müssen die gefundenen Parameter in beide Achsen Y1,2 geladen werden. Danach wiederum die Y2-Achse im Simulations-Mode aktivieren und mit den optimierten PID-Parameter/Filter festhalten. Den Sweep für Achse Y1 wiederholen.
- Um optimale Parameter zu finden die über die gesamte Arbeitsfläche gültig sind sollten mehrere Sweeps aufgezeichnet werden. Dazu eine Matrix mit Punkten im Abstand von 10cm festlegen. Mit dem Gantry alle Punkte anfahren und jeweils einen Sweep mit **beiden** Achsen X1 und X2 erstellen.

12 Schrittmotor ohne Feedback in Betrieb nehmen

2-phasige Stepper-Motoren werden mit je einem separaten Stromregler für jede Phase angesteuert:

Phase-U = Iq (Wirkstromregler)

Phase-V = Id (Blindstromregler)

Es gibt also 2 Stromregler, Phase U = Iq und Phase V = Id, die den sinus/cosinus Strömen folgen müssen. Daher geht hier nur ein einfacher P-Regler, das KI wird nicht benutzt und bleibt 0.

Einen Positions-Regler gibt es natürlich nicht, da kein FeedBack.

Das kP von PosCtrl wird benutzt um den Grundstrom ein zu stellen:

Istandby = Motor.In * PosCtrl.standby.kP

Iforward = Motor.In * PosCtrl.forward.kP

Ibackward = Motor.In * PosCtrl.backward.kP

Wenn kP = 1 wird der definierte I_nom Strom des Motors eingepreßt.

Die VorhalteWerte phvSpeed und phvAcc funktionieren wie immer

Das Feldbus-Feedback wird aus FieldAxisPos gemacht, also mit 4096Inc/turn

Wichtig

- Die Schrittmotoren haben ein sehr hohes Ke (24V Motoren ca 30-60V, am besten zB mit Akku-Bohrmaschine messen)
- Feld-Gesteuert laufen sie daher nur bis 400-600U/min. Darüber müsste eine Feldschwächung (ansteigendes -Id) eingeführt werden, ist noch nicht implementiert.
- Ohne FB gesteuert laufen sie höher, aber ab dieser Drehzahl fällt der Strom massiv zusammen, so dass bei 2000Umin statt 2A nur noch 200mA fließen und das Drehmoment somit abnimmt.
- Das Strom kP muss auf ca. 2-8 sein
- Rs, Ls, Ke ist noch für 3Ph_pp eingegeben -> umrechnen
- Ohne Last sind Resonanzen zu erwarten, Motor steht nur
- Mit Last ist je nach Drehzahl ein kleinerer Strom besser (Pos_kP=0.8, er wird ja sowieso meist nicht erreicht)
- Encoder mit zB nur 4x400=1600Inc/T reichen nicht für Feldsteuerung, da für 90Grad Motor-Feld nur 1600/50/4 = 8 Inc bleiben.
- unter 4096Inc/T sicher _woFB betreiben, Encoder an SAM, kleines PID_P in SAM

13 Firmware Update, Parameter Update

Nebst der Firmware benötigt der INFO-ACS Regler zwei Konfigurations-Files:

Regler-Firmware	System.s
Motordaten	File.cpf

Die Motordaten beinhalten die PID-Parametersätze, physikalische Konstanten wie ohmscher Widerstand, Induktivität des Motors, usw.

Firmware und Parameter-Updates können entweder aus dem Parametrierprogramm `INIX-Motion` oder mit dem Konsolenprogramm `ACSUpdate.exe` ins Flash-Prom des Reglers gebrannt werden.

Firmware und Motordaten können auch über ein Netzwerk auf Remote-Targets ausgeführt werden.

13.1.1 Updates von Parametern und Software

Voraussetzungen:

- INCO-Server muss in Betrieb sein
- 24V Speisung für SAC-Regler
- Target muss registriert sein
- Aktiver Link wenn nicht mit seriellem Target kommuniziert wird
- ACSUpdate.exe

Das Programm ACSUpdate.exe befindet sich im Verzeichnis `..\acs\bin`. ACSUpdate.exe ist ein DOS-Programm. Um es zu starten öffnen Sie eine DOS-Box. Aufbau des Kommandos:

13.1.2 Firmware oder Motor-Parameter ins Flash-Prom brennen

Die Files (Motor-Parameter und Firmware) werden dem Programm via Kommandozeile übergeben. ACSUpdate.exe hält sich an die File-Endungen, die Reihenfolge spielt dabei keine Rolle. ACS-Update prüft vor dem Brennen die Versionsnummer der bestehenden und der zu ladenden Software. Mit der Extension `[-a]` (always) wird die zu ladende Version in jedem Fall gebrannt; ohne Extension wird nur gebrannt wenn es sich um eine neuere Version handelt.

```
c:\ACSUpdate TargetName [*.*] [-a] [*.*cpf] [-a] [*.*chf] [-a]
```

Beispiele

Brenne die Reglersoftware `"system.s"` und die Motorparameter `"motor.cpf"` ins Flash-PROM sofern es sich um eine **neuere** Version handelt als sich im Flash-Prom befindet. Das Target ist Achse 0, registriert auf einem Power-PC-Master.

```
C:\ACSUpdate PPC\Axis0 system.s motor.cpf
```

Brenne nur die Regler-Firmware `"system.s"` in jedem Fall ins Flash-PROM auch wenn es sich dabei um eine ältere Version handelt. Das Target ist ein ACS-Regler, der über die serielle Schnittstelle angesprochen wird.

```
C:\ACSUpdate INFO-ACS a:\acs\update\system.s -a
```


13.1.3 Motorparameter in File speichern

Motorparameter werden mit der Extension [-s] (save) in ein File mit der vorgegebenen Endung [* .cpf] gespeichert:

```
c:\ACSUpdate TargetName [* .cpf] [-s]
```

[* .cpf] = Motor-Parameter

Beispiele

Speichere die Motorparameter in ein File. Das Target ist ein ACS-Regler an einer seriellen Schnittstelle und befindet sich in einem Netzwerk.

```
C:\ACSUpdate Remote_ACS motor.cpf -s
```

13.1.4 Parameter vom RAM ins Flash-Prom kopieren

ACSUpdate mit der Extension [-h] brennt die aktuellen Motor-Parameter vom RAM des Reglers ins Flash-Prom:

```
c:\ACSUpdate TargetName [-p]
```

[-p] = Motor-Parameter

Beispiel

Brenne die Motorparameter vom ACS-Regler an einem StandAlone Master ins Flash-PROM.

```
C:\ACSUpdate INFO-SAM/AXIS0 -p
```

13.1.5 Informationen

Die Extension [-i] vergleicht die Firmware-Version bzw. die Regler-Parameter Version im Flash-PROM des Reglers mit der zu ladenden Version und zeigt an, ob ein Update nötig ist. Das Update wird nicht ausgeführt!

```
c:\ACSUpdate TargetName [* .s] [-i] [* .cpf] [-i]
```

13.1.6 Flash-PROM Updates automatisieren

Um für eine Maschine oder Anlage mit mehreren Motoren, auch in einem Netzwerk verteilt, rationell Flash-PROM Updates brennen zu können, kann dem Programm ACSUpdate.exe ein Konfigurations-File übergeben werden. Das Konfig-File enthält alle Aufforderungen zum Brennen mehrerer Flash-PROMs. Die File-Endung muss [**.CFG*] heissen!

Die Extension [*-a*] kann auch dem Konfig-File übergeben werden.

```
C:\ACSUpdate [*.cfg] [-a]
```

Inhalt des Config-Files

```
; Lokaler PowerPC-Master, Achsen 0,1,2,3
PPC/AXIS0 system.s axis0.cpf
PPC/AXIS1 system.s axis1.cpf
PPC/AXIS2 system.s axis2.cpf
PPC/AXIS3 system.s axis3.cpf

; PPC-Master in Netzwerk, Achsen 0,1
Remote_PPC/AXIS0 system.s axis4.cpf
Remote_PPC/AXIS1 system.s axis5.cpf

; ACS-Regler an serieller Schnittstelle im Netzwerk
Remote_ACS system.s axis6.cpf
```

Beispiel

Führe obiges Config-File aus:

```
C:\ACSUpdate ACSConfig.cfg
```

13.1.7 Version und Hilfe

```
C:\ACSUpdate -V          zeigt die aktuelle Version der Relger-Software an.
C:\ACSUpdate -?         zeigt eine kurze Hilfe an.
```

13.1.8 Updates mit Laptop

Voraussetzungen:

- INCO-Server muss in Betrieb sein
- 24V Speisung für INFO-ACsR
- Serielles Target muss registriert sein
- Important Notes
- ACS-Show oder ACSUpdate.exe

Um mit einem Laptop Flash-PROM Updates in den Regler zu brennen, beachten Sie bitte die Kapitel:

```
"3.3 Laptop Installation"
"4.2 ACS-Show starten"
"4.8.1 Software und Parameter laden"
```

oder verwenden Sie das Konsolenprogramm "ACSUpdate.exe". Für ein Update benötigt der Regler INFO-ACsR lediglich eine 24V Speisung und die serielle Verbindung zum Laptop. Ein aktiver Link (d.h. "Trans.exe" ausgeführt) ist bei einer seriellen Verbindung nicht nötig.

Wichtige Hinweise

Folgende Reihenfolge ist beim Anschliessen eines Laptop-Computers an die serielle Schnittstelle des Reglers einzuhalten:

1. Netzeinspeisung von Laptop trennen, sodass er nur vom Akku mit Strom versorgt wird.
2. INFO-ACsR und Laptop mit entsprechendem seriellen Kabel verbinden.
3. Netzteil wieder einstecken.

Grund: Durch die galvanische Trennung des Transformators wird die Laptop-Speisung auf ein Potential von 110V angehoben (vorausgesetzt, der Laptop wird über das 230V-Netz gespiesen). Weil bei herkömmlichen D-SUB Steckern nicht gewährleistet werden kann, dass der Schirm vor den Signalleitungen kontaktiert, besteht die Gefahr, dass der Potentialausgleich über die Signalgroundleitung erfolgt. Das hat die Zerstörung des betreffenden SIO-Kanals zur Folge.

13.2 Notsystem

Falls beim Brennen der Motor-Parameter ein Fehler auftritt und das Flash-Prom zerstört wird, kann der Drive immer noch im Notsystem gestartet werden.

Um den Drive im Notsystem starten zu können, muss ein Kurzschlussstecker auf die serielle Schnittstelle (Frontplatte) gesteckt werden. Im Notsystem wird Flash-PROM brennen unterstützt.

Verbindungen:	Signale Pin	SAC2/3 D-SUB 9-pol.	SAC3x3 RJ-45
	RxD, TxD	2, 3	1, 2
	DSR, DTR	6, 4	3, 4

Nachdem der Regler aufgestartet ist, kann der Kurzschlussstecker entfernt und das serielle Kabel zum PC wieder eingesteckt werden.

14 Trouble Shooting

14.1 INFO-Link Probleme

Zuerst müssen Link Probleme beseitigt werden. Der Fehler Zähler darf nicht hochzählen. Siehe Lichtmengen Messgerät "INFO-Mess".

Zuerst müssen Link Probleme beseitigt werden. Der Fehler Zähler darf nicht hochzählen. Siehe Lichtmengen Messgerät "INFO-Mess".

14.2 Probleme mit analogen Gebern: SinCos, Resolver

Bei Problemen mit analogen Messsystemen finden Sie hier weitere Informationen:

Kapitel:

3.5 Resolver

3.6 SinCos

3.7 Sinus Cosinus / Resolver Pegel überprüfen

14.3 Verschmutzung

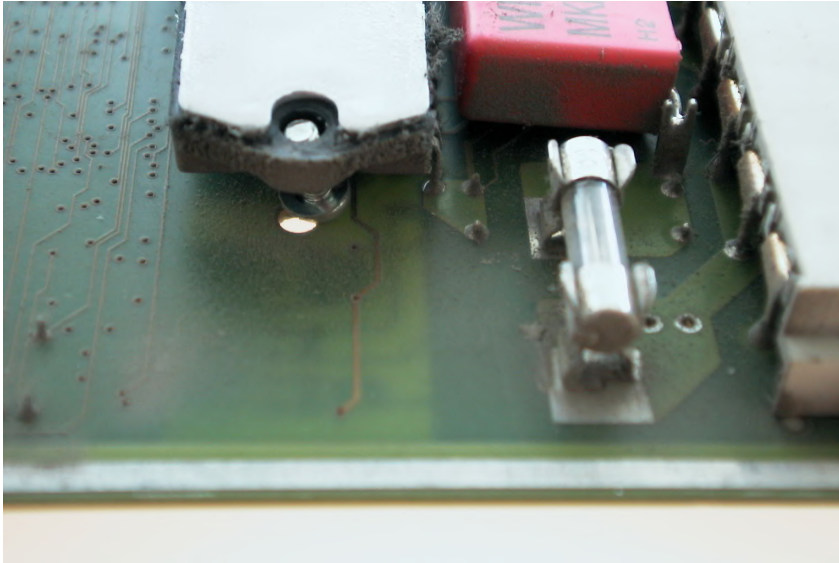


Abb 70: Verschmutzung

Ventilatoren müssen mit einem Staubfilter ausgerüstet werden. Schmutz und Feuchtigkeit können zu Kurzschlüssen auf der Karte führen!

14.4 Speisung

14.4.1 Zwischenkreis Spannung



Abb 71: Zwischenkreis

Der Strom (gelb) nimmt ab, obwohl der Positionserror zunimmt. Der Unterschied zwischen Soll- (grün) und Ist-Drehzahl (blau) nimmt ebenfalls zu.

Fehler: Die Zwischenkreisspannung ist zu klein.

Ursachen

- Zu viele Regler an einem Netzteil
- Motor überlastet
- Speise-Spannung (230VAC, 400VAC) zu klein

14.4.2 Spannungseinbrüche

Die Phasenerkennung kann ansprechen, wenn eine einzelne Phase zu stark einbricht. Um diesen Fehler zu beseitigen, kann das Flag No_PhaseFailure auf 1 gesetzt werden. Damit wird die Phasenerkennung ganz ausgeschaltet.

Pfad im Inco-Tree zum Resolver: `Ctrl.MotorConfig.Power.Supply.Flag`

14.4.3 Einspeisung MAX-Board

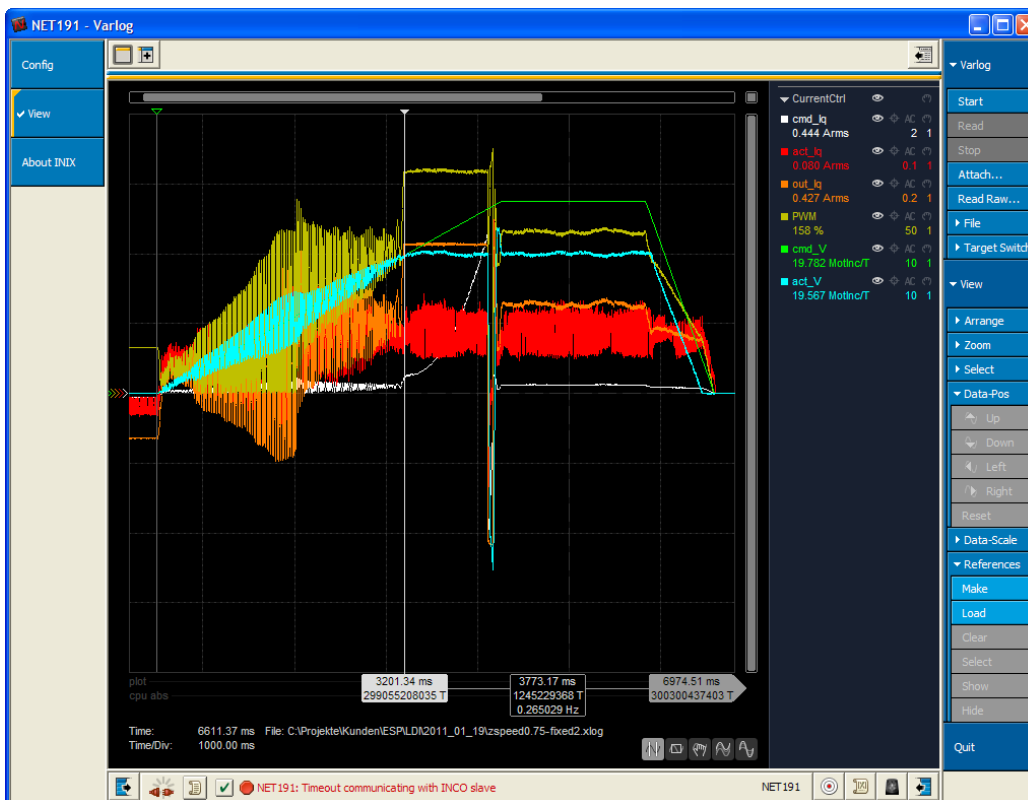


Abb 72: Einspeisung MAX-Board

Die Einspeisung der Motor-Versorgung (+V_MOT) vom MAX2/4 Board ist zu niedrig. Die minimale Speise-Spannung beträgt +15V, im obigen Beispiel ist die Speisung +12V.

14.5 Last

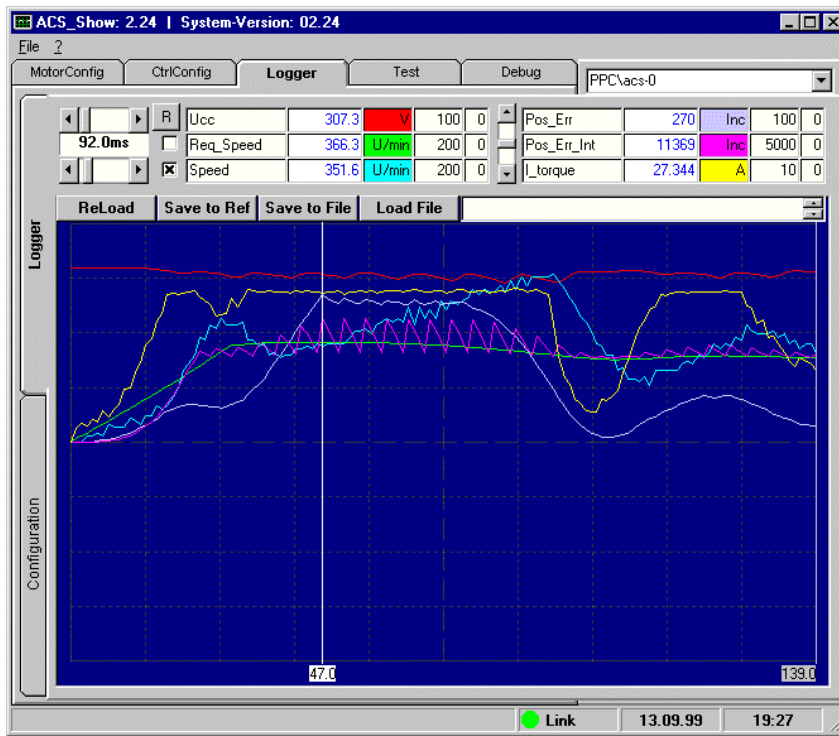


Abb 73: Last

Der maximale Strom ist erreicht. Die ganze Bahnfahrt wird mit maximalem Strom gefahren.

Fehler: Der Regler ist voll ausgelastet.

Bei der Cursor-Position ist der Positionier-Fehler ca. 270 Inkremente. Maximal zugelassen sind 250 Fehler-Inkremente. Die Ist-Geschwindigkeit hat sehr grosse Abweichungen zur Sollgeschwindigkeit. In diesem Zustand kann der Motor nicht mehr geregelt werden!
Die Regler-Software begrenzt den Positionier-Fehler auf 250 Inkremente.

14.6 PID-Parameter

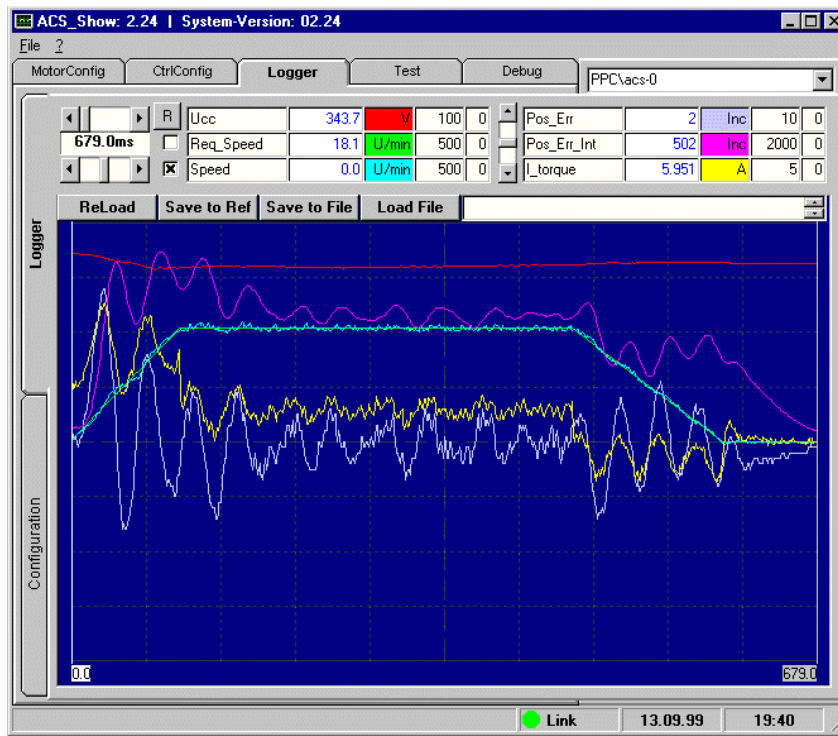


Abb 74: PID-Parameter

Die PID-Parameter für "Vorwärts" sind nicht optimal eingestellt.

14.7 Störungen

Störungen auf dem Resolver-Kabel. Die ± 3 Inkremente Positionierfehler können unmöglich von einem Wirkstrom (I_{Torque}) von weniger als 100mA erzeugt werden. Beachten Sie die Verdrahtungshinweise in Kapitel 2.3 "Verdrahtung" und in der Verdrahtungs-Richtlinie.

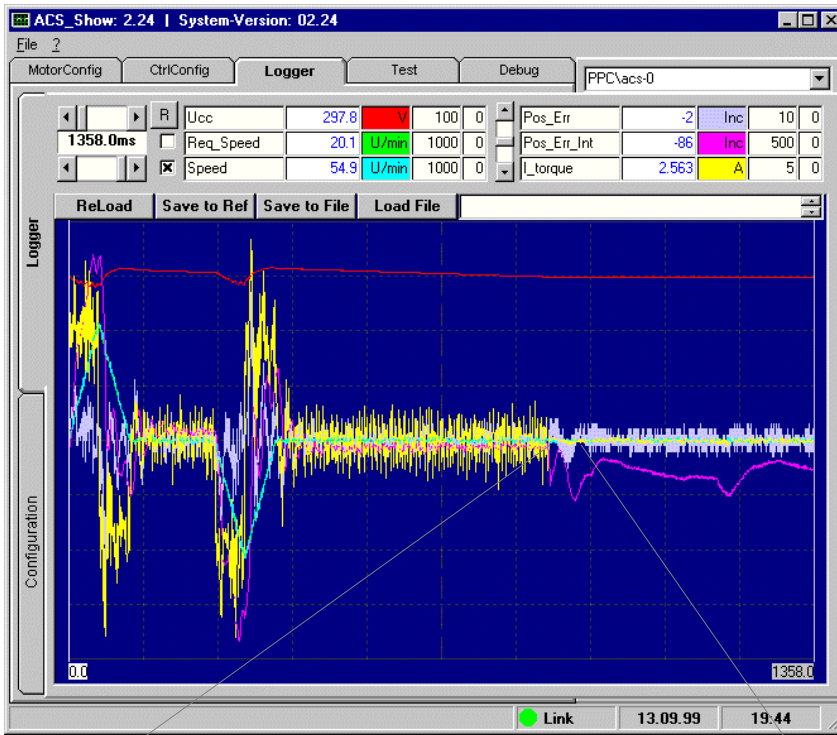


Abb 75: Störungen

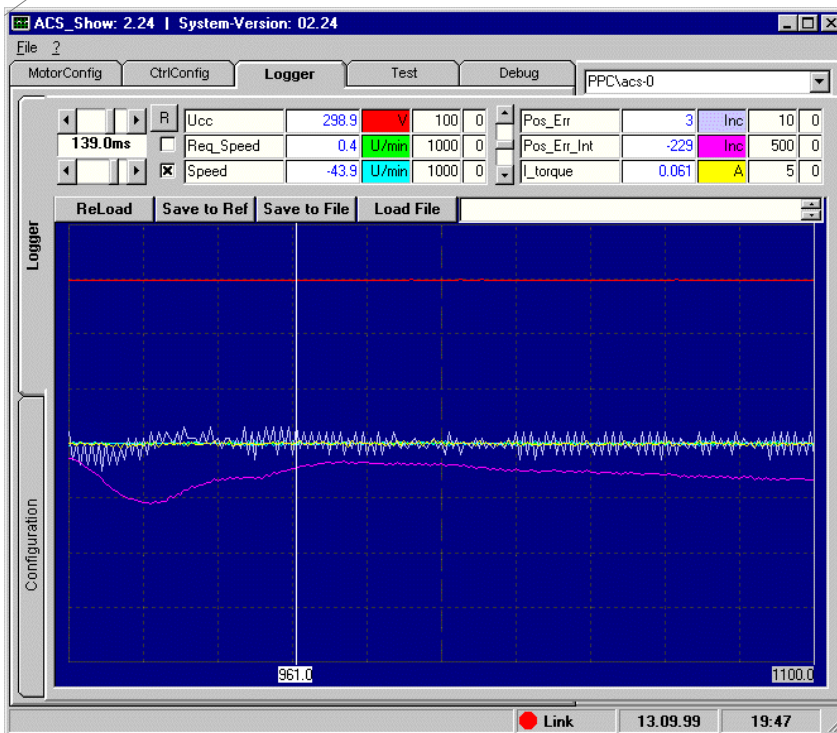


Abb 76: Störungen

14.8 Vorhaltewerte

Der Vorhaltewert `phvSpeed` ist massiv zu gross gewählt. Die Ist-Geschwindigkeit ist um 40U/min grösser als die Sollgeschwindigkeit. Die Regelung ist nicht mehr in der Lage, den zu grossen `phv` zu kompensieren. Der Integrator `Pos_Err_Int` (violett) und der `Pos_Err` (grau) geraten an ihr Maximum.

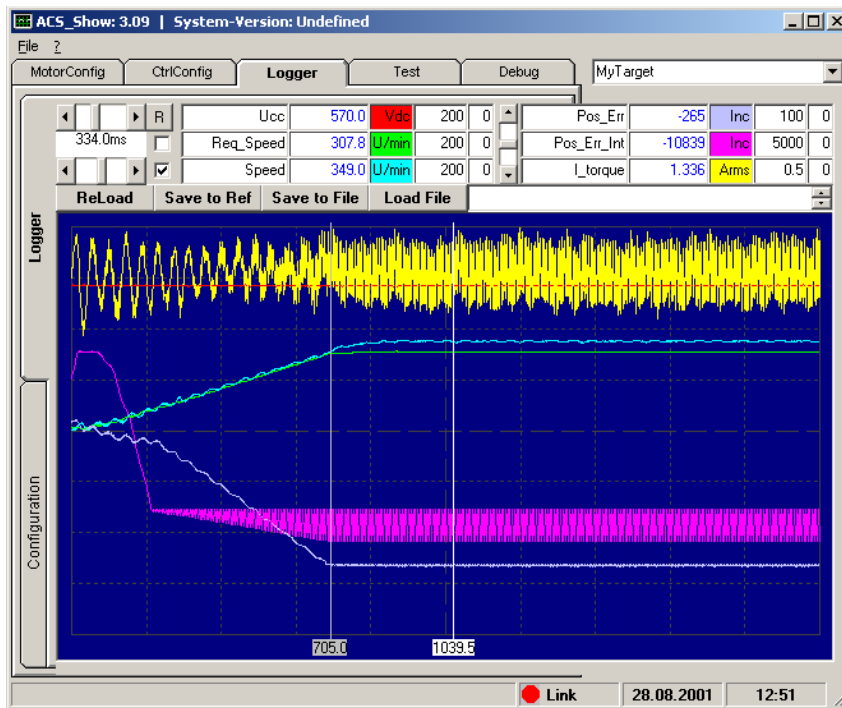


Abb 77: Vorhaltewerte

14.9 Normierungs Fehler

In der INFO-Link Konfiguration muss die Drehzahl bei 10V angegeben werden. Im Servo Regler muss ebenfalls angegeben werden, wieviel Drehzahl bei 10V gefahren wird. Stimmt diese Normierung nicht, ist das Regelverhalten nicht optimal. Unterscheidet sich die Normierung zu stark, kann die Achse nicht bewegt werden. D.h. sie geht auf Schleppfehler oder auf Überstrom.

Normierungsfehler erkennt man an der Kurve "Sollgeschwindigkeit" bzw. "Req_Speed" (grüne Kurve). Muss der Feldbusmaster die Sollgeschwindigkeit während der Fahrt anpassen, stimmt die Normierung nicht.



Abb 78: Normierungsfehler

14.10 Falsches Ke

Der Wert für das Ke ist zu gross eingestellt. Beim Bremsen erhält der Motor deshalb zuwenig Spannung und kann nicht richtig abbremesen.

Beachte Kapitel 7.12 "Feinabgleich von Ke, Rs und Ls"



Abb 79: Falsches Ke; Aufzeichnung mit Speed Waves

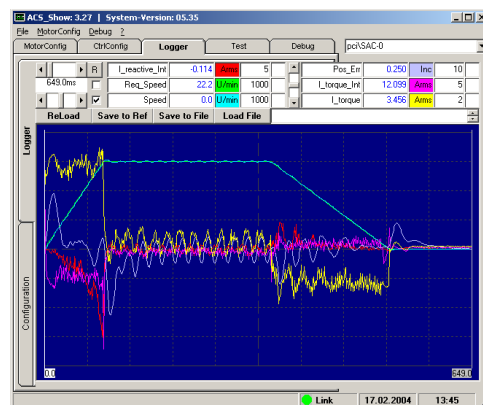
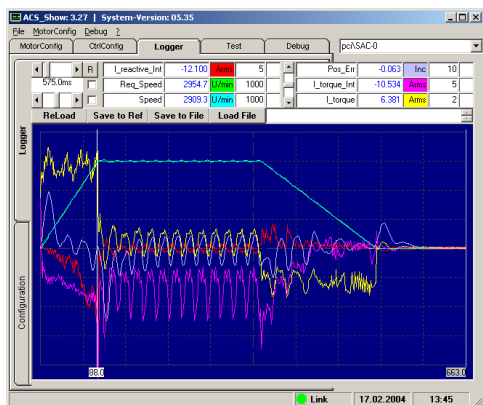


Abb 80: Falsches Ke; Aufzeichnung mit Wirkstrom- Blindstrom-Integral Bild links: falsches Ke, Bild rechts richtiges Ke

14.11 Falscher Resolver Offset

Der Resolver Offset ist falsch eingestellt. Dies äussert sich indem die Achse nicht mit voller Geschwindigkeit fahren kann. Beim falschem Resolver Offset wird die Blindspannung falsch berechnet, der Regler geht mit "Imax recieved" auf Fehler.

Beachte Kapitel 10.16.3 "Resolver Offset abgleichen"

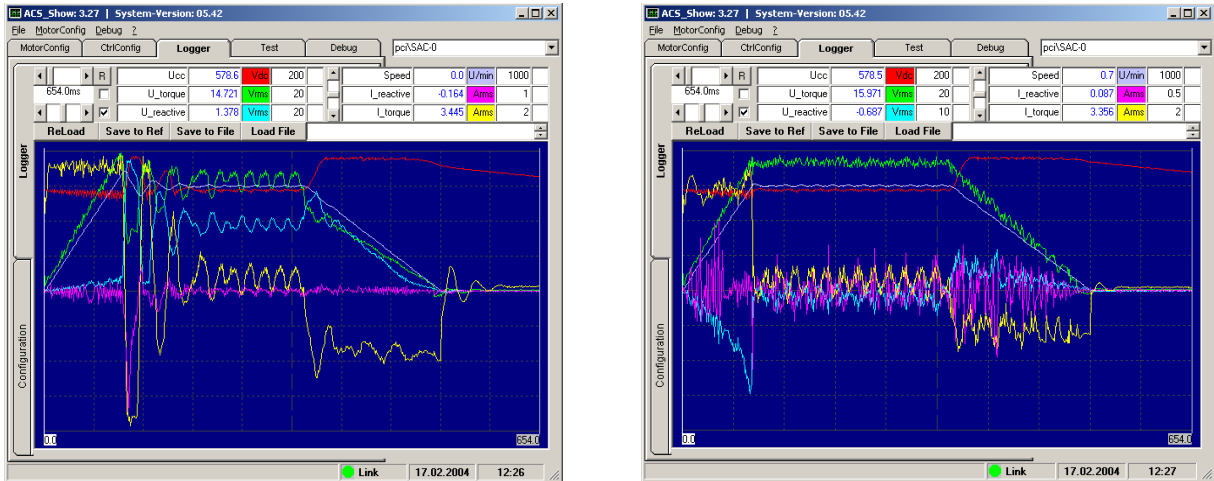


Abb 81: Falscher Resolver Offset; Aufzeichnung mit Voltage Waves Bild links: falscher , Bild rechts richtiger Resolver Offset

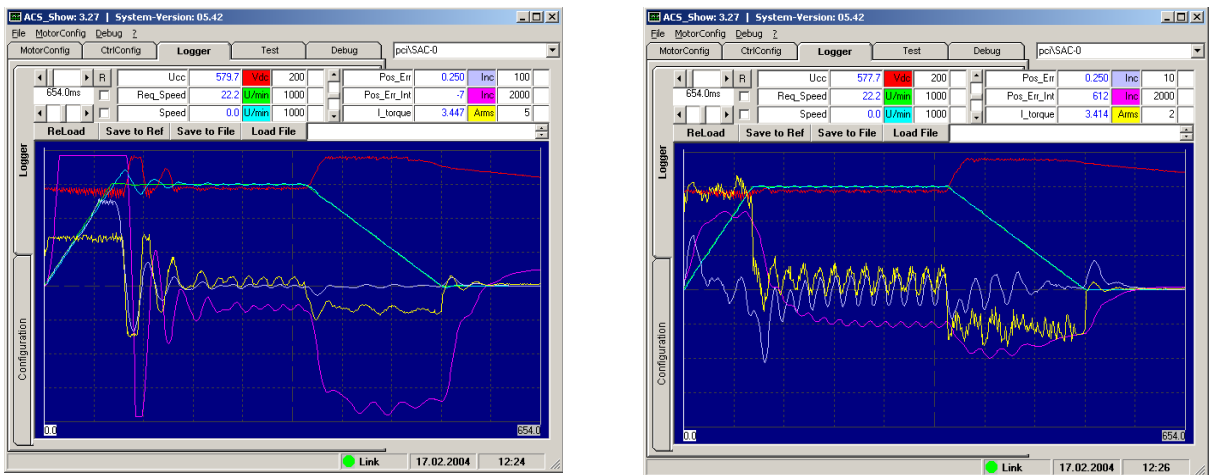


Abb 82: Falscher Resolver Offset; Aufzeichnung mit Speed Waves Bild links: falscher , Bild rechts richtiger Resolver Offset

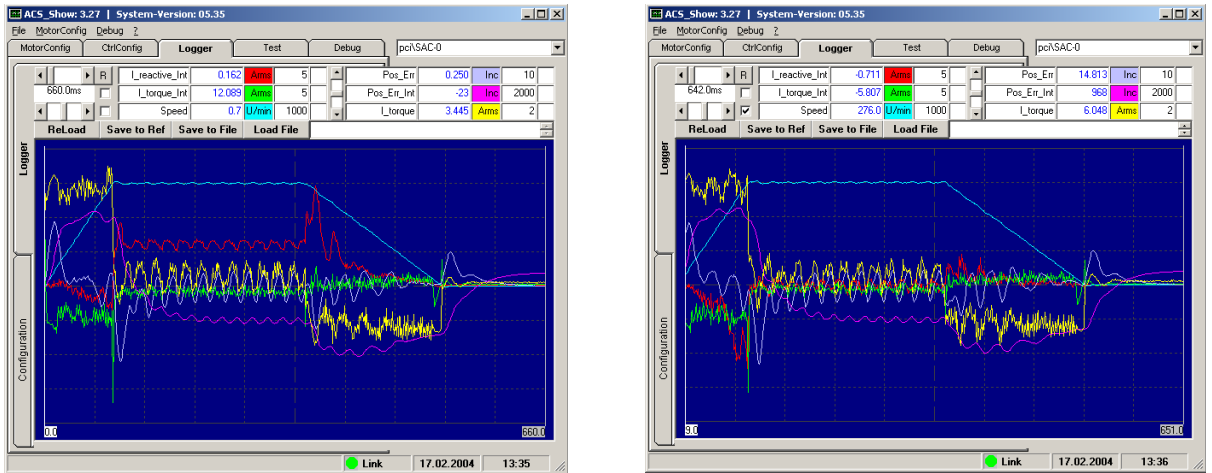


Abb 83: Falscher Resolver Offset; Aufzeichnung mit Speed Waves Bild links: falsches , Bild rechts richtiges Ke



Abb 84: Falscher Resolver Offset, Speed Waves

15 Weiterführende Dokumentationen

Drive-Inbetriebnahme-Manual.pdf
Indel-Safety-Manual.pdf
Hardware-Manual-Motion-Boards.pdf
Hardware-Manual-SAC3.pdf
Verdrahtungsrichtlinie.pdf
Aufbauanleitung.pdf

16 Abbildungsverzeichnis

Abb 1: INIX Motion.....	8
Abb 2: Variablen Logger.....	9
Abb 3: Firmware, Motor- Controller-Konfiguration laden.....	11
Abb 4: Motor-Config Parameter brennen, speichern.....	12
Abb 5: Motor-Konfig Parameter speichern und brennen.....	13
Abb 6: Test-Modi.....	15
Abb 7: INIX Motion.....	16
Abb 8: Firmware Version im Motion-Tool	17
Abb 9: Firmware Version im Inco-Explorer.....	17
Abb 10: Absolut Encoder Konfiguration.....	19
Abb 11: Ist-Werte Absolut Encoder.....	20
Abb 12: Fehler Absolut Encoder Endat.....	21
Abb 13: Fehler Absolut Encoder Hiperface.....	22
Abb 14: Fehler Absolut SSI.....	23
Abb 15: Encoder.....	24
Abb 16: Ist-Werte Encoder.....	24
Abb 17: Resolver	25
Abb 18: Ist-Werte Resolver	26
Abb 19: SinCos	26
Abb 20: Ist-Werte SinCos	27
Abb 21: SinCos	28
Abb 22: Auto Kommutierung UVW pulse.....	30
Abb 23: Auto Kommutierung Two-Phase Stepper.....	31
Abb 24: Flag not_Unwind nicht gesetzt.....	33
Abb 25: Flag not_Unwind gesetzt.....	33
Abb 26: Hall-Sensor Kommutierung.....	34
Abb 27: Beschaltung mit nicht invertierten Eingängen.....	35
Abb 28: Beschaltung mit invertierten Eingängen.....	35
Abb 29: Hallsensor Sequenz, Normdrehrichtung CCW.....	37
Abb 30: Hallsensor Sequenz, Normdrehrichtung CW.....	37
Abb 31: Ist-Werte Auto-Kommutierung UVW.....	39
Abb 32: Ist-Werte 360° Kommutierung.....	39
Abb 33: Stromregler.....	40
Abb 34: Extern Enable.....	43
Abb 35: Ist-Werte Enxtern Enable.....	45
Abb 36: Motor Field Feedback.....	46
Abb 37: Position Control.....	47
Abb 38: GinLink.....	48
Abb 39: Motor.....	50
Abb 40: Trapezregler.....	55
Abb 41: Positions Regler.....	56
Abb 42: Power Supply.....	58
Abb 43: Average Filter.....	59
Abb 44: Observer.....	59
Abb 45: Hardware Istwerte.....	60
Abb 46: Beispiel: STO, verriegelte Schutztüre.....	65
Abb 47: GinLink Konfiguration	66
Abb 48: Move Kommandos	70
Abb 49: ACS-Show	73

Abb 50: Ist-Werte Resolver	83
Abb 51: Position der Achse	84
Abb 52: Feld-Mode.....	86
Abb 53: Stromregler berechnen.....	88
Abb 54: Kritische Verstärkung kP krit.....	98
Abb 55: Kritische Verstärkung kP krit zu hoch.....	98
Abb 56: Erste Version der PID-Parameter.....	99
Abb 57: kP wurde erhöht bis Strom und Schleppfehler wieder schwingen.....	100
Abb 58: Wirkung des kP-Anteils.....	110
Abb 59: Wirkung des kI-Anteils.....	110
Abb 60: Wirkung des kD-Anteils.....	111
Abb 61: Wirkung des kd-Anteils.....	111
Abb 62: Low-Pass Filter -6dB Verstärkung.....	113
Abb 63: Low-Pass Filter 6dB Verstärkung.....	113
Abb 64: Beispiel: Low Pass Filter mit 6dB Verstärkung bei 210Hz.....	113
Abb 65: Notch-Filter -6dB Dämpfung.....	114
Abb 66: Notch-Filter 6dB Verstärkung.....	114
Abb 67: Two-Load Filter Güte=3.....	114
Abb 68: Tiefpassfilter 2. Ordnung.....	115
Abb 69: Average Filter.....	116
Abb 70: Verschmutzung	124
Abb 71: Zwischenkreis	125
Abb 72: Einspeisung MAX-Board.....	126
Abb 73: Last	127
Abb 74: PID-Parameter	128
Abb 75: Störungen	129
Abb 76: Störungen	129
Abb 77: Vorhaltewerte	130
Abb 78: Normierungsfehler	131
Abb 79: Falsches Ke; Aufzeichnung mit Speed Waves	132
Abb 80: Falsches Ke; Aufzeichnung mit Wirkstrom- Blindstrom-Integral Bild links: falsches Ke, Bild rechts richtiges Ke	132
Abb 81: Falscher Resolver Offset; Aufzeichnung mit Voltage Waves Bild links: falscher , Bild rechts richtiger Resolver Offset.....	133
Abb 82: Falscher Resolver Offset; Aufzeichnung mit Speed Waves Bild links: falscher , Bild rechts richtiger Resolver Offset.....	133
Abb 83: Falscher Resolver Offset; Aufzeichnung mit Speed Waves Bild links: falsches , Bild rechts richtiges Ke	134
Abb 84: Falscher Resolver Offset, Speed Waves	134

17 Dokumentenstatus

Disclaimer

Für die gemachten Angaben besteht keine Gewähr für Richtigkeit oder Vollständigkeit. Technische Änderungen vorbehalten.

File-History

1.20	29.04.2011	Änderungen im Motor-Config File für GinLink Achsen, Block-Kommutierung für Maxon-Motoren Formel für KTY-Temperatursensoren, Ext_En Flag Short Circuit
1.21	09.05.2011	Block-Kommutierung für Maxon-Motoren erweitert, I2t erweitert
1.22	10.05.2011	Hardware Istwerte
1.23	20.05.2011	i2t-Regelung max Motor-Temp.
1.24	21.07.2011	Abschalt-Sequenz mit STO und Bremsrampe
1.25	10.09.2011	Neue Bilder
1.26	26.09.2011	Ablauf für STO
1.27	27.09.2011	Diverse kleine Anpassungen
1.28	11.10.2011	Umschaltung PID Parameter für Positions Regler
1.29	27.04.2012	Brennen der System Software mit ZIP- File (GinLink), Kapitel Dokumentenstatus Hinzugefügt Standartbelegung Feedback Kanäle auf GinLink
1.30	27.04.2012	Konfiguration Inkrementalgeber an SinCos Interface Beschreibung dlq_dt_corr Flag
1.31	27.04.2012	DeadTime Konfiguration in IMD hinzugefügt, IBN Schrittmotor erweitert
1.32	20.06.2012	Enable Bits für GinLink deklariert
1.33	21.09.2012	Kapitel 10.11.2.1, Auto-Kommutierung mit Hiperface hinzugefügt
1.34	17.04.2013	Generelle Erweiterung des Kapitel 10. Neu mit Querverweisen zu tiefgründigeren Informationen in anderen Kapiteln. Kann nun als Step by Stap Anleitung verwendet werden. Beschreibung des asym_acc Flags. Kapitel 3.15 erweitert. Reglerfrequenz ist nun im MotorenConfig File Konfigurierbar. Kapitel 5.2 mit Beschreibung des Motor.dt2 Files erweitert. Neues Kapitel 1.2:Übersicht des Dokuments Beschreibung I2t erweitert, neue Aufteilung in die Parameter I2t_up_run und I2t_up_halt. Kapitel 11.4 erweitert. Phasengewinn im unteren Frequenzbereich mittels Tiefpass mit pos. Gain
1.35	22.04.2013	Div. Kleinigkeiten angepasst. Div. Rechtschreibfehler korrigiert.
1.36	26.04.2013	Kapitel 3.14.1Motor Konfiguration angepasst Kapitel 3.16.1 Beschreibung des Asym_acc Flag korrigiert
1.37	19.08.2015	Neuer Parameter MasPosErr_kl in Kapitel 3.16.1 Neues Kapitel 10.16.4 Spezialfall Z-Achse mit 360 deg FieldRotation
1.38	27.07.2020	BMX: Schreibfehler „Feher“ und „akitive“ korrigiert.