



Leitfaden

zur Optimierung der Parameter
des feldorientierten **Servoregler-IC TMC4671**

M.Sc.
Selim Maurice Jendoubi
Field Application Engineer
Motion Control

MEV Elektronik Service GmbH | Nordel 5a | D - 49176 Hilter a.T.W.
Tel.: +49 (0) 5424 / 2340-0 | info@mev-elektronik.com

www.mev-elektronik.com

Leitfaden zur Optimierung der Parameter des feldorientierten Servoregler-IC TMC4671



Inhaltsverzeichnis

1.	Leitfaden zur Parameteroptimierung	3
1.1	Wizard	3
1.1.1	Allgemeine Einstellungen	5
1.1.2	Open Loop-Modus.....	7
1.1.3	ADC Kanäle	9
1.1.4	Hall-Signale	13
1.1.5	ABN-Encoder Konfiguration.....	16
1.1.6	Analog-Encoder Konfiguration	21
1.1.7	Zusammenfassung	21
1.2	Parameteroptimierung	23
1.2.1	Limit-Einstellungen	24
1.2.2	Parameteroptimierung des Stromreglers (Open Loop)	26
1.2.3	Parameteroptimierung des Stromreglers (Closed Loop)	30
1.2.4	Analyse der Bode-Diagramme	35
1.2.5	Parameteroptimierung des Geschwindigkeitsreglers	37
1.2.6	Parameteroptimierung des Positionsreglers	40
	Referenzen	44

1. Leitfaden zur Parameteroptimierung

In diesem Kapitel wird der Leitfaden zur Erstinbetriebnahme der Motoren und anschließenden Parameteroptimierung erstellt. Zu Beginn wird dabei auf den *Wizard* eingegangen.

1.1 Wizard

In dem *Wizard* erfolgt durch einzelne Schritte die Anfangskonfiguration des jeweiligen Motors. Die Durchführung der Konfiguration ist zu Beginn zwingend erforderlich, sodass im Anschluss daran die Reglerparameter optimiert werden können. Um den *Wizard* zu starten, wird zunächst das Evaluationskit via USB mit einem PC verbunden und ein Netzteil an die Endstufen-Platine angeschlossen.

Sollte die TMC4671-EVAL-Platine nicht direkt in der TMCL-IDE erkannt werden, ist ein manuelles Auswählen erforderlich. Dies erfolgt durch einen Klick auf die Landungsbrücke im Aufklappmenü. Daraufhin öffnet sich ein neues Fenster, in dem die TMC4671-EVAL-Platine ausgewählt werden kann (siehe *Abbildung 1*).

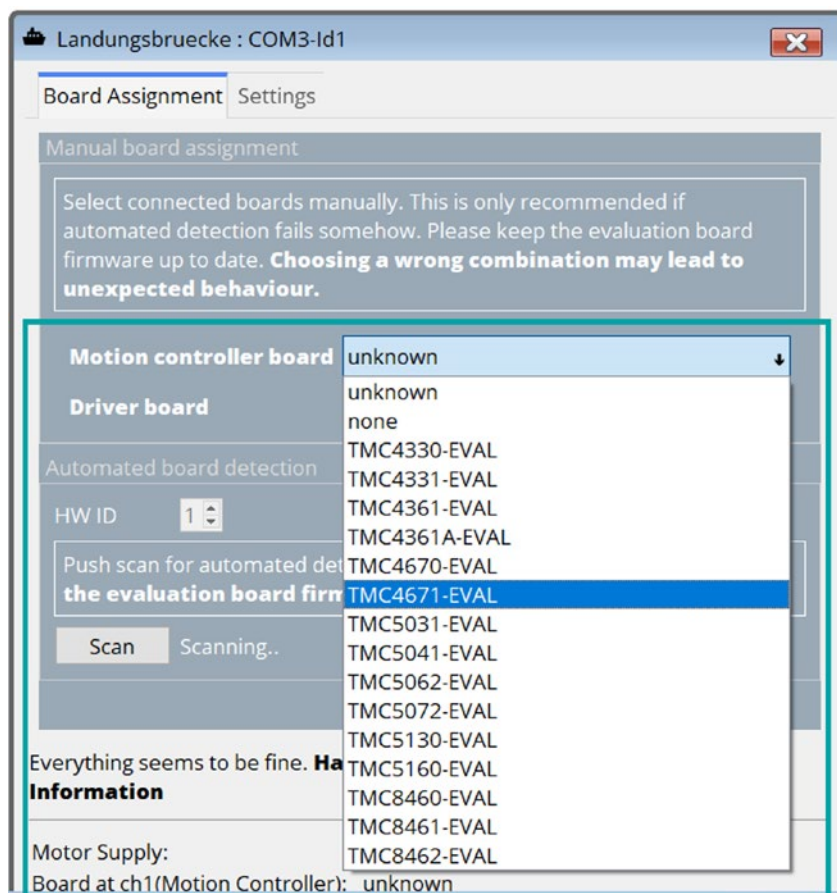


Abbildung 1: Auswählen der TMC4671-EVAL-Platine

Nachdem die TMC4671-EVAL-Platine von der TMCL-IDE erkannt wurde, weist eine bunte Linie auf den in der rechten oberen Ecke befindlichen Zauberstab (Abbildung 2).

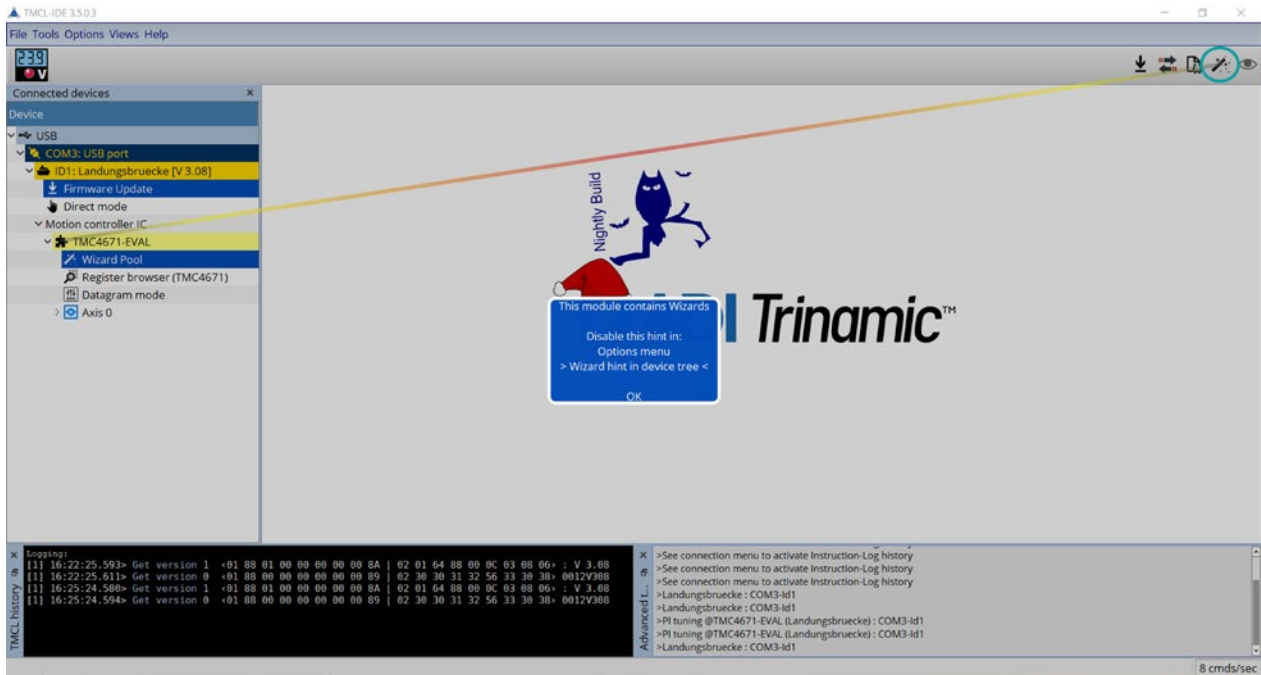


Abbildung 2: Verweis auf den Wizard

Durch Anwählen des Zauberstabs gelangt man, wie in Abbildung 3 zu sehen, zu einem neuen Fenster, in dem der so genannte *Weasel configurator* ausgewählt werden muss.

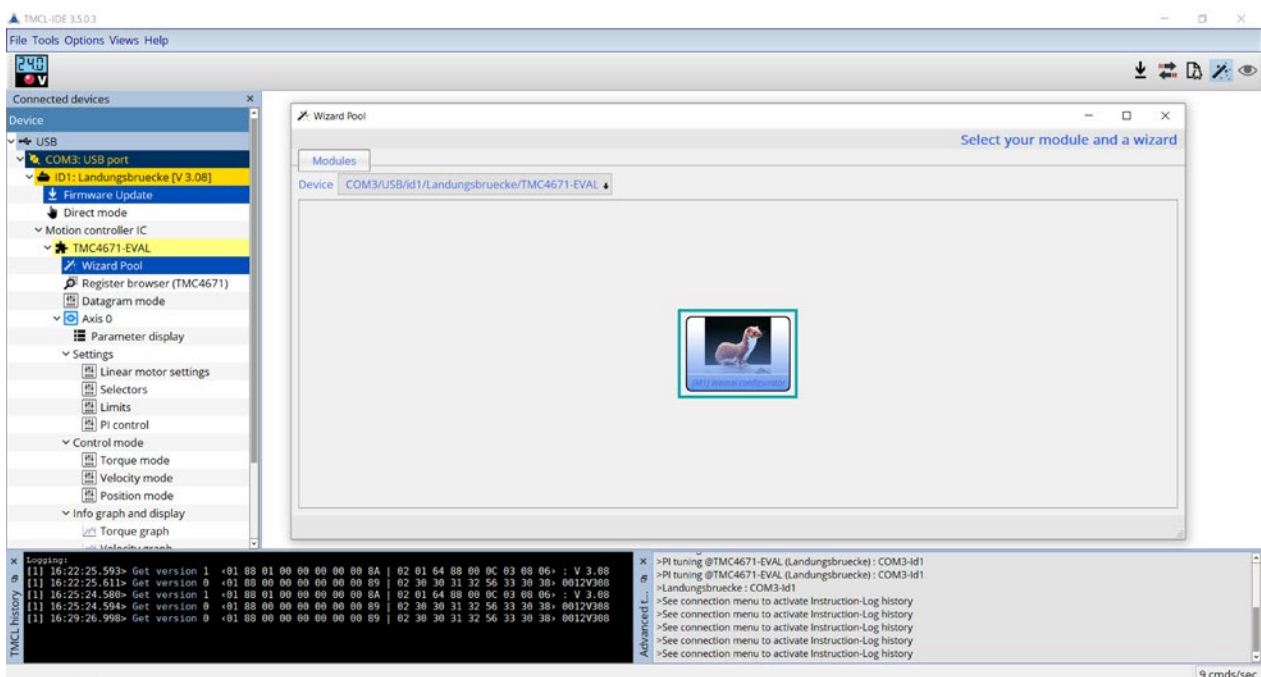


Abbildung 3: Auswählen des Weasel configurators, um zu den Konfigurationsschritten zu gelangen

Anschließend öffnet sich eine Einführungsseite, auf der alle Konfigurationsschritte aufgelistet sind (Abbildung 4). Hierbei muss bezüglich der digitalen Hall-Sensoren, des ABN- oder Analog-Encoders, abhängig von dem verwendeten Feedback-System, nur bei den Systemen ein Haken gesetzt werden, welche verwendet werden. Zur Vollständigkeit werden an dieser Stelle alle Konfigurationsschritte durchlaufen. Mit Auswahl auf den Pfeil rechts unten gelangt man zum nächsten Schritt.

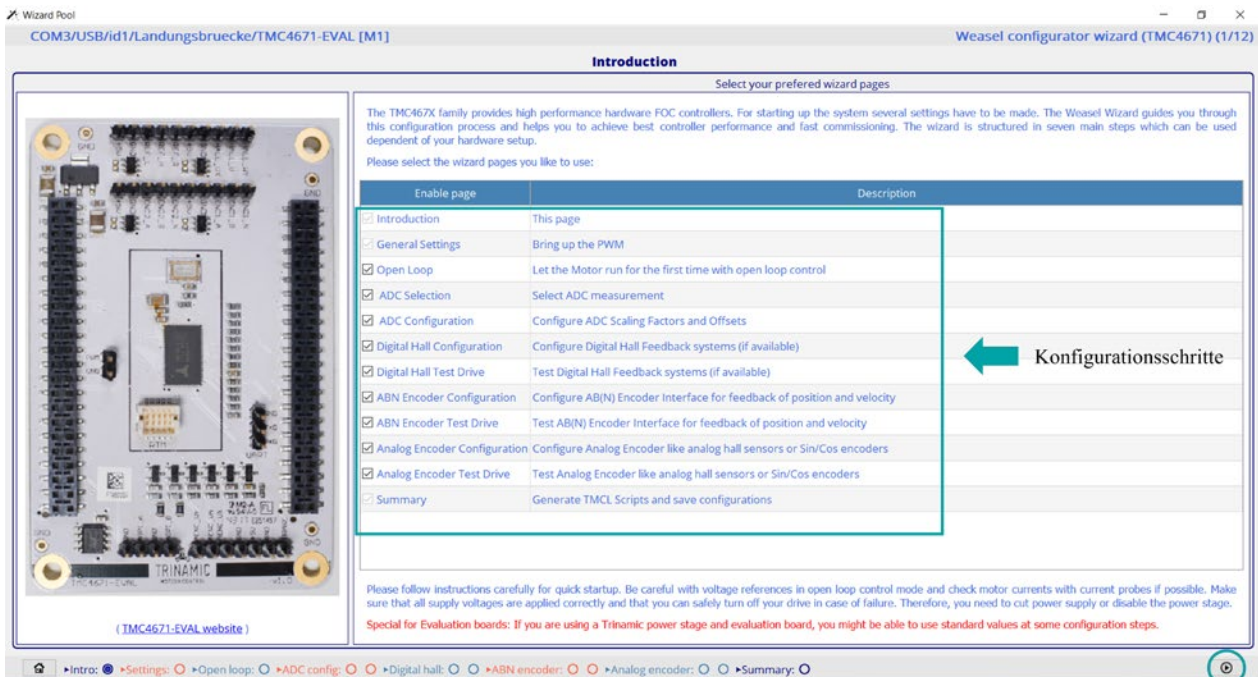


Abbildung 4: Einführungsseite des Konfigurators mit Darstellung der einzelnen Konfigurationsschritte

1.1.1 Allgemeine Einstellungen

In diesem Schritt werden allgemeine Einstellungen bezüglich des verwendeten Motors vorgenommen (Abbildung 5). Die Konfiguration erfolgt zu Beginn anhand eines 3-phasigen BLDC-Motors des Herstellers MOONS (AM42BL60L2), welcher über Hall-Sensoren verfügt. In den allgemeinen Einstellungen ist für die sechs Punkte folgendes durchzuführen:

- 1. Endstufen-Platine:** Zunächst wird die verwendete Endstufen-Platine ausgewählt. Desweiteren können über die so genannten *set defaults* Schaltflächen des entsprechenden Motors, die standardmäßigen Werte für die Register gesetzt werden. In diesem Fall wurde die TMC-UPS-10A70V-EVAL-Endstufen-Platine ausgewählt.
- 2. Motortyp:** Es erfolgt die Auswahl des angeschlossenen Motortyps, was durch das Aktivieren der *set defaults* Schaltfläche in Punkt 1 automatisch festgelegt wird.
- 3. Polpaarzahl:** Die Polpaarzahl des Motors ist in das freie Feld einzutragen. Wichtig ist hierbei, dass der eingetragene Wert durch die Eingabetaste bestätigt wird, sodass das Feld nicht mehr gelb hinterlegt ist. Somit wurde die Eingabe korrekt übernommen. Der hierbei verwendete BLDC-Motor verfügt über 3 Polpaare.

4. **PWM-Frequenz:** An dieser Stelle wird die PWM-Frequenz gewählt. Über das Register 0x18 kann diese sehr fein eingestellt werden. Es empfiehlt sich eine der drei voreingestellten Werte zu nutzen, da sich diese in der Praxis als sinnvoll erwiesen haben. In diesem Fall wurden 25 kHz gewählt.
5. **Break before make Zeiten:** Auswählen der Zeiten für die sogenannte break before make Schaltung. Die Zeiten gelten als Pause zwischen dem Umschalten von zwei Schaltern beziehungsweise MOSFETs. Somit wird verhindert, dass es bei zwei Spannungsbereichen zu einem Kurzschluss kommt. Die Zeiten sind hierbei auf 100 ns eingestellt und für die angeschlossene Endstufen-Platine ausreichend.
6. **PWM-Modus:** Dieser wird durch die am Anfang gewählte *set defaults* Schaltfläche automatisch auf *centered PWM*, den Modus für die FOC, eingestellt.

Die Seite kann anschließend über den Pfeil verlassen werden.

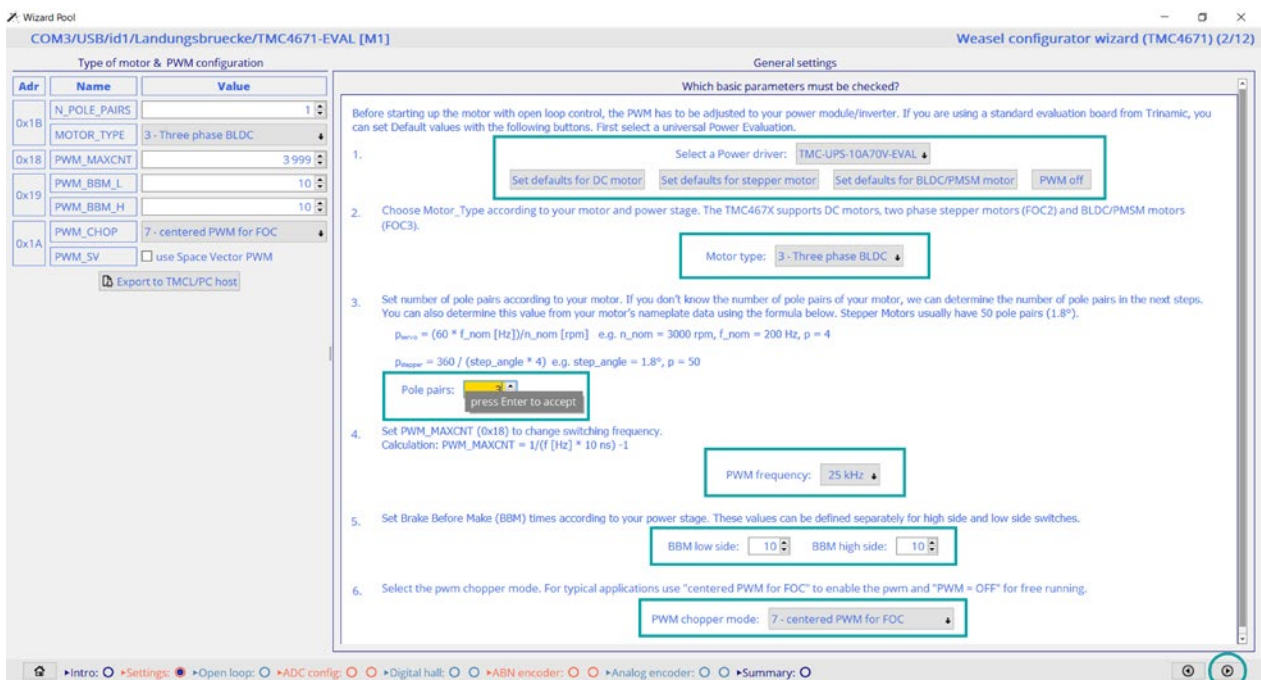


Abbildung 5: Erste Einstellungen bezüglich des Motors

1.1.2 Open Loop-Modus

Bei diesem Konfigurationsschritt wird der so genannte Open Loop-Regelungsmodus dargestellt (*Abbildung 6*). Dieser Regelungsmodus ermöglicht ein erstes Drehen des Motors bei entkoppelter oder sehr niedriger Last. Durch eine weitere *set defaults* Schaltfläche wird der Open Loop-Modus aktiviert und die entsprechenden Werte für die Register auf der linken Seite gesetzt. Dabei wird in Register 0x52 zunächst ein Winkelsignal mit der Bezeichnung *phi_e_openloop* generiert. Bei den Kommutierungswinkeln wird im weiteren Verlauf der Arbeit zwischen den Bezeichnungen *phi_e* und *phi_m* unterschieden.

Phi_e steht dabei für einen elektrischen und *phi_m* für einen mechanischen Winkel. Der elektrische Winkel *phi_e* wird sowohl für eine korrekte Kommutierung als auch für die Stromregelung benötigt. Der mechanische Winkel *phi_m* dagegen ist für die Geschwindigkeits- und Positionsregelung erforderlich. Des Weiteren wurde in Register 0x63 der Modus *uq_ud_ext* ausgewählt, anhand dessen durch den Schieberegler ein Spannungswert vorgegeben werden kann. Der Wert wird in Register 0x24 angezeigt, welches über einen Maximalwert von 32.767 verfügt.

Die Umrechnung des Spannungswerts auf die SI-Einheit Volt soll an folgendem Beispiel verdeutlicht werden: Der Maximalwert von 32.767 entspricht der maximalen Versorgungsspannung des Netzteils, welche in diesem Fall bei 24V liegt. Ein beispielhafter Registerwert von 4.000 entspricht bezogen auf den maximalen Registerwert einem Prozentsatz *p* von:

$$p = \frac{4000}{32.767} \times 100 = 12,21\% \quad (1.1)$$

Die Spannung *U* lässt sich, bezogen auf die maximale Versorgungsspannung, wie folgt berechnen:

$$U = \frac{12,21\% \times 24V}{100} = 2,93V \quad (1.2)$$

In diesem Fall würde aufgerundet eine Spannung $U \approx 3V$ am Motor anliegen. Zudem werden in Register 0x20 eine Beschleunigung und in Register 0x21 eine Zielgeschwindigkeit vorgegeben.

In dem Eingabefeld *OPENLOOP_VELOCITY_TARGET* kann die Geschwindigkeit auch manuell verändert oder durch Anwählen der Pfeile eine Änderung der Drehrichtung des Motors veranlasst werden. Durch langsames Erhöhen der Spannung durch den Schieberegler beginnt der Motor mit der Drehung. Hierbei ist es wichtig, dass die Geschwindigkeit bei Schrittmotoren, für eine gute Darstellung der Motorströme, die im nächsten Konfigurationsschritt veranschaulicht werden, sehr klein gewählt werden sollte. In diesem Fall empfiehlt es sich 1 rpm als Zielgeschwindigkeit festzulegen.

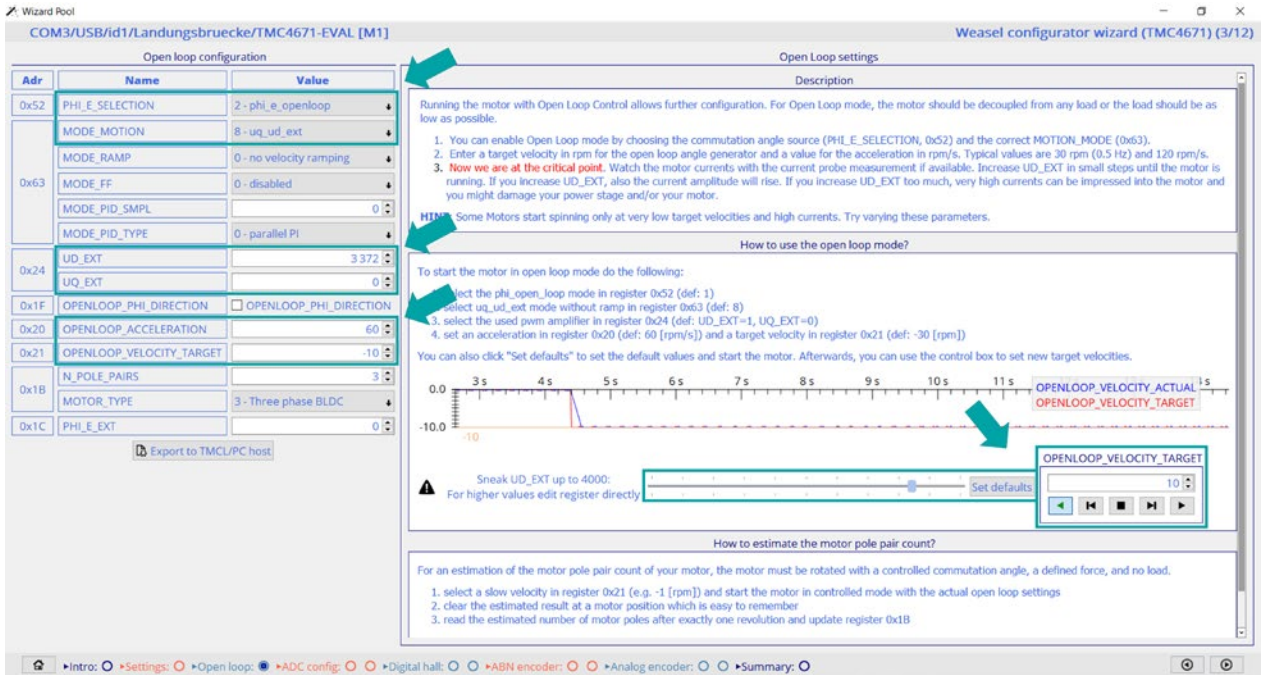


Abbildung 6: Open Loop-Regelungsmodus

Zum Abschluss des Open Loop-Modus gibt es zusätzlich die Möglichkeit die Polpaarzahl des verwendeten Motors schätzen zu lassen, falls diese nicht bekannt sein sollte. Ein Zähler zeigt dabei nach einer Umdrehung des Motors die geschätzte Polpaarzahl an (siehe Abbildung 7). Damit kann der Open Loop-Modus verlassen und zum nächsten Konfigurationsschritt gewechselt werden.

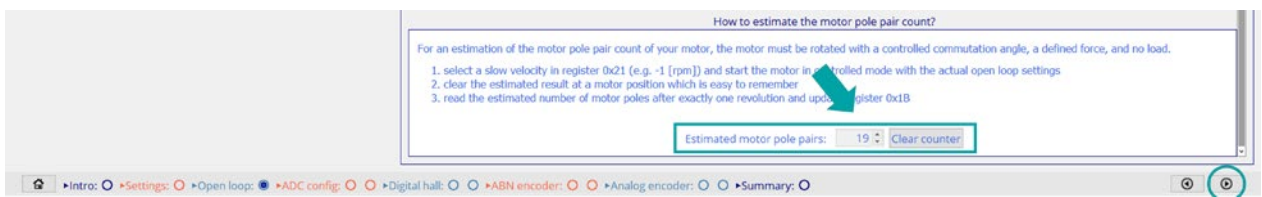


Abbildung 7: Schätzung der Polpaarzahl

1.1.3 ADC Kanäle

Der nachfolgende Schritt dient als Überprüfung und zeigt die Verläufe der ADC Signale zur Messung der Motorströme auf. Diese sollten wie in Abbildung 8 dargestellt einen sinusförmigen Verlauf aufweisen.

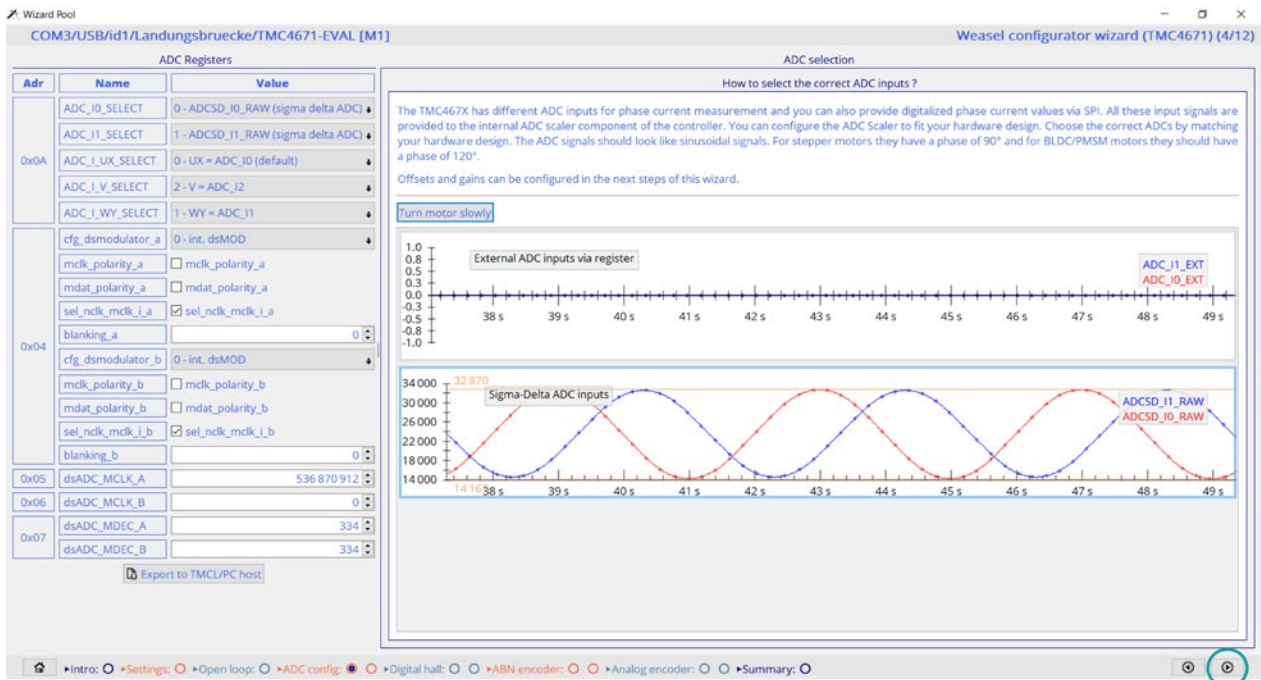


Abbildung 8: Verläufe der ADC Signale

Im nächsten Schritt wird die Kalibrierung der ADC Kanäle durchgeführt. In Abbildung 9 sind in den beiden linken Graphen die Verläufe der ADC Strommesskanäle *ADC_I0_RAW* und *ADC_I1_RAW* dargestellt. Für die FOC mit drei Motorphasenströmen wird noch ein weiterer Strommesskanal namens *ADC_I2* benötigt. Hierfür wird jedoch kein Verlauf gemessen, sondern folgende Berechnung auf der Grundlage der anderen beiden Kanäle durchgeführt:

$$\text{ADC_I2} = - (\text{ADC_I1} + \text{ADC_I0}). \quad (1.3)$$

Die Graphen auf der rechten Seite zeigen die Verläufe an, die von der *ADC Einheit* an die *FOC Einheit* weitergeleitet werden. Aus Abbildung 9 wird ersichtlich, dass die Verläufe auf der rechten Seite über einen Offset verfügen. Dieser muss durch eine Kalibrierung korrigiert werden. Die Kalibrierung wird durchgeführt, indem die Software den minimalen und maximalen Wert beider gemessenen Sinuskurven misst und daraus den Mittelwert errechnet. Dieser wird daraufhin als Zahlenwert in dem Feld neben den Graphen angezeigt. Anschließend muss der Wert über die Schaltfläche *Set* bestätigt werden.

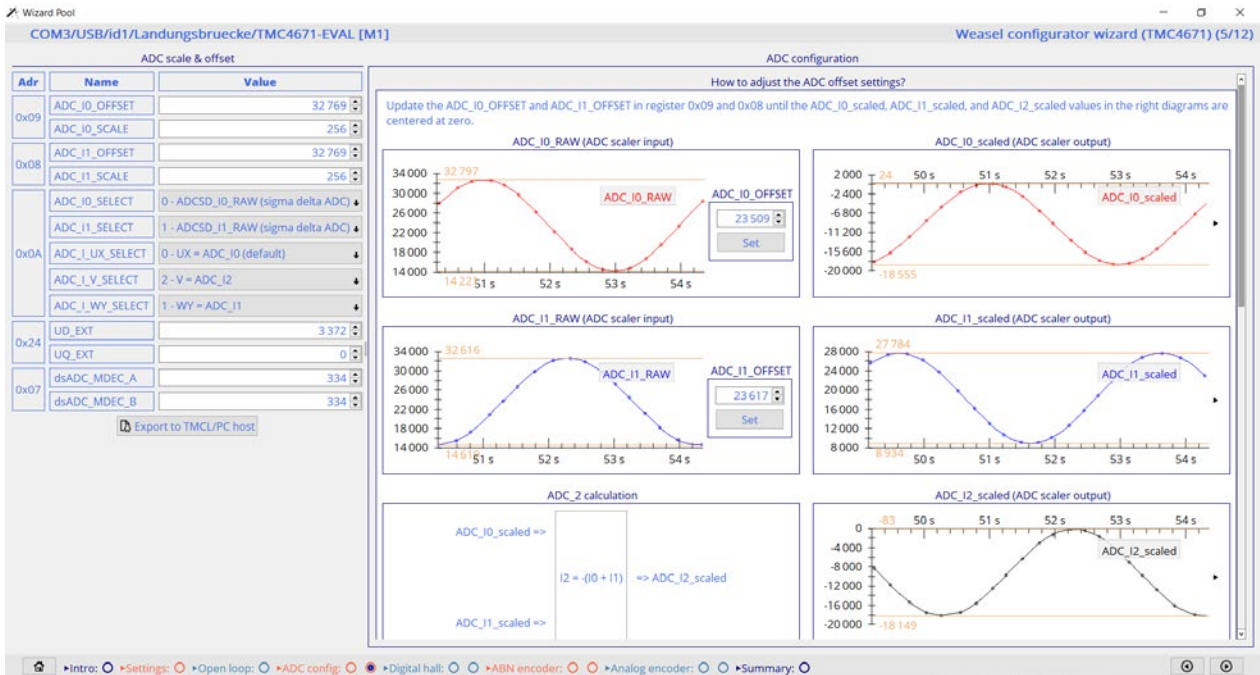


Abbildung 9: Verläufe der ADC-Kanäle mit Offset

Durch das Bestätigen wird der Offset korrigiert und die drei Verläufe auf der rechten Seite können für die FOC verwendet werden (siehe Abbildung 10).

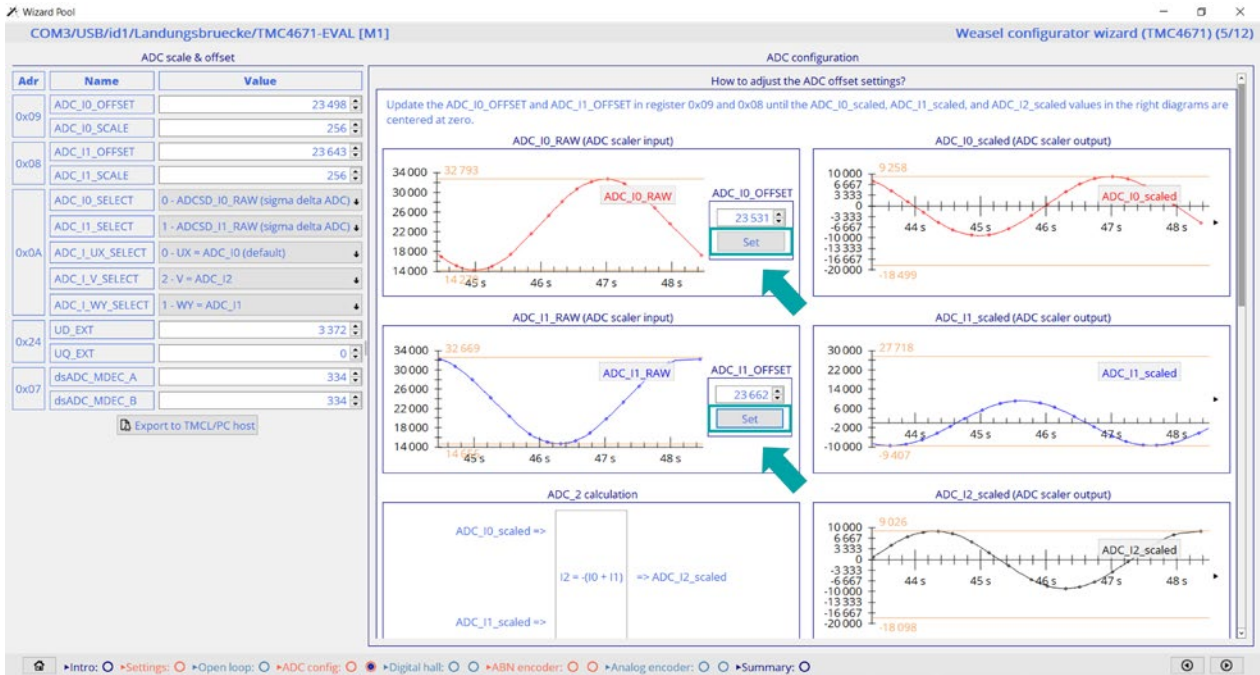


Abbildung 10: Verläufe der ADC-Kanäle ohne Offset

Eine weitere Einstellung, die bezüglich der ADC Strommesskanäle vorgenommen werden muss, ist die korrekte Zuordnung der Kanäle zu den PWM Ausgangssignalen für die Ansteuerung der Endstufen-Platine. Am Ende der Seite befindet sich eine Anzeige mit den Verläufen der Phasenströme und Phasenspannungen.

In Abbildung 11 ist zu erkennen, dass in Kanal V in der Mitte Strom und Spannung gleichphasig verlaufen. Die weiteren beiden Kanäle müssen allerdings noch korrekt eingestellt werden, da an dieser Stelle durch die Endstufen-Platine die falschen Kanäle ausgewählt wurden. Dazu sind in Register 0x0A die Kanäle korrekt zuzuordnen. In diesem Fall sind dies die Kanäle ADC_I1 und ADC_I0, welche ausgetauscht werden müssen.

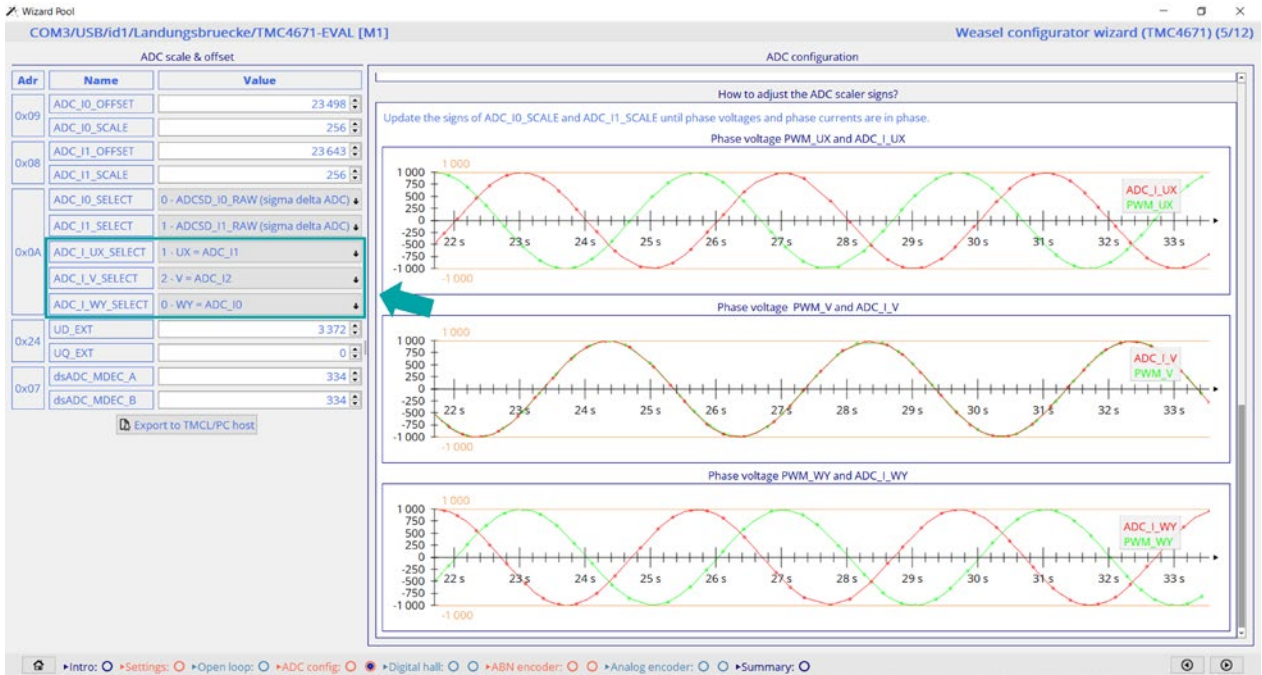


Abbildung 11: Phasenverschobene Kanäle U und W

Nachdem die Kanäle richtig zugeordnet wurden, verlaufen Strom und Spannung bezüglich der drei Kanäle gleichphasig, womit dieser Konfigurationsschritt abgeschlossen ist (Abbildung 12).

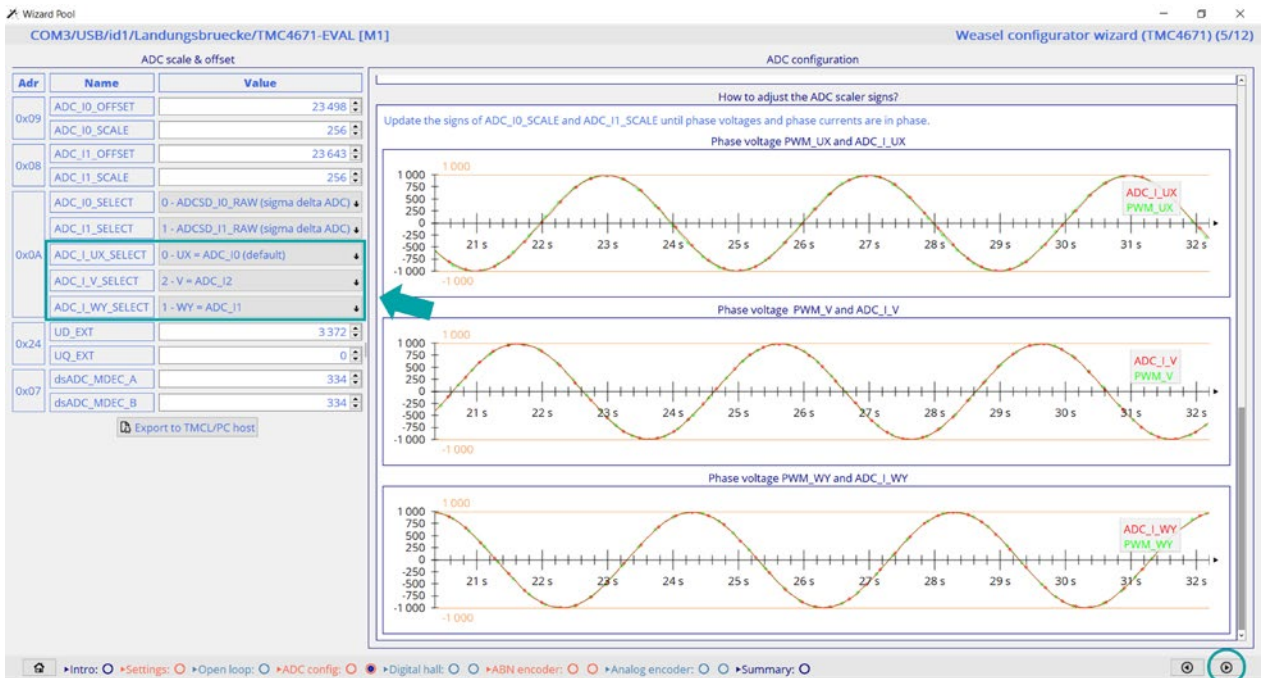


Abbildung 12: Strom und Spannung verlaufen gleichphasig

1.1.4 Hall-Signale

Der darauffolgende Konfigurationsschritt bezieht sich auf die Hall-Signale. Die blauen Linien in Abbildung 13 stellen eine durch den Open Loop-Modus generierte Position dar. Die roten Linien entsprechen den Positionen der Hall-Sensoren. Dabei werden sechs verschiedene Positionen während einer elektrischen Umdrehung angezeigt.

Verfügt ein Motor über ein Polpaar, entspricht eine mechanische Umdrehung einer elektrischen Umdrehung. Bei einem Motor mit n Polpaaren, entspricht eine mechanische Umdrehung n elektrischen Umdrehungen.

Für eine korrekte Konfiguration müssen die blauen und roten Linien in die **gleiche** Richtung zeigen und **aufeinanderliegen**. In diesem Beispiel aus Abbildung 13 weisen beide Linien dieselbe Richtung auf, allerdings muss die Polarität geändert werden, sodass sie auch übereinanderliegen. Dazu muss *polarity* im Bereich *Value* aktiviert werden. Zusätzlich kann im Register 0x37, falls notwendig, manuell ein Offset eingegeben werden, um einen möglichen Versatz auszugleichen. Sollte die Richtung der Linien unterschiedlich sein muss ebenfalls *direction* ausgewählt werden.

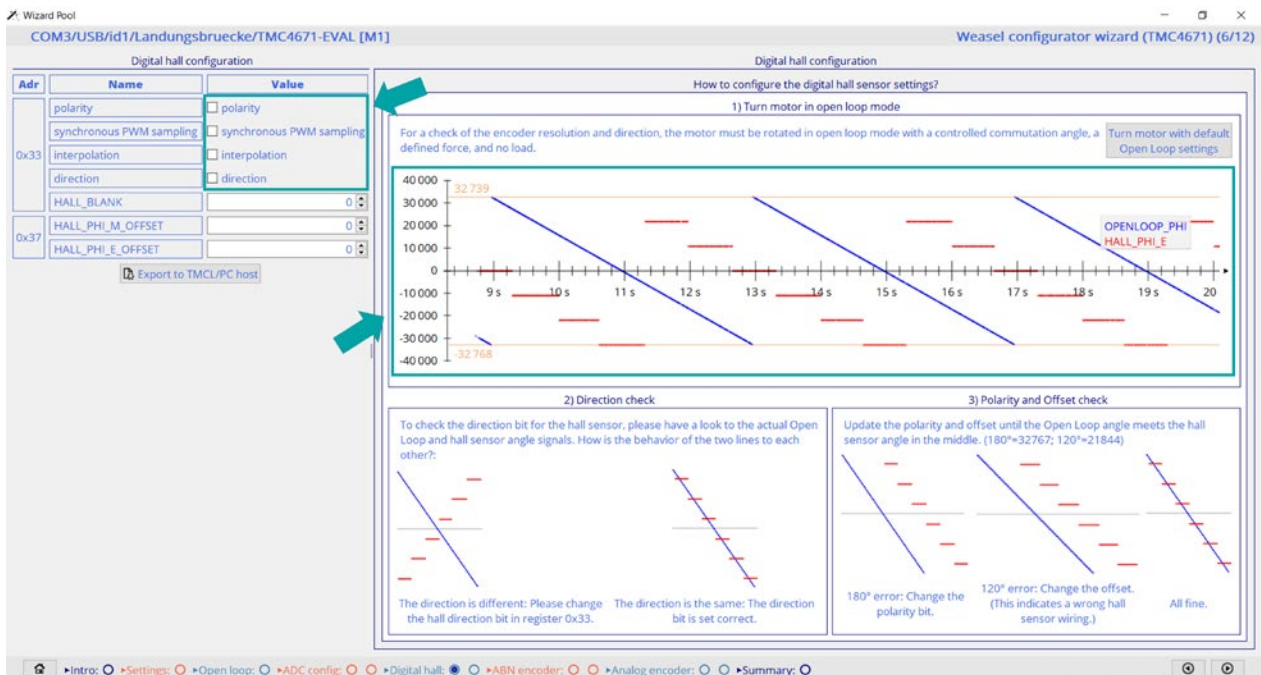


Abbildung 13: Beide Linien weisen dieselbe Richtung auf, liegen jedoch nicht übereinander

Der TMC4671 verfügt über die Möglichkeit einer Interpolation, um die Auflösung der Hall-Sensoren zu verbessern. Um diese Funktion zu nutzen, muss *interpolation* ausgewählt werden. Dadurch wird diese ab einer Geschwindigkeit von über 60 rpm aktiviert und Zwischenpositionen ermittelt, die eine höhere Auflösung der Hall-Sensoren bewirken. Die Interpolation eignet sich jedoch nicht für Positionieraufgaben sowie Richtungswechsel während des Betriebs.

Nach der Korrektur der Polarität liegen die Linien übereinander und dieser Konfigurationsschritt ist beendet (Abbildung 14).

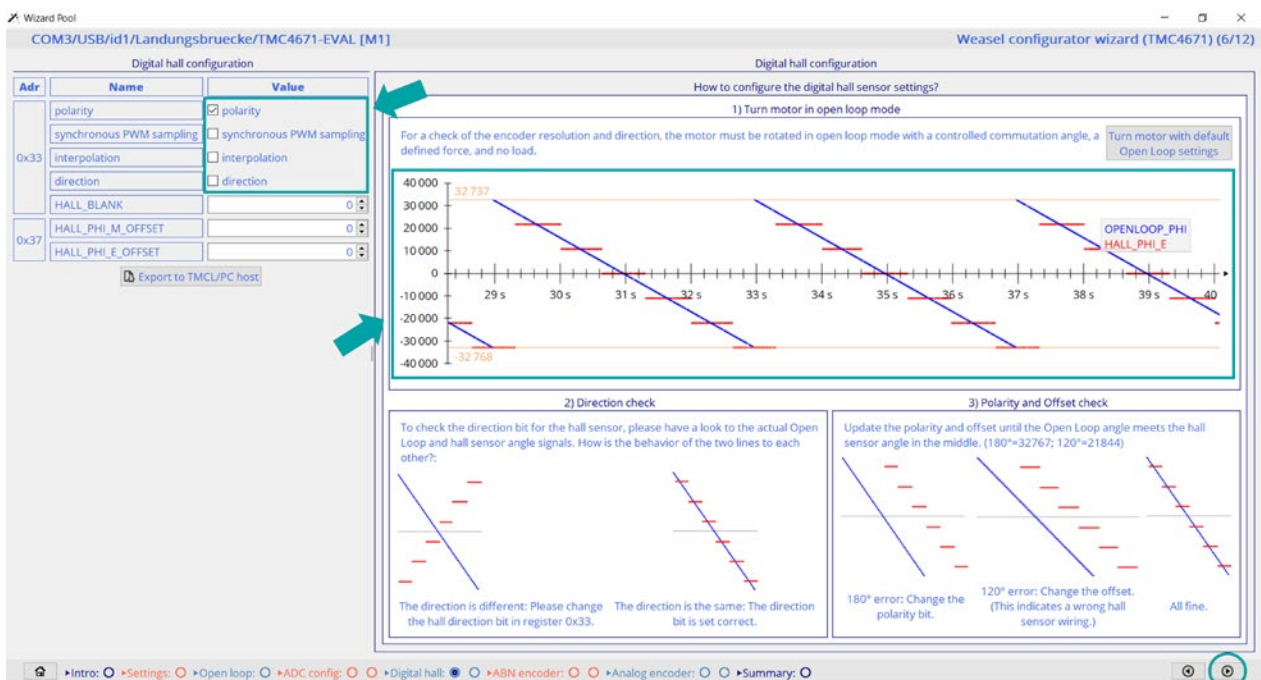


Abbildung 14: Beide Linien weisen dieselbe Richtung auf und liegen übereinander

Anschließend werden die eingestellten digitalen Hall-Signale im sogenannten Closed Loop-Modus getestet. Im Closed Loop-Modus wird im Vergleich zum Open Loop-Modus ein geschlossener Regelkreis aufgebaut und erste Reglerparameter durch das System gesetzt. Durch Anwählen der in Abbildung 15 sichtbaren *Set defaults and start* Schaltfläche wird in Register 0x63 der Stromregelungsmodus *torque mode* aktiviert. Des Weiteren wird bezüglich der Kommutierung in Register 0x52 das Winkelsignal der Hall-Sensoren ausgewählt. Die Register 0x54 und 0x56 zeigen die gesetzten Reglerparameter auf. Zudem wird in Register 0x5E eine Strombegrenzung und in Register 0x64 ein Zielwert für den Strom festgelegt. Dieser findet sich auch in dem Eingabefeld neben der *Set defaults and start* Schaltfläche wieder. Unterhalb des Eingabefelds kann zudem mittels der Pfeile die Drehrichtung des Motors verändert werden (PID_TORQUE_TARGET). Im oberen Graphen ist anhand der roten Linie der Sollwert des Stroms dargestellt, die dunkelblaue Linie zeigt dagegen den Istwert an. Der Sollwert des magnetischen Flusses ist dagegen anhand orangener Punkte auf der Nulllinie dargestellt, der Istwert durch die türkise Linie.

Durch Auswählen des Winkelsignals *phi_m_hal* in Register 0x50, wird im unteren Graphen anhand der roten Linie die mechanische Drehzahl angezeigt.

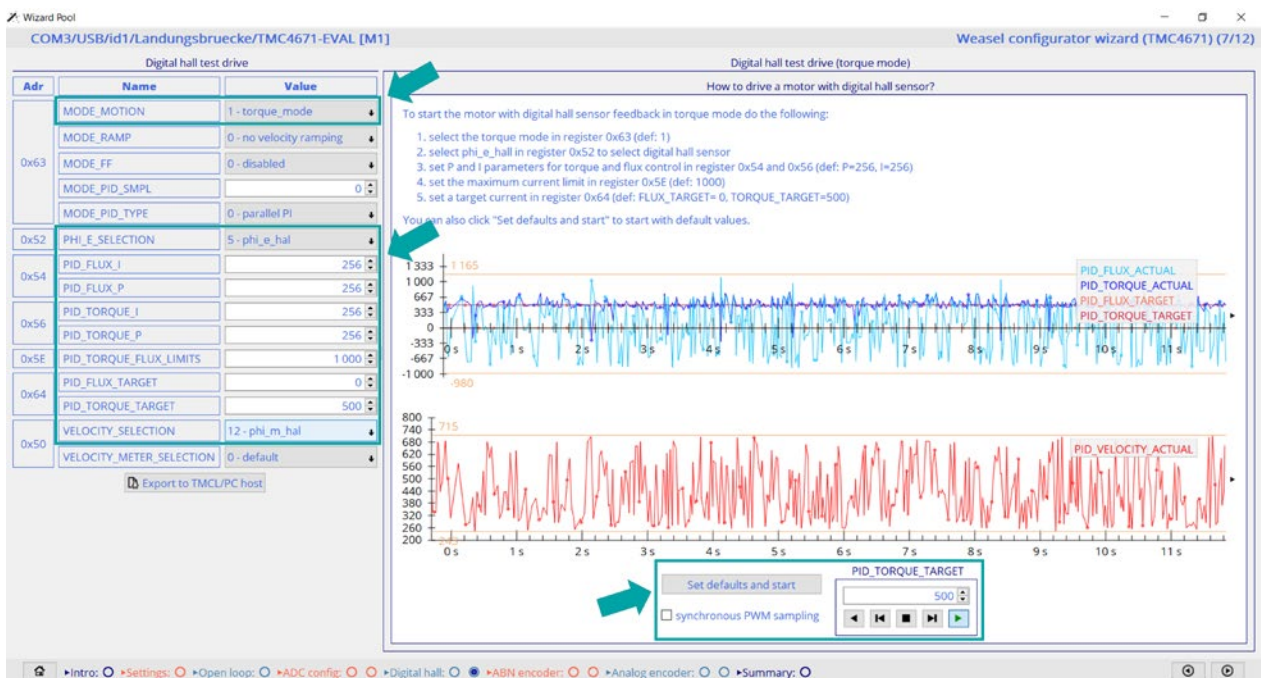


Abbildung 15: Testfahrt mit digitalen Hall-Signalen im Closed Loop-Modus

1.1.5 ABN-Encoder Konfiguration

Im Folgenden werden die erforderlichen Einstellungen bezüglich eines ABN-Encoders als Feedback-System an einem Schrittmotor des Herstellers MOONS (ML23HS8L4550-01E) demonstriert. Abbildung 16 zeigt die Konfigurationsseite. Die gepunktete blaue Linie stellt wieder die durch den Open Loop-Modus generierte Position dar, die rote Linie die Position des ABN-Encoders. Die beiden Linien müssen am Ende übereinanderliegen. Dafür muss zunächst der PPR-Wert des ABN-Encoders in Register 0x26 eingegeben werden.

Achtung: Die Abkürzung PPR steht in diesem Fall nicht für Pulses Per Revolution, sondern wurde von Trinamic auf Positions Per Revolution festgelegt und ist mit Counts Per Revolution gleichzusetzen.

Anhand des Datenblatts des ABN-Encoders wurde ein Pulses Per Revolution-Wert von 2.000 ermittelt. Aufgrund der zwei um 90° phasenverschobenen Signale und der jeweils steigenden und fallenden Flanken der Signale errechnet sich der benötigte Positions Per Revolution-Wert aus dem angegebenen Wert multipliziert mit dem Faktor 4. Der Wert für die erforderliche Eingabe beläuft sich somit auf 8.000. Sollte die Auflösung des Encoders nicht bekannt sein, kann auch der von der Software geschätzte Positions Per Revolution-Wert als Eingabe genutzt werden. Dieser wird unter dem Punkt *Encoder resolution* angezeigt.

The screenshot shows the 'Weasel configurator wizard (TMC4671) (8/12)' interface. On the left, a table lists register addresses and their values:

Adr	Name	Value
0x25	apol	<input type="checkbox"/> apol
	bpol	<input type="checkbox"/> bpol
	npol	<input type="checkbox"/> npol
	use_abn_as_n	<input type="checkbox"/> use_abn_as_n
	cln	<input type="checkbox"/> cln
	direction	<input type="checkbox"/> direction
0x26	ABN_DECODER_PPR	65 536
0x27	ABN_DECODER_COUNT	54 156
0x28	ABN_DECODER_COUNT_N	48 826
0x29	ABN_DECODER_PHI_M_OFFSET	0
	ABN_DECODER_PHI_M_OFFSET	0
0x2A	ABN_DECODER_PHI_M	-11 367
	ABN_DECODER_PHI_E	21 474
0x6B	PID_POSITION_ACTUAL	-24 226 223
0x1B	N_POLE_PAIRS	50
	MOTOR_TYPE	2 - Two phase Stepper

The main configuration area shows the following steps:

- 1) Turn motor in Open Loop mode:** A graph shows the encoder position (red line) and the open loop position (blue line) over time. The y-axis ranges from -40,000 to 40,000. The x-axis shows time from 56s to 6s. A green arrow points to the 'Estimated encoder steps' field in step 3.
- 2) Direction check:** A diagram shows two lines representing the encoder angle and the open loop angle. A green arrow points to the text: 'The direction is the same, the direction bit is set correct.'
- 3) Encoder resolution:** A text box says: 'Update the encoder resolution in register 0x26. The estimated value can help you to find the correct value.' The 'Estimated encoder steps' field is set to 8172. A green arrow points to this field.
- 4) Encoder N-Pulse setting:** A text box with a 'Reset estimation' button.

Abbildung 16: ABN-Encoder Konfiguration

Nach der Eingabe des Positions Per Revolution-Werts von 8.000 ist in Abbildung 17 mittels des grünen Pfeils zu erkennen, dass die roten Linien ihre Richtung verändert haben. Allerdings weisen diese noch nicht dieselbe Richtung der blauen Linien auf. Daher muss für die Richtungsänderung unter *Value, direction* ausgewählt werden.

The screenshot shows the 'Weasel configurator wizard (TMC4671) (8/12)' interface. On the left, the 'Encoder configuration' table is visible:

Adr	Name	Value
0x25	apol	<input type="checkbox"/> apol
	bpol	<input type="checkbox"/> bpol
	npol	<input type="checkbox"/> npol
	use_abn_as_n	<input type="checkbox"/> use_abn_as_n
	cln	<input type="checkbox"/> cln
	direction	<input type="checkbox"/> direction
0x26	ABN_DECODER_PPR	8 000
0x27	ABN_DECODER_COUNT	4
0x28	ABN_DECODER_COUNT_N	11 962
0x29	ABN_DECODER_PHI_M_OFFSET	0
	ABN_DECODER_PHI_E_OFFSET	0
0x2A	ABN_DECODER_PHI_M	81
	ABN_DECODER_PHI_E	4 050
0x6B	PID_POSITION_ACTUAL	-89 606 108
0x1B	N_POLE_PAIRS	50
	MOTOR_TYPE	2 - Two phase Stepper

The main configuration area shows the following steps:

- 1) Turn motor in Open Loop mode: Includes a graph showing 'OPENLOOP_PHI' (blue) and 'ABN_DECODER_PHI_E' (red) signals. A green box highlights the graph, and a green arrow points to the 'direction' checkbox in the settings table.
- 2) Direction check: Includes a diagram showing two cases: 'The direction is different, please change the encoder direction bit in register 0x25.' and 'The direction is the same, the direction bit is set correct.'
- 3) Encoder resolution: Includes a field for 'Estimated encoder steps: 7945' and a 'Reset estimation' button.
- 4) Encoder N-Pulse setting

Abbildung 17: Korrektur der Auflösung des ABN-Encoders

Nachdem *direction* ausgewählt wurde, wird aus Abbildung 18 ersichtlich, dass die beiden Linien die gleichen Richtungen aufweisen. Zwischen den Linien besteht jedoch noch ein kleiner Offset. Um diesen zu eliminieren, wird eine Methode namens *Bang-Bang* genutzt. Anhand dieser Methode erfolgt eine kurze starke Bestromung des Motors, sodass sich dieser neu ausrichten kann. Anschließend wird der Offset ermittelt und der ABN-Encoder korrekt referenziert.

The screenshot shows the 'Weasel configurator wizard (TMC4671) (8/12)' interface. On the left, the 'Encoder configuration' table is visible, with the 'direction' checkbox checked. The main window displays a graph titled 'Encoder configuration' with the subtitle 'How to configure encoder settings?'. The graph shows two lines: 'OPENLOOP_PHI' (red) and 'ABN_DECODER_PHI_E' (blue). The lines are nearly parallel but have a small offset. Below the graph, there are instructions for direction check and encoder resolution.

Adr	Name	Value
0x25	apol	<input type="checkbox"/> apol
	bpol	<input type="checkbox"/> bpol
	npol	<input type="checkbox"/> npol
	use_abn_as_n	<input type="checkbox"/> use_abn_as_n
	cdn	<input type="checkbox"/> cdn
	direction	<input checked="" type="checkbox"/> direction
0x26	ABN_DECODER_PPR	8 000
0x27	ABN_DECODER_COUNT	2 818
0x28	ABN_DECODER_COUNT_N	7 235
0x29	ABN_DECODER_PHI_M_OFFSET	0
	ABN_DECODER_PHI_E_OFFSET	0
0x2A	ABN_DECODER_PHI_M	23 011
	ABN_DECODER_PHI_E	-29 098
0x6B	PID_POSITION_ACTUAL	-115 264 527
0x1B	N_POLE_PAIRS	50
	MOTOR_TYPE	2 - Two phase Stepper

1) Turn motor in Open Loop mode
For a check of the encoder resolution and direction, the motor must be rotated in Open Loop mode with a controlled commutation angle, a defined force, and no load.

2) Direction check
Check whether the Open Loop angle and the encoder angle run in the same direction and update the direction bit if necessary:

The direction is different, please change the encoder direction bit in register 0x25.

The direction is the same, the direction bit is set correct.

3) Encoder resolution
Update the encoder resolution in register 0x26. The estimated value can help you to find the correct value.
Estimated encoder steps: 7989 Reset estimation

4) Encoder N-Pulse setting

Abbildung 18: Beide Linien weisen die gleiche Richtung auf, haben jedoch noch einen Offset

Um die *Bang-Bang* Methode zu aktivieren, muss die erste Schaltfläche *Init with offset estimation (Firmware)* angewählt werden (Abbildung 19).

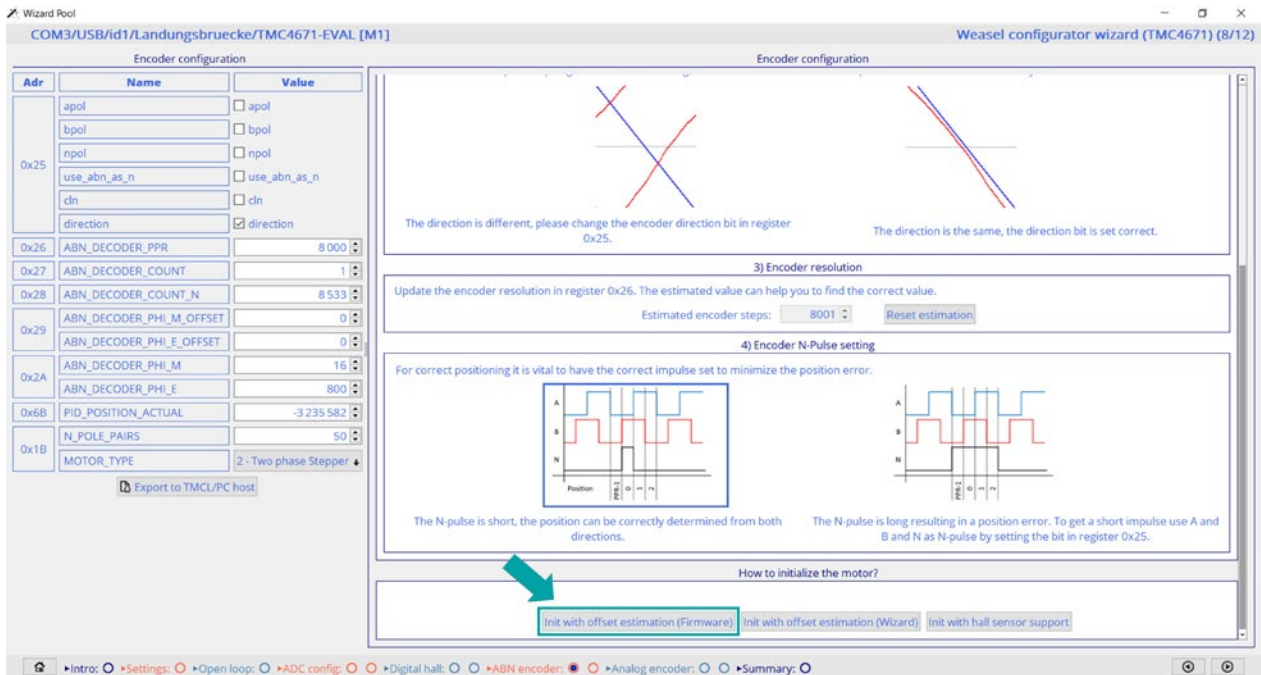


Abbildung 19: Initialisierung des Motors

Nach der Durchführung der *Bang-Bang* Methode wurde der Offset beseitigt und der ABN-Encoder ist korrekt referenziert (Abbildung 20). Damit ist dieser Konfigurationsschritt beendet.

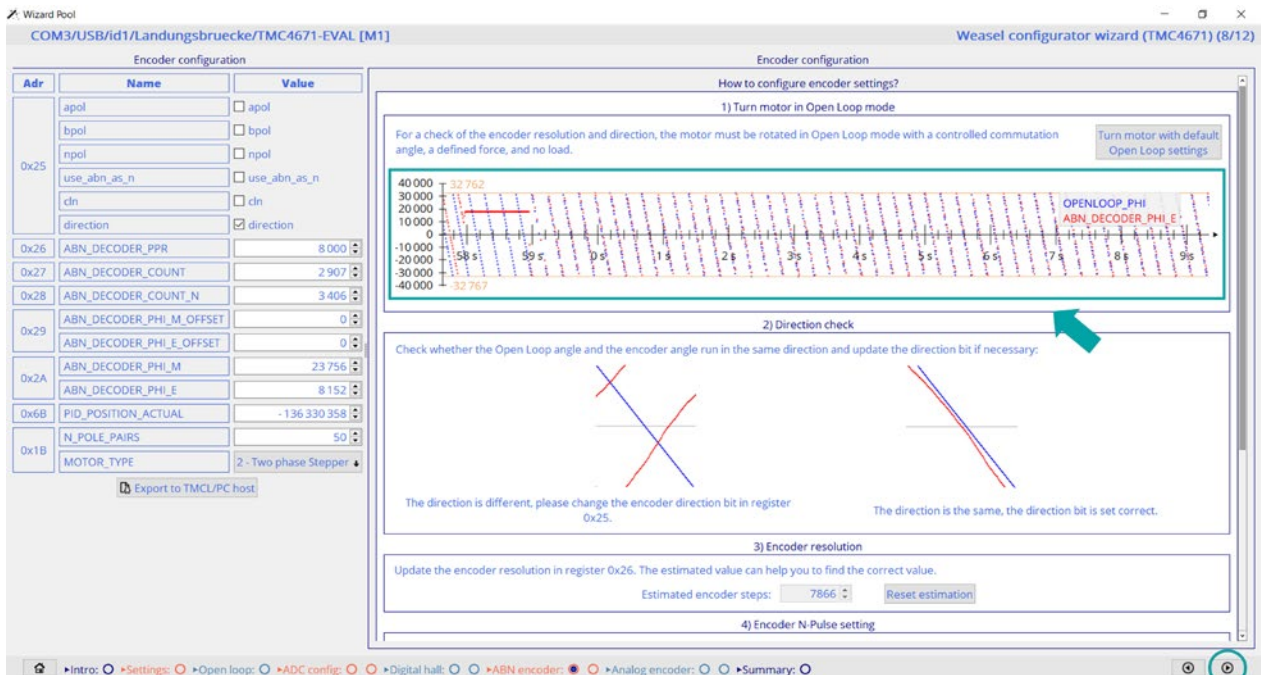


Abbildung 20: Eliminierung des Offsets durch die Bang-Bang Methode

Identisch zu den Einstellungen der Hall-Signale kann auch der eingestellte ABN-Encoder im nächsten Schritt anhand des Closed Loop-Modus getestet werden (Abbildung 21). Dazu muss an dieser Stelle ebenfalls die *Set defaults and start* Schaltfläche ausgewählt werden, sodass der Stromregelungsmodus *torque mode* aktiviert wird. Die dadurch hervorgerufenen Veränderungen in den Registern sind bis auf Register 0x52 identisch zu dem Test mit den Hall-Signalen. In Register 0x52 wird bezüglich der Kommutierung jedoch das Winkelsignal des ABN-Encoders, *phi_e_abn*, ausgewählt. Für die Darstellung der mechanischen Drehzahl, anhand der roten Linie im unteren Graphen, wurde in Register 0x50 das Winkelsignal *phi_m_abn* ausgewählt.



Abbildung 21: Closed Loop-Modus zum Testen der Einstellungen des ABN-Encoders

1.1.6 Analog-Encoder Konfiguration

Sollte ein Analog-Encoder genutzt werden, lässt sich dieser in den letzten beiden Schritten einstellen. Die vorzunehmenden Einstellungen sind identisch zu den bereits vorgestellten ADC Einstellungen und dem Testen im Closed Loop-Modus.

1.1.7 Zusammenfassung

Die Grundeinstellungen im *Wizard* sind damit beendet. Zum Schluss erscheint eine letzte Seite, auf der ein Skript mit einer Zusammenfassung der gesamten durchgeführten Einstellungen generiert wird (*Abbildung 22*). Über den zweiten Reiter C-Code wird das Skript als C-Code erstellt. Weitere Funktionen finden sich unter den vier Schaltflächen. Mithilfe der ersten Schaltfläche, welche mit einem kleinen Pfeil versehen ist, lässt sich das Feedback-System auswählen, das verwendet werden soll. Nach der Auswahl des jeweiligen Feedback-Systems wird das Skript durch entsprechende weitere Zeilen Code angepasst. Über *Export to TMCL/PC host* erfolgt eine Übertragung des Skripts in den TMCL/PC host. Dieser stellt ein Werkzeug dar, um vorhandene Skripte hochzuladen oder selbst zu schreiben und diese dann ausführen zu lassen. Um die durchgeführten Einstellungen abzuspeichern, muss die Schaltfläche *Export Script* ausgewählt werden. Für ein Abspeichern der Einstellungen als C-Code, wird die Schaltfläche *Export C-Code* verwendet. Ein Abspeichern der Einstellungen ist empfehlenswert, da somit die bereits durchlaufenen Schritte im *Wizard* nicht noch einmal durchgeführt werden müssen.



```

Script for TMCL/PC host: C-Code

// Select module
#module 1 COM3/USB/Id1/Landungsbruecke/TMC4671-EVAL [M1]

// Use TMC4671 register addresses
#include TMC4671_register_addresses.tpc
// (C:/Users/sjendoubi/AppData/Roaming/TRINAMIC Motion Control GmbH & Co. KG/TMCL-IDE/TMCL-Script/TMC4671_register_addresses.tpc)

// Motor type & PWM configuration
WMC MOTOR_TYPE_N_POLE_PAIRS, 0, $00030003, 1
WMC PWM_POLARITIES, 0, $00000000, 1
WMC PWM_MAXCNT, 0, $00000F9F, 1
WMC PWM_BDM_H_BDM_L, 0, $00000A04, 1
WMC PWM_SV_CHOP, 0, $00000007, 1

// ADC configuration
WMC ADC_I_SELECT, 0, $10000100, 1
WMC dsADC_MCFG_B_MCFG_A, 0, $00100010, 1
WMC dsADC_MCLK_A, 0, $20000000, 1
WMC dsADC_MCLK_B, 0, $00000000, 1
WMC dsADC_MDEC_B_MDEC_A, 0, $014E014E, 1
WMC ADC_I0_SCALE_OFFSET, 0, $01005090, 1
WMC ADC_I1_SCALE_OFFSET, 0, $01005C06, 1

// Open loop settings
WMC OPENLOOP_MODE, 0, $00000000, 1
WMC OPENLOOP_ACCELERATION, 0, $0000003C, 1
WMC OPENLOOP_VELOCITY_TARGET, 0, $FFFFFFFB, 1

// Feedback selection
WMC PHI_E_SELECTION, 0, $00000002, 1
WMC U0_ID_EXT, 0, $00000001, 1

// ===== Open loop test drive =====

// Switch to open loop velocity mode
WMC MODE_RAMP_MODE_MOTION, 0, $00000008, 1

// Rotate right
WMC OPENLOOP_VELOCITY_TARGET, 0, $0000003C, 1
WAIT TICKS, 1, 2000
  
```

Abbildung 22: Zusammenfassung aller durchgeführten Einstellungen im Wizard

Abbildung 23 zeigt abschließend und zusammenfassend den Ablauf der Anfangskonfiguration anhand des Wizard's sowie den Inhalt der einzelnen Schritte auf.

Anfangskonfiguration über den Wizard

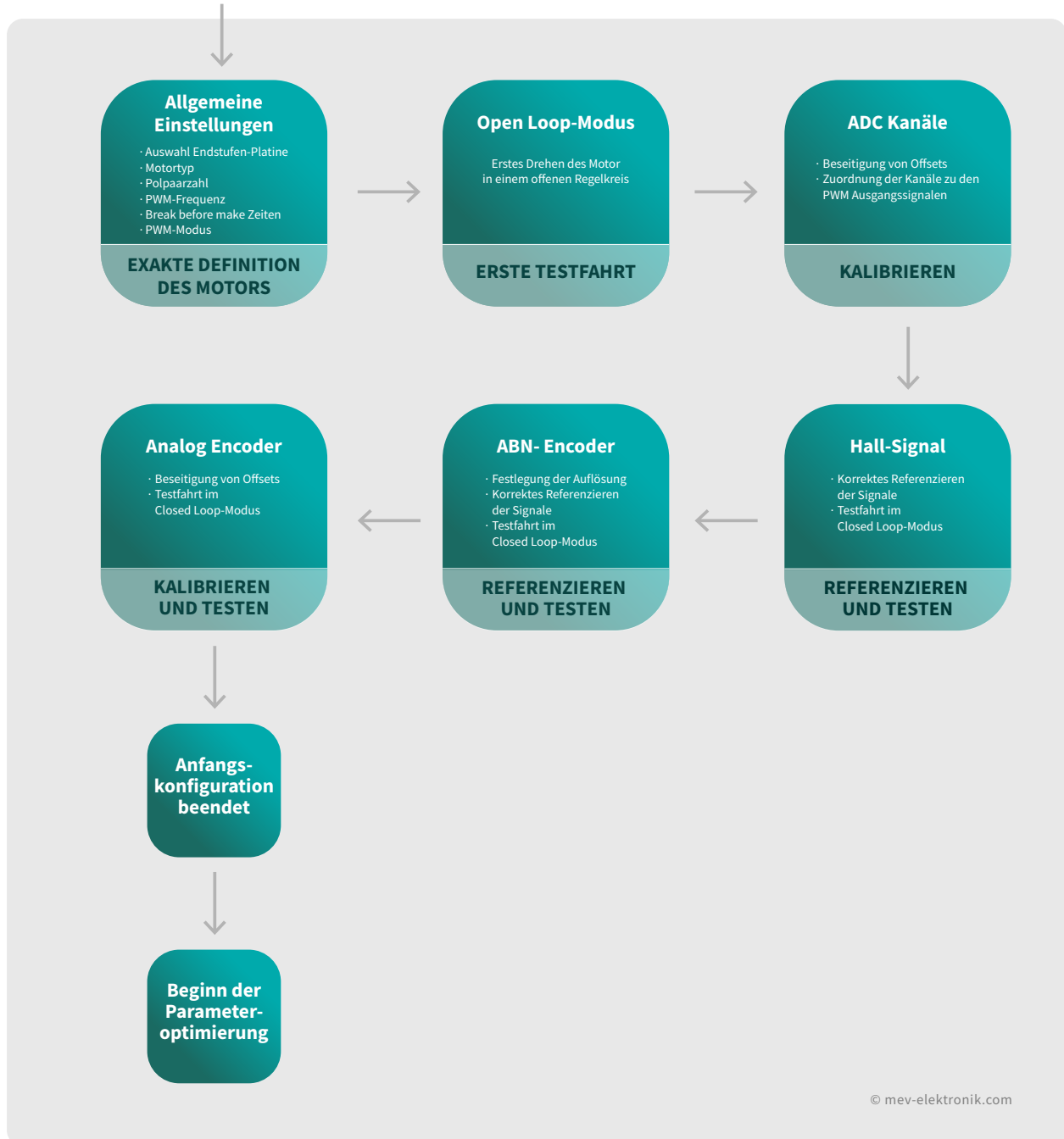


Abbildung 23: Ablauf der Anfangskonfiguration anhand des Wizard's

“

1.2 Parameteroptimierung

Nach der Durchführung der Grundeinstellungen erfolgt das Optimieren der Reglerparameter. Da die Regelkreise im TMC4671 als Kaskadenregelung aufgebaut sind, werden zu Beginn die Parameter für die Stromregelung, anschließend für die Geschwindigkeitsregelung und zuletzt für die Positionsregelung optimiert. Die Optimierung wird anhand des USB-2-RTMI Moduls durchgeführt, mit dem in Echtzeit auf die Registerwerte zugegriffen werden kann. Nachdem das Modul über den USB-Anschluss mit einem PC verbunden wurde, wird es in der linken Leiste der TMCL-IDE erkannt und angezeigt (Abbildung 24). Unterhalb der Bezeichnung des Moduls öffnet sich ähnlich der Landungsbrücke ein weiteres Aufklappmenü. Für die Parameteroptimierung sind unter der Rubrik *Tuning* das *Torque/Flux tuning*- und das *Step response-tool* erforderlich (Abbildung 24). Mithilfe des *Torque/Flux tuning tool*'s lassen sich die Reglerparameter des Torque- und Flux-Reglers im Open Loop-Modus, in einem offenen Regelkreis, ermitteln. Anhand des *Step response tool*'s wird das Systemverhalten mittels der Sprungantwort im Closed Loop-Modus, in einem geschlossenen Regelkreis, analysiert.

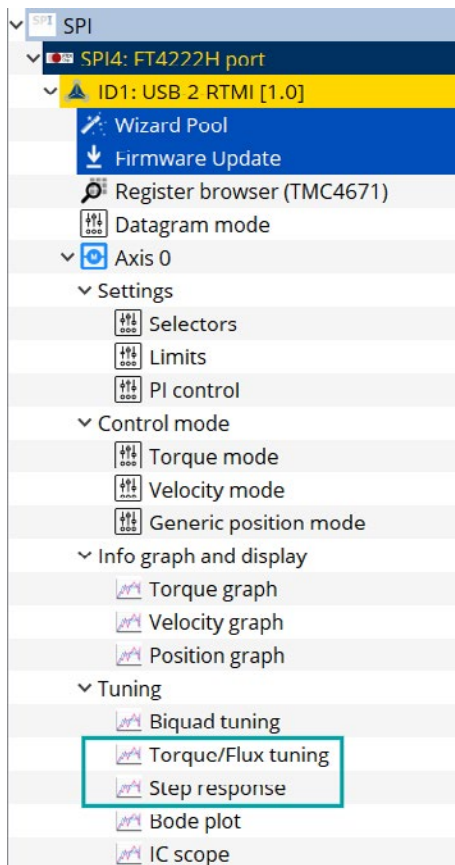


Abbildung 24: Aufklappmenü des USB-2-RTMI Moduls

1.2.1 Limit-Einstellungen

Die Parameteroptimierung wird hier beispielhaft anhand des bereits vorgestellten Schrittmotors des Herstellers MOONS (ML23HS8L4550-01E) mit einem ABN-Encoder durchgeführt.

Bevor mit der Parameteroptimierung begonnen werden kann, erfolgt die Festlegung einiger Limits. Dieser Schritt der Begrenzung ist erforderlich, damit der verwendete Motor beispielsweise nur mit dem im Datenblatt maximal zulässigen Strom gespeist und nicht beschädigt wird. Die Einstellung erfolgt im Aufklappmenü unter dem Punkt *Limits*. Dabei öffnet sich ein Fenster für die Eingabe von Werten in diversen Registern (*Abbildung 25*). In Register 0x5D muss ein Wert für die maximale Spannung, die zur Verfügung stehen soll, eingetragen werden. Die durch das Netzteil angelegte Spannung beträgt 24V und sollte dem Motor komplett zur Verfügung stehen, weshalb in diesem Feld der maximale Registerwert von 32.767 eingetragen werden muss.

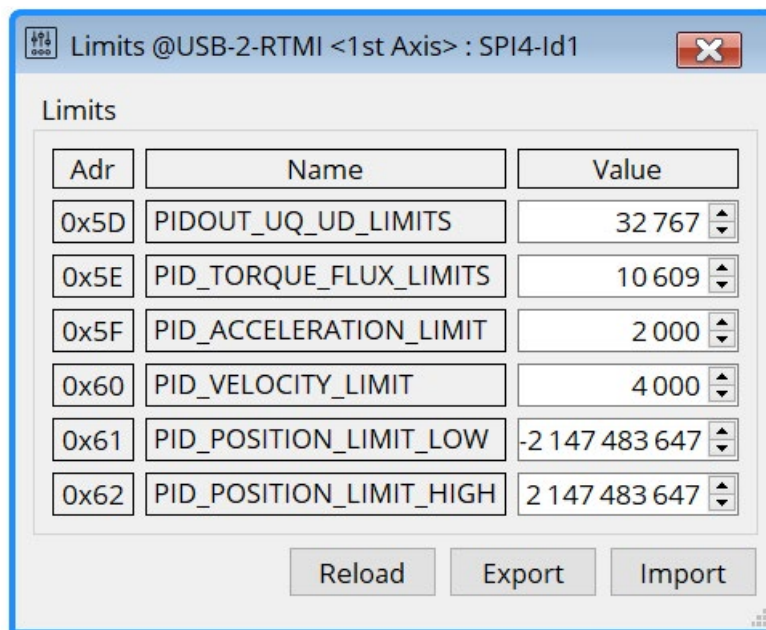


Abbildung 25: Einstellung der Limits

Register 0x5E bezieht sich auf das Stromlimit, um den maximal zulässigen Strom für den Motor einzustellen. Für die Umwandlung des im Datenblatt des Motors angegebenen Maximalstroms in den erforderlichen Wert für das Register, wird ein Umrechnungsfaktor benötigt. Dieser lässt sich für jede Endstufen-Platine über folgende Formel berechnen:

$$\frac{5V}{R_{sense} \times G \times 65.535} \cdot (1.4)$$

In der Formel werden die maximal verfügbare Eingangsspannung der *ADC Einheit* (5 V) sowie der maximale Registerwert von 65.535 verwendet. R_{sense} ist dabei ein Messwiderstand, über den eine Spannung abfällt, welche durch den Verstärkungsfaktor G verstärkt und anschließend an die *ADC Einheit* weitergeleitet wird. Die Werte für R_{sense} und G müssen den Schaltbildern der jeweiligen verwendeten Endstufen-Platine entnommen werden. Für die in diesem Beispiel genutzte TMC-UPS-10A70V-A-EVAL-Endstufen-Platine lauten diese: $R_{sense} = 0,0075 \Omega$ und $G = 20\%$. Anhand von Formel 1.4 lässt sich folgender Umrechnungsfaktor berechnen:

$$\frac{5 V}{0,0075 \Omega \times 20\% \times 65.535} = 0,000509 A. \quad (1.5)$$

Der berechnete Umrechnungsfaktor bezieht sich auf das Bit mit der niedrigsten Wertigkeit (least significant Bit, LSB) und wird daher als 0,000509 A/LSB angegeben. Dem Datenblatt des hierbei verwendeten Schrittmotors wurde ein Maximalstrom von 5,5 A entnommen. Aus Sicherheitsgründen erfolgte die Festlegung eines maximalen Werts von 5,4 A. Der erforderliche Registerwert ergibt sich aus folgender Berechnung:

$$\frac{5,4 A}{0,000509 \frac{A}{LSB}} = 10.609. \quad (1.6)$$

Das Register 0x5F für das Beschleunigungslimit kann ignoriert werden, da dieses momentan keine Funktion hat und daher für die Parameteroptimierung nicht relevant ist. Ein Geschwindigkeitslimit muss in Register 0x60 auf die maximal gewünschte Geschwindigkeit der Anwendung eingestellt werden. Dieses Limit ist, je nach gewähltem Kommutierungswinkel, auf eine mechanische oder elektrische Umdrehung normiert. In diesem Fall wurde dafür ein Geschwindigkeitslimit von 4.000 rpm eingetragen. In den Registern 0x61 und 0x62 ist keine Änderung der Werte notwendig. Sie stellen den minimalen und maximalen Wert des Eingangs der Positionsregler dar.

1.2.2 Parameteroptimierung des Stromreglers (Open Loop)

Zu Beginn wird die Stromregelung anhand des *Torque/Flux tuning tool's* optimiert, indem zunächst im offenen Regelkreis Reglerparameter für die Stromregler ermittelt werden. Dabei wird zur Anregung des Systems über den Sollwert eine Spannung vorgegeben, um eine Sprungantwort zu erzeugen.

Daraufhin wird das Verhalten des Stroms analysiert. Unter *Stimulation* muss der Sollwert und die Anzahl an Samples eingegeben werden (*Abbildung 26*). Die Samples müssen abhängig von dem Verlauf der Sprungantwort und der gewünschten Dauer der Messung gewählt werden. In diesem Fall wurden eine Samplezahl sowie ein Sollwert von 1.000 vorgegeben. Allgemein kann zu Beginn jeder Messung der Sollwert durch den Anwender frei gewählt werden. Im weiteren Verlauf der Parameteroptimierung der Stromregler handelt es sich bei den Sollwerten um reine Registerwerte, denen keine physikalischen Einheiten zugordnet werden können.

Anschließend kann die Messung über die Schaltfläche *Start* begonnen werden. In *Abbildung 26* ist zu erkennen, dass die Sprungantwort nicht komplett sichtbar ist und eine schwankende blaue Linie generiert wurde.

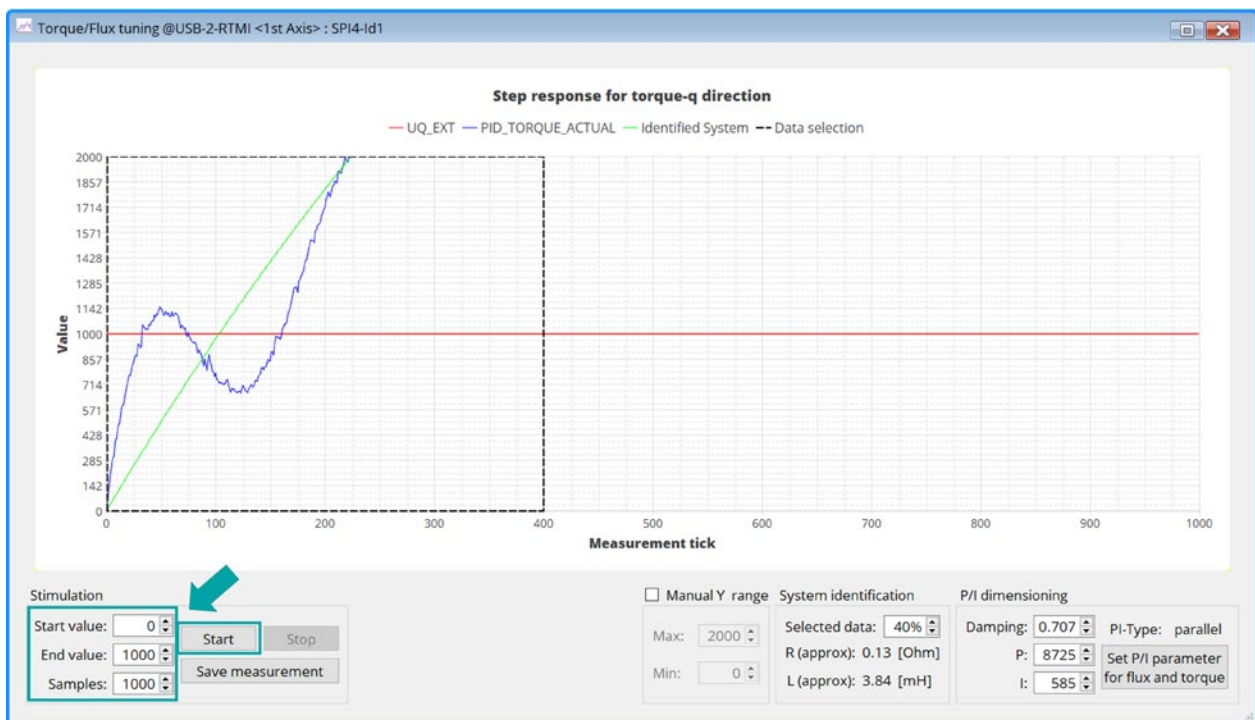


Abbildung 26: Erste aufgenommene Sprungantwort des Torque/Flux tuning tool's

Für eine bessere Darstellung sollte der Maximalwert auf der Y-Achse unter *Manual Y range* erhöht und die Messung noch einmal durchgeführt werden, sodass sich der Motor erneut ausrichten kann [1]. Wie in Abbildung 27 dargestellt, wurde durch das Vergrößern des Maximalwerts der Y-Achse und einer zweiten Messung, eine deutlich verbesserte Sprungantwort generiert. Der Sollwert, der für die Spannung vorgegeben wurde, ist anhand der roten Linie im Graphen zu sehen. Die grüne Linie stellt die vom System mathematisch berechnete Sprungantwort dar, die theoretisch durch die Sollwertvorgabe generiert werden sollte. Die blaue Linie zeigt dabei die tatsächliche Sprungantwort in Form des Stroms auf.

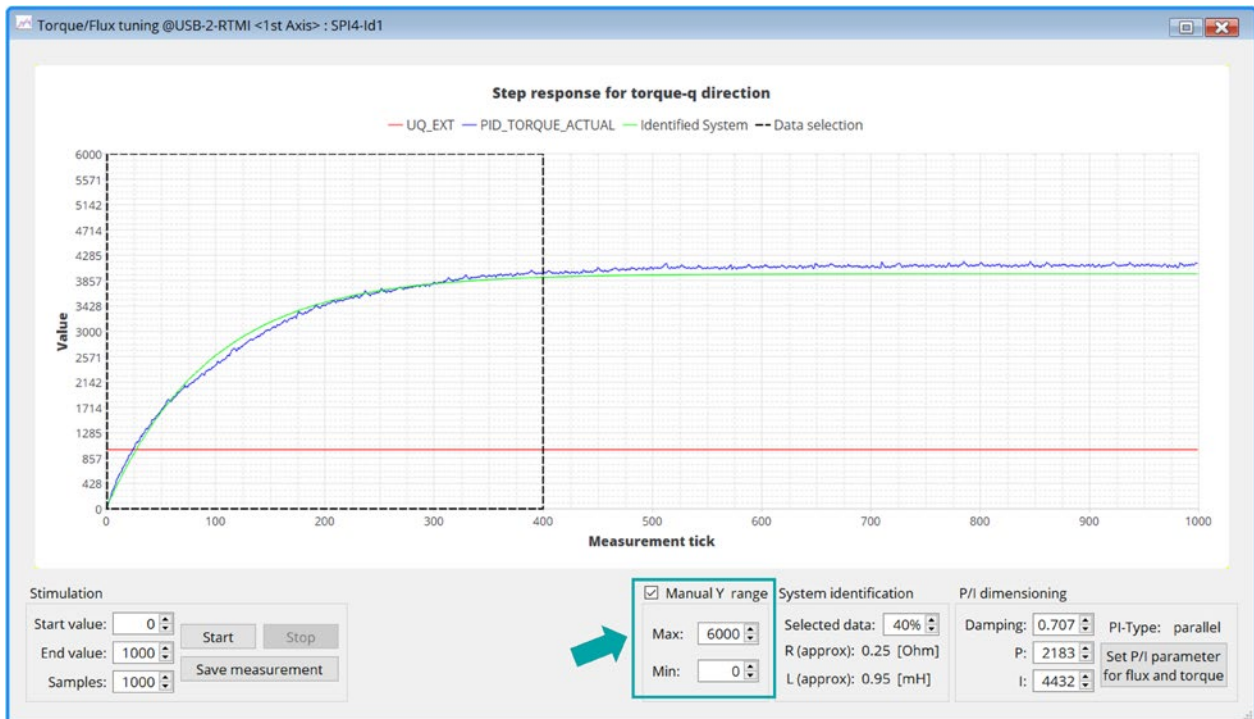


Abbildung 27: Verbesserte Darstellung nach Anpassen des Maximalwerts der Y-Achse und erneut durchgeführter Messung

Der schwarz gestrichelte Rahmen stellt den Identifikationsbereich der Sprungantwort dar und kann unter *System identification* prozentual angepasst werden. Er sollte gemäß Abbildung 28 den Verlauf der Sprungantwort bis zum Erreichen des stationären Endwerts abdecken [1]. Unter *P/I dimensioning* werden die vom System ermittelten Parameter für den P- und I-Regler angezeigt. Diese sind anschließend über die Schaltfläche *Set P/I parameter for flux and torque* zu bestätigen. Zusätzlich steht über den Werten der Reglerparameter ein Feld für einen Dämpfungsparameter zur Verfügung, anhand dessen der P-Anteil gedämpft werden kann. Wird der Wert des Dämpfungsparameters erhöht, verringert sich der P-Anteil und umgekehrt.

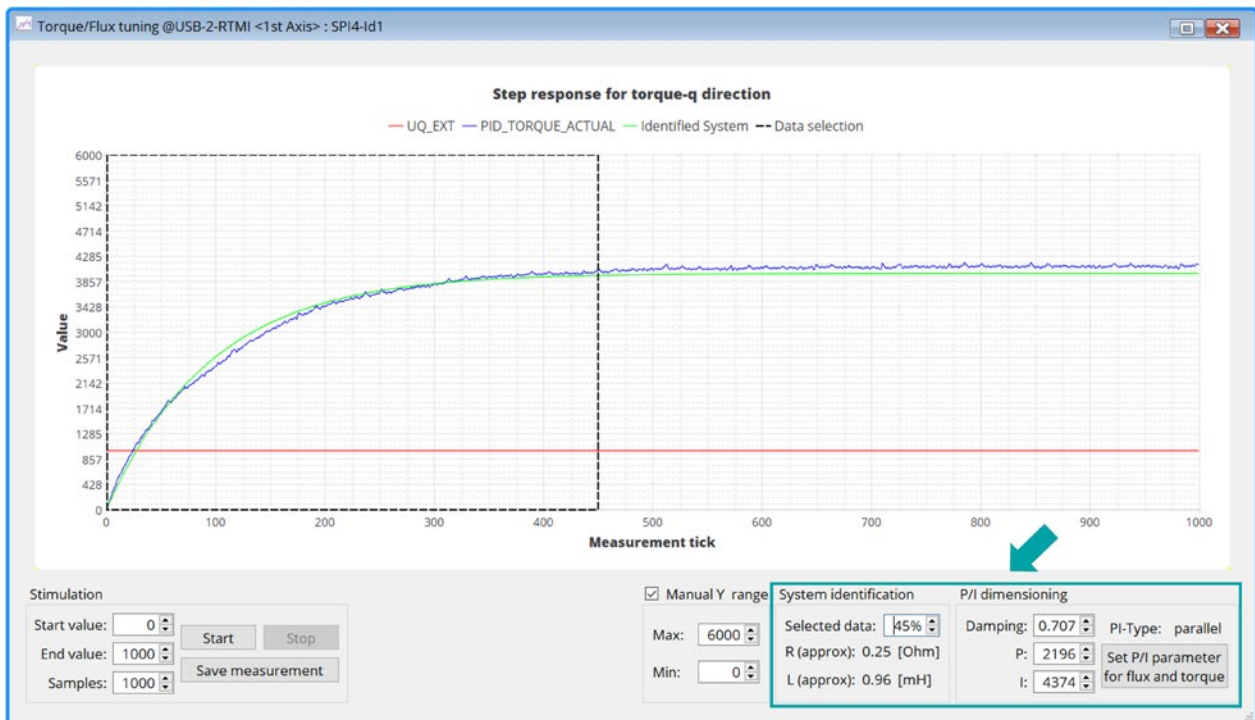


Abbildung 28: Einstellen des Identifikationsbereichs und Bestätigen der Reglerparameter

Im Aufklappmenü können unter *PI control* die übernommenen Reglerparameter eingesehen werden. In diesem Fall wurden bisher nur die durch das *Torque/Flux tuning tool* ermittelten Parameter übernommen und wie in Abbildung 29 zu sehen, in die Register der Stromregler eingetragen.

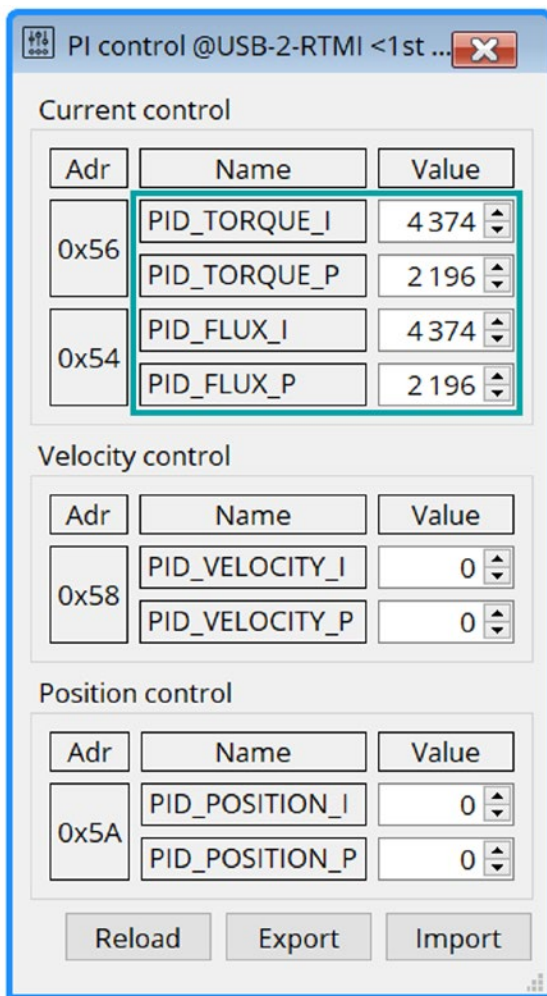


Abbildung 29: Übernommene Parameter für die Stromregler

1.2.3 Parameteroptimierung des Stromreglers (Closed Loop)

Im nächsten Schritt der Parameteroptimierung wird das *Step response tool* herangezogen, um das Systemverhalten im Closed Loop-Modus zu testen. Dabei können die durch das *Torque/Flux tuning tool* ermittelten Parameter getestet und angepasst werden. Abbildung 30 zeigt den Startbildschirm nach Öffnen des Tool's.

Für die erste Messung wurde *PID_FLUX* als Ziel- und Messwert ausgewählt. Unter *Sampling settings* ist ebenfalls eine Anzahl an Samples und zusätzlich eine Samplerate festzulegen. Die Samplerate stellt die Abtastfrequenz f_s dar und dient der Auflösung des Graphen, um eine bestmögliche Darstellung der Sprungantwort zu erhalten. Diese wird gleich der im *Wizard* eingestellten PWM-Frequenz gesetzt. Im *Wizard* wurden dabei 25 kHz ausgewählt.

Ist jedoch nicht der komplette Verlauf der Sprungantwort zu sehen, gilt es die Samplerate zu verringern. Um sich den zeitlichen Verlauf der Messung auf der X-Achse des Graphen in Sekunden anzeigen zu lassen, wird das Abtastintervall t_s benötigt. Dieses gibt die Zeit zwischen den Samples an. Mithilfe der Abtastfrequenz f_s berechnet sich dieses aus:

$$t_s = \frac{1}{f_s}. \quad (1.7)$$

In diesem Fall lautet die Berechnung:

$$t_s = \frac{1}{25 \text{ kHz}} = 0,00004 \text{ s}. \quad (1.8)$$

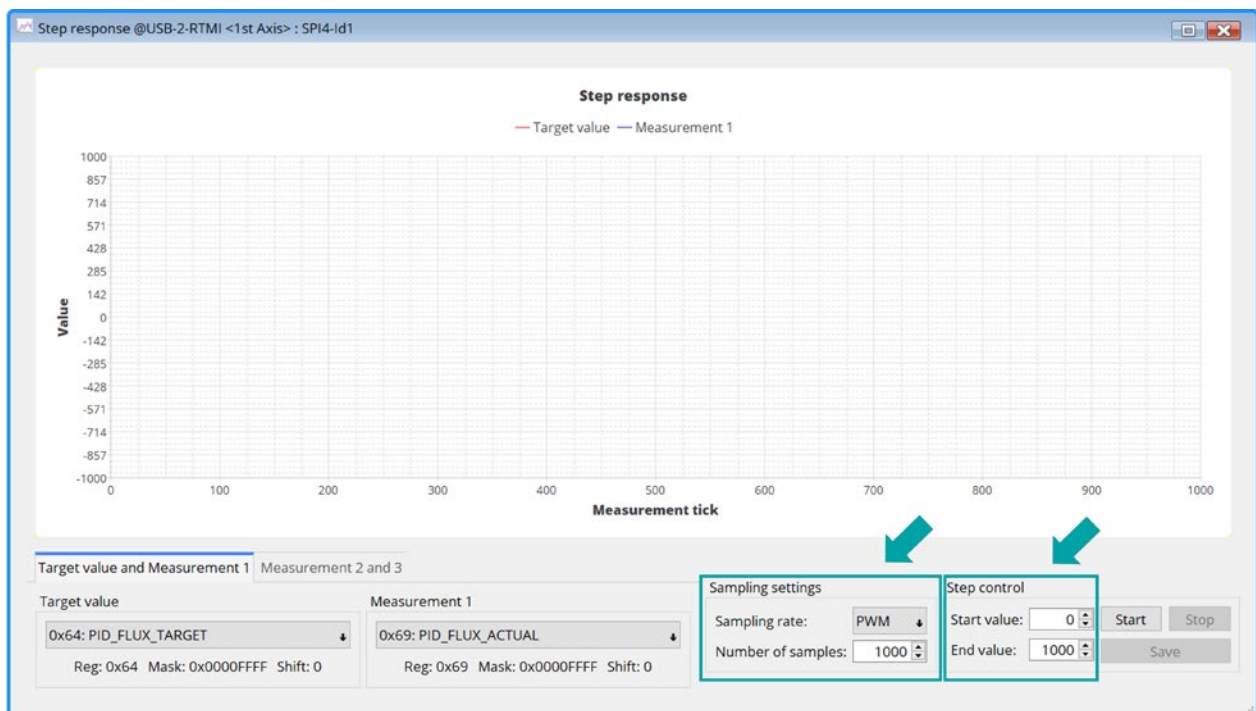


Abbildung 30: Startbildschirm des *Step response tool*'s

Auf Grundlage des berechneten Abtastintervalls t_s entspricht eine Messung mit 1.000 Samples:

$$1.000 \times 0,00004 \text{ s} = 0,04 \text{ s} = 40 \text{ ms.} \quad (1.9)$$

Mithilfe dieser Berechnungen kann somit in den folgenden Sprungantworten immer eine Umrechnung der Anzahl der Samples in Sekunden durchgeführt werden.

Des Weiteren ist auch an dieser Stelle unter *Step control* die Festlegung eines Sollwerts erforderlich, der in diesem Fall auf 1.000 festgelegt wurde. Nach Auswählen der *Start* Schaltfläche wird die erste Messung aufgenommen. Die rote Linie in Abbildung 31 stellt den festgelegten Sollwert und die blaue Linie die Regelgröße dar. Es ist zu erkennen, dass zu Beginn der Messung ein sprunghafter Anstieg der Regelgröße auf den Sollwert erfolgt, dieser jedoch nicht konstant gehalten wird. Es ist ein Schwanken der Regelgröße wahrzunehmen und zudem eine, durch die eingestellte Samplezahl, sehr lange ausgeführte Messung. Daher ist es sinnvoll diese für eine bessere Darstellung der Sprungantwort in einer weiteren Messung zu verringern.

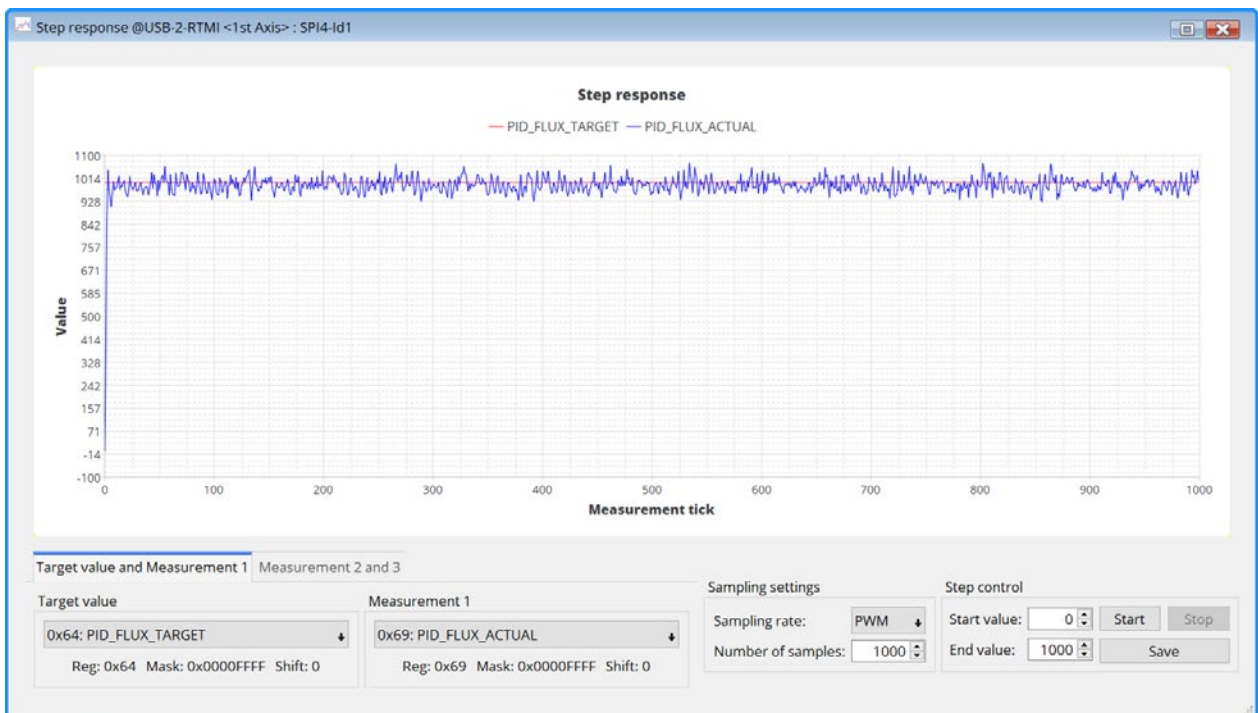


Abbildung 31: Erste aufgenommene Sprungantwort des Step response tool's

Für eine zweite Messung wurde die Samplezahl auf 100 eingestellt. In Abbildung 32 ist folglich eine verbesserte Darstellung der Sprungantwort zu sehen. Die Regelgröße schwankt im weiteren Verlauf um den Sollwert. Aufgrund des leichten Überschwingens zu Beginn der Messung und den Schwankungen im weiteren Verlauf, sollten die Reglerparameter für ein besseres Ergebnis noch weiter optimiert werden.



Abbildung 32: Weitere Messung mit reduzierter Samplezahl

Eine weitere Verbesserung der Sprungantwort wird über das Eintragen von Werten im *PI control* Fenster erreicht. Dabei werden der P- und I-Anteil für den Flux-Regler variiert und daraufhin eine neue Sprungantwort aufgenommen. Konnte eine zufriedenstellende Sprungantwort generiert werden, sind die gewählten Parameter zu übernehmen.

Durch die Optimierung der Parameter ist in Abbildung 33 zu sehen, dass ein verbessertes Ergebnis erzielt wurde. Das anfängliche Überschwingen konnte beseitigt und eine Reduzierung der Schwankung der Regelgröße erreicht werden. In Bezug auf die Schwankungen um den Sollwert ist zu erwähnen, dass es sich bei den Torque- und Flux-Reglern um hochdynamische Regler handelt, die sehr stark oszillieren. Die Schwingungen, die in den Sprungantworten zu sehen sind, weisen kein schlechtes Regelkreisverhalten auf, sondern sind auf das Verhalten der Regler zurückzuführen. Konnte eine Sprungantwort, wie in Abbildung 33, mit leichten Schwankungen der Regelgröße um den Sollwert generiert werden, ist ein gutes Regelkreisverhalten erreicht worden.

Weiter ist anzumerken, dass die Sprungantworten während dem Testen von neuen Parametern in ihrer Darstellung leichte Unterschiede aufweisen können. Dies ist den unterschiedlichen Rotorpositionen geschuldet, von denen aus die Messung begonnen wird.

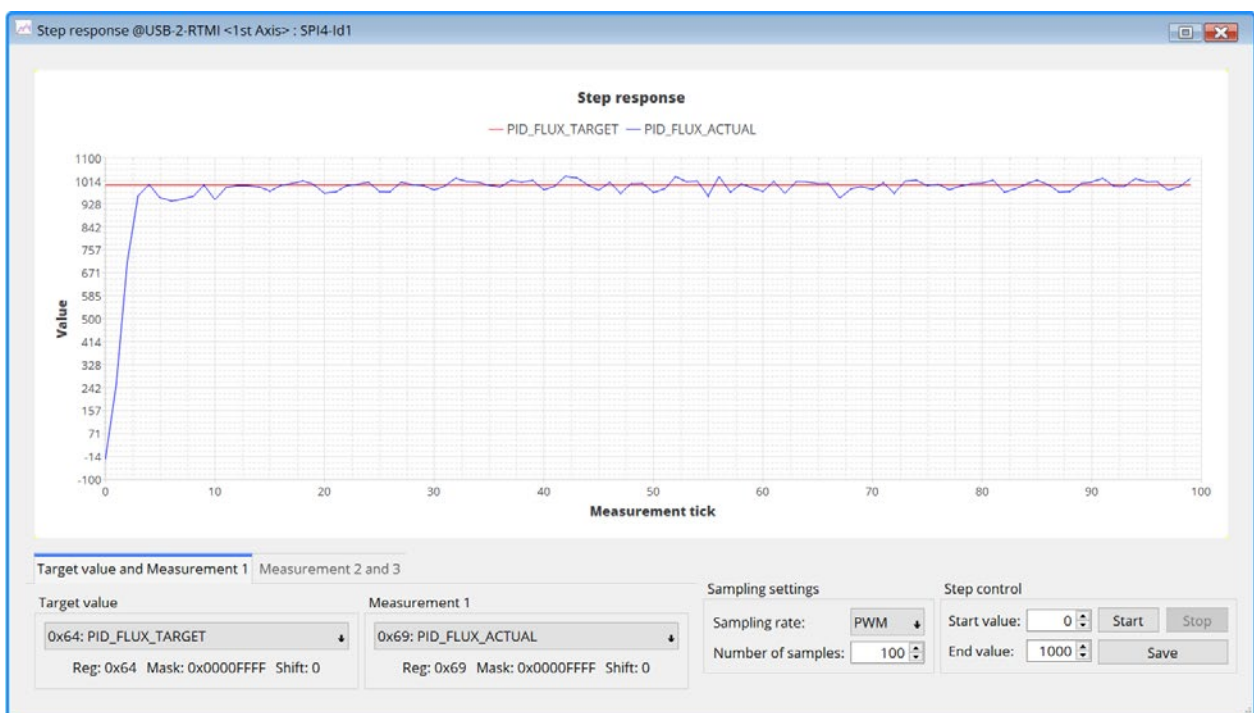


Abbildung 33: Optimierte Parameter zeigen eine verbesserte Sprungantwort

Abbildung 34 zeigt die ermittelten Parameter für den Flux-Regler im *PI control* Fenster.

Für die Parameteroptimierung des Torque-Reglers wird analog zum Ablauf des Flux-Reglers in der gleichen Weise verfahren. Es muss lediglich der Ziel- und Messwert im *Step response tool* auf *PID_TORQUE* geändert werden. An dieser Stelle wird mit der Analyse der Bode-Diagramme und anschließend mit der Parameteroptimierung des Geschwindigkeits- und Positionsreglers fortgefahren.

Current control		
Adr	Name	Value
0x56	PID_TORQUE_I	4 374
	PID_TORQUE_P	2 196
0x54	PID_FLUX_I	3 800
	PID_FLUX_P	1 800

Velocity control		
Adr	Name	Value
0x58	PID_VELOCITY_I	0
	PID_VELOCITY_P	0

Position control		
Adr	Name	Value
0x5A	PID_POSITION_I	0
	PID_POSITION_P	0

Reload Export Import

Abbildung 34: Übernahme der optimierten Parameter in das *PI control* Fenster

1.2.4 Analyse der Bode-Diagramme

Zur weiteren Analyse des Systemverhaltens kann das *Bode plot tool* herangezogen werden, welches sich im Aufklappmenü des USB-2-RTMI Moduls unter dem *Step response tool* befindet.

Abbildung 35 zeigt ein Bode-Diagramm, welches vor der Parameteroptimierung der Stromregler generiert wurde. Es wurden hierbei die vom System gewählten Parameter übernommen. Dabei handelt es sich um die Parameter, welche das System während des Testens der Feedback-Systeme im *Wizard* für den Torque- und Flux-Regler gewählt hat. In Abbildung 35 wurde über die Schaltfläche *Start* ein Bode-Diagramm für den Torque-Regelkreis generiert. Um Aussagen über die Stabilität des Systems zu treffen, müssen die Amplituden- und Phasenreserve abgelesen werden. Da die 0 dB Linie im oberen Diagramm der Verstärkung durch die Skala nicht angezeigt wird, ist ein exaktes Ablesen der Amplituden- und Phasenreserve nicht möglich. Ein weiterer Kennwert aus dem Bode-Diagramm, der zur Analyse des Regelkreises herangezogen werden kann, ist die *Cutoff frequency* (deutsch: *Grenzfrequenz*). Mit nichtoptimierten Reglerparametern liegt die Grenzfrequenz ω_{gr} bei 48 Hz, wodurch die Verstärkung schon an dieser Stelle um 3 dB gesunken ist. Die geringe Grenzfrequenz deutet auf eine geringere Bandbreite und infolgedessen auf eine lang-samere Reaktion des Regelkreises hin.

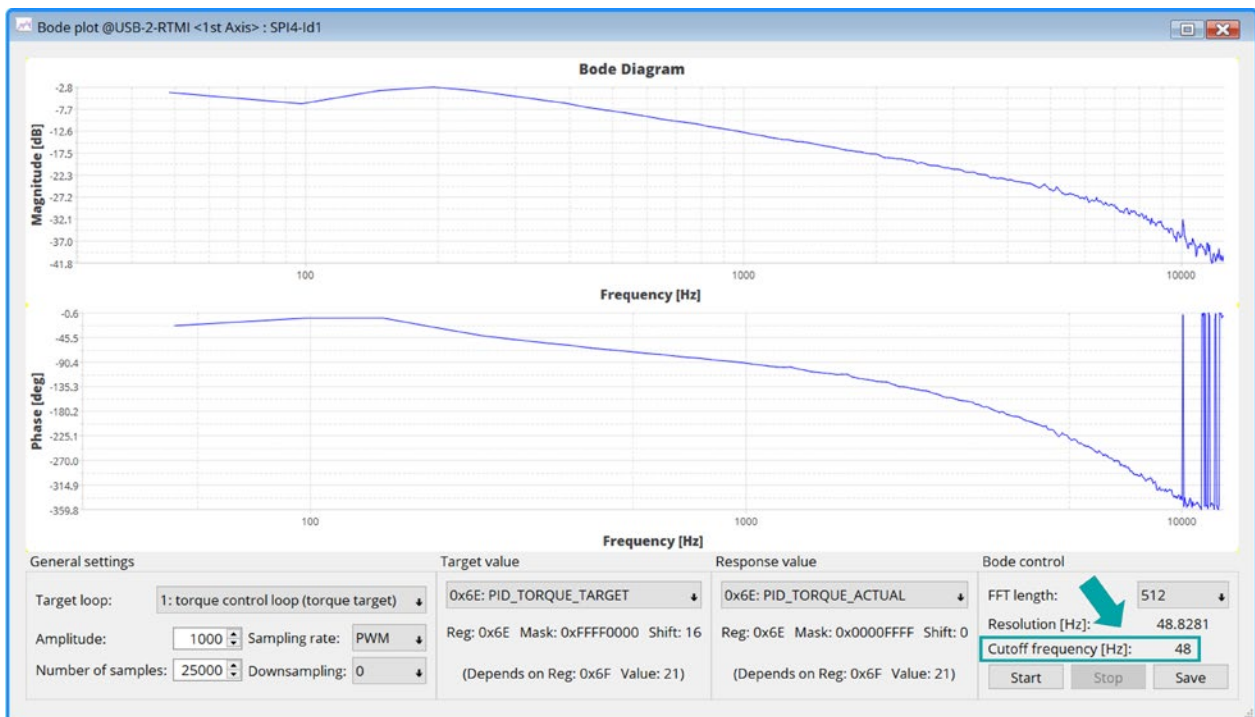


Abbildung 35: Bode-Diagramm mit nicht optimierten Reglerparametern

In einem weiteren Schritt wurde nach der Optimierung der Parameter des Torque-Reglers ein zweites Bode-Diagramm generiert. In Abbildung 36 ist zu sehen, dass die Grenzfrequenz ω_{gr} auf 3906 Hz gestiegen ist und die Amplitudenkennlinie somit erst bei dieser Frequenz um 3 dB abfällt. Damit steht eine höhere Bandbreite zur Verfügung und das System verfügt über eine schnellere Reaktion.

Da auch anhand dieses Bode-Diagramms nicht genau zu erkennen ist wo die Amplitudenkennlinie die 0 dB Marke passiert, beziehungsweise die Durchtrittsfrequenz ω_D nicht markiert ist, ist ein Ablesen der Phasenreserve ebenfalls nicht exakt möglich. Die Amplitudenreserve kann mit der angegebenen Skala nur annähernd abgelesen werden. Diese liegt ungefähr bei 4,2 dB. Stabiles oder instabiles Verhalten des Regelkreises wird deswegen über die Sprungantworten ermittelt. Die starken Anstiege der Phasenkennlinien zum Schluss in den Abbildungen 35 und 36 sind zu ignorieren, da es sich hierbei laut Trinamic um einen Softwarefehler handelt.

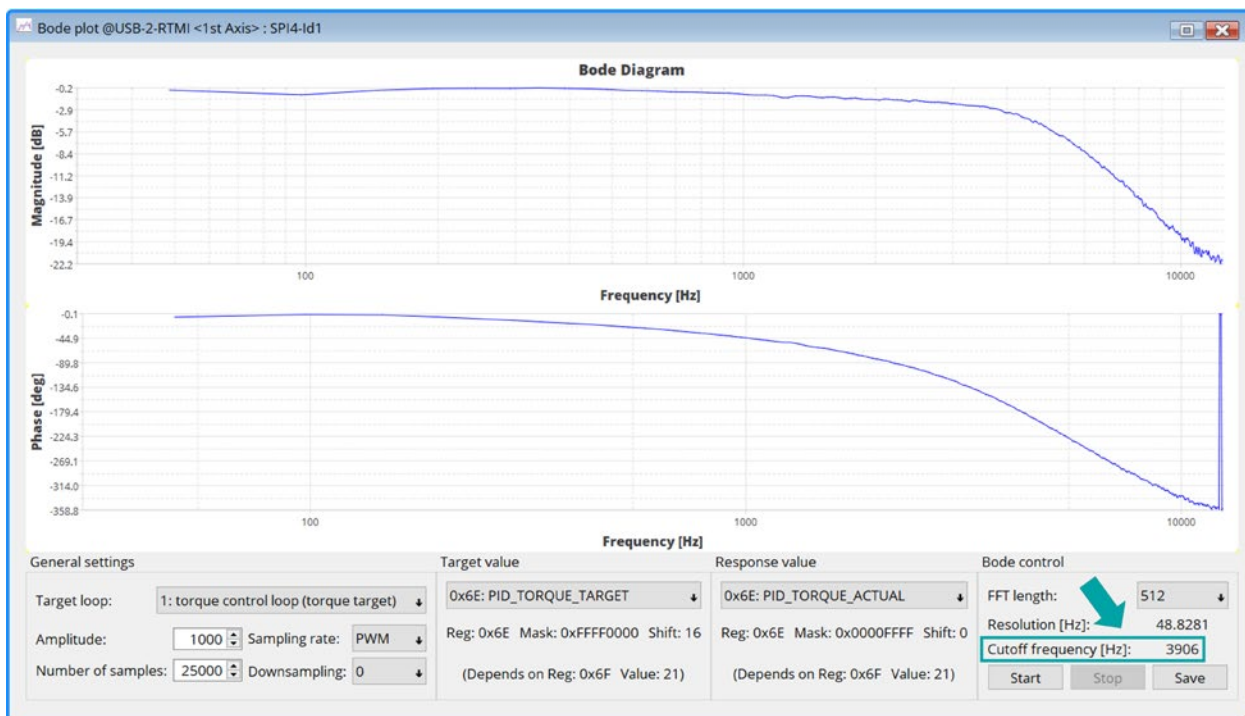


Abbildung 36: Bode-Diagramm mit optimierten Reglerparametern

1.2.5 Parameteroptimierung des Geschwindigkeitsreglers

Nachdem die Parameter für den Torque-Regler ermittelt wurden, erfolgt im nächsten Schritt die Parameteroptimierung des Geschwindigkeitsreglers. Zu Beginn muss unter den *Selectors*-Einstellungen überprüft werden, dass in Register 0x52 nicht mehr das Winkelsignal des Systems (*phi_e_openloop*) ausgewählt ist, sondern das Signal des jeweiligen Feedback-Systems. In diesem Beispiel das des ABN-Encoders (*phi_e_abn*). Zudem ist eine Änderung in Register 0x50 vorzunehmen, indem der mechanische Kommutierungswinkel *phi_m_abn* ausgewählt wird (siehe *Abbildung 37*). Somit wird im Folgenden die mechanische Drehzahl angezeigt [1].

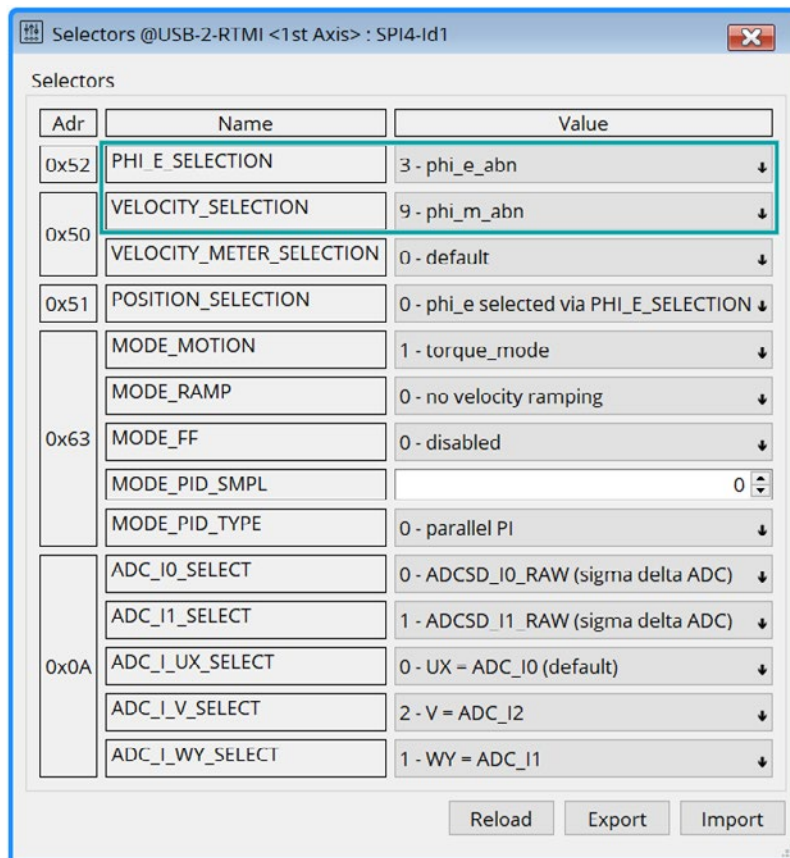


Abbildung 37: Änderung der Winkelsignale

Für die Messung wurde der Ziel- und Messwert auf `PID_VELOCITY` umgestellt, eine Anzahl von 1.000 Samples sowie zur verbesserten Darstellung eine verringerte Samplerate von PWM/32 gewählt. Als Sollwert wurde anhand der Motorkennlinie der Betriebspunkt von 500rpm festgelegt.

Hierbei wurde über das `PI control` Fenster mit `PID_VELOCITY_P=100` die Messung gestartet. In Abbildung 38 ist ersichtlich, dass der Parameter zu niedrig gewählt wurde, da die Regelgröße nahezu bei 0 verweilt.

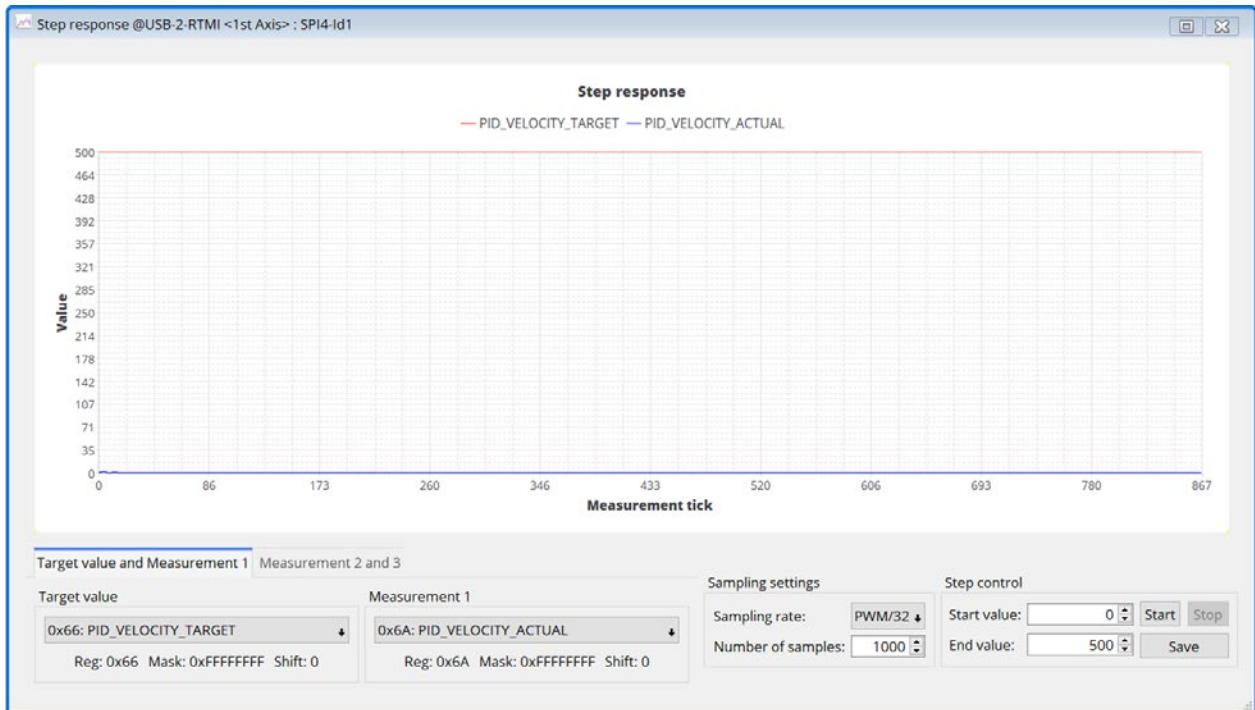


Abbildung 38: Sprungantwort mit `PID_VELOCITY_P=100`

In den nächsten Schritten sollte der P-Anteil weiter erhöht werden, bis sich die Regelgröße circa zur Hälfte dem Sollwert angenähert hat. Wie in Abbildung 39 dargestellt, wurde dies mit einem P-Anteil von 800 erreicht. Als nächstes muss der I-Anteil schrittweise erhöht werden, bis die Regelgröße den gewünschten Sollwert erreicht [1].

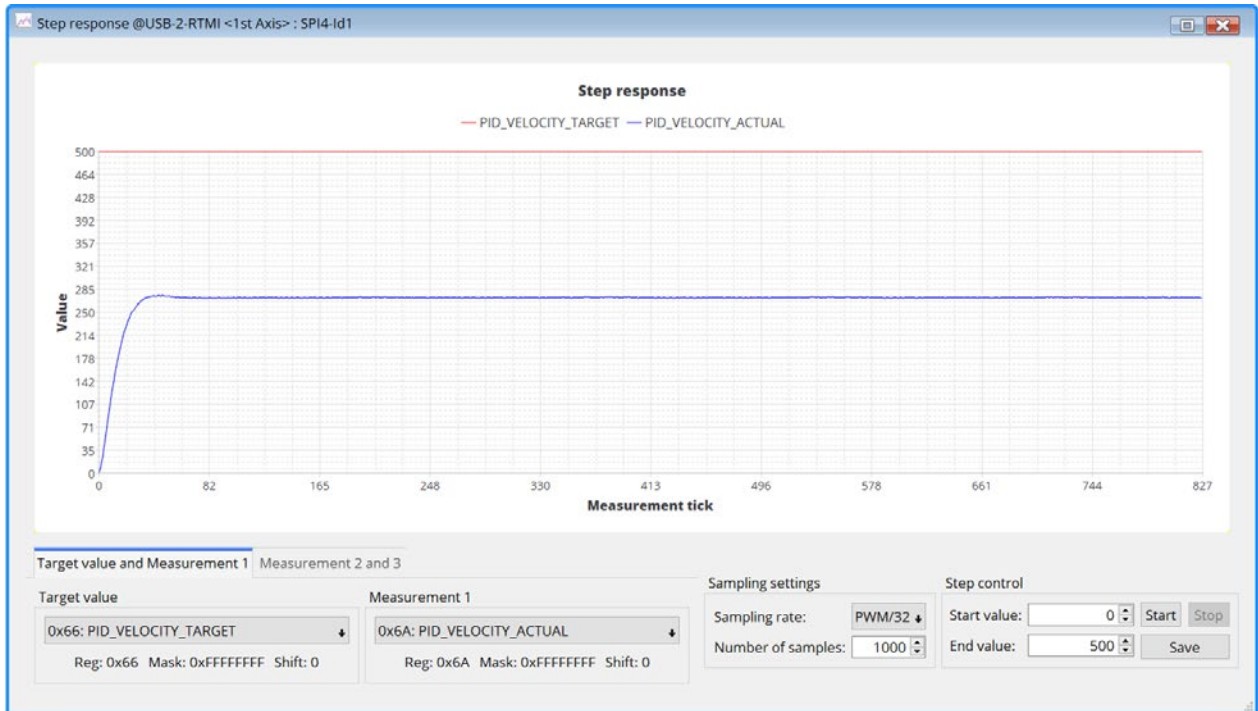


Abbildung 39: Sprungantwort mit $PID_VELOCITY_P=800$

Durch das schrittweise Erhöhen von $PID_VELOCITY_I$ auf 670, erreicht die Regelgröße nach 46 Samples (58,88 ms) einen stationären Endwert von 500 rpm. Damit weist der Regelkreis stationäre Genauigkeit auf (Abbildung 40). Somit ist die Parameteroptimierung für den Geschwindigkeitsregler beendet.

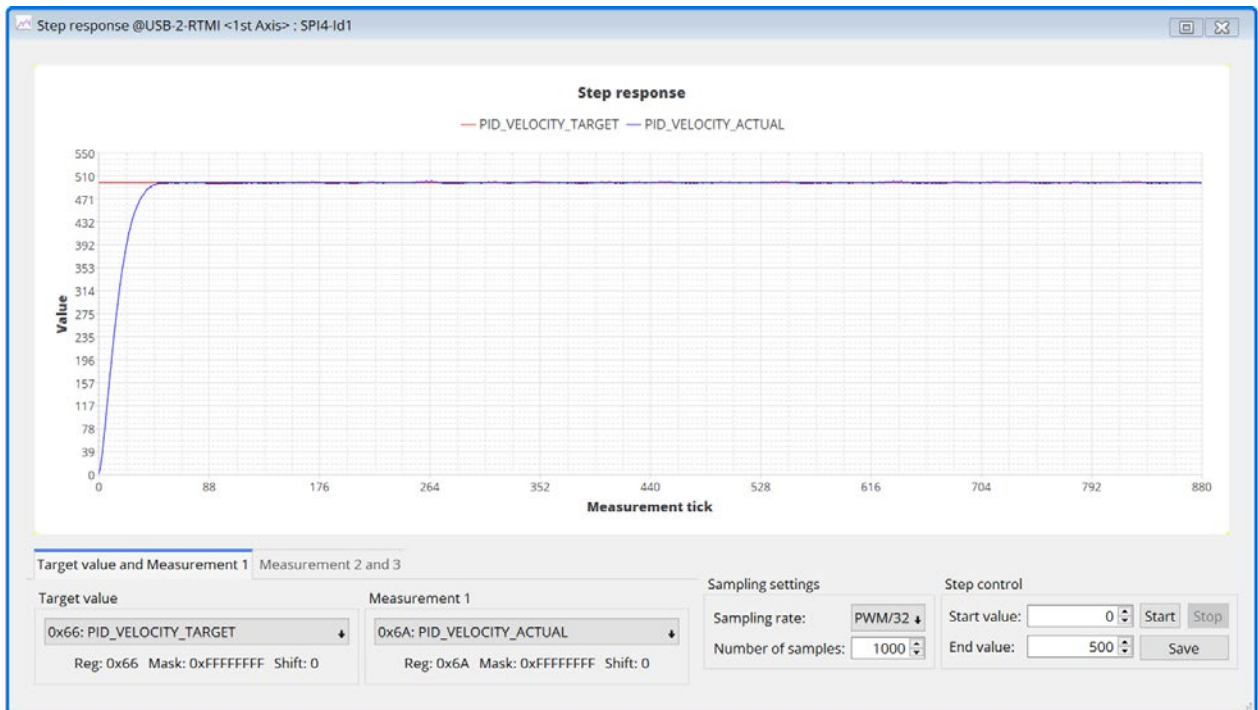


Abbildung 40: Erreichen des stationären Endwerts von 500 rpm mit $PID_VELOCITY_P = 800$ und $PID_VELOCITY_I = 670$

1.2.6 Parameteroptimierung des Positionsreglers

Bevor mit der Parameteroptimierung des Positionsreglers begonnen werden kann, ist unter den Selectors-Einstellungen in Register 0x51 der Kommutierungswinkel ϕ_{m_abn} auszuwählen [1] (Abbildung 41). Somit entspricht eine mechanische Umdrehung einem Registerwert von 65.536.

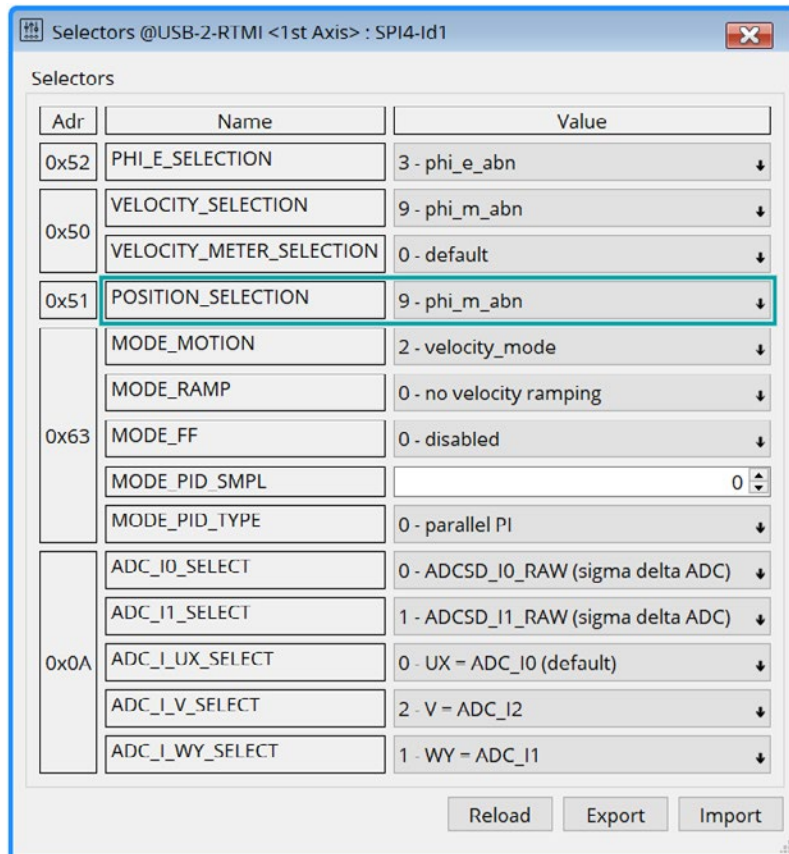


Abbildung 41: Änderung des Kommutierungswinkels auf ϕ_{m_abn}

Zunächst ist der Ziel- und Messwert auf `PID_POSITION` umzustellen. In diesem Beispiel wurde wieder eine Samplezahl von 1.000 sowie eine Samplerate von PWM/32 gewählt.

Der festgelegte Sollwert beträgt 65.536, was einer mechanischen Umdrehung entspricht. Für die erste Messung, in Abbildung 42 dargestellt, wurde mit `PID_POSITION_P = 10` begonnen. Da die Positionsregelung lediglich anhand eines P-Reglers erfolgt, muss an dieser Stelle nur der `PID_POSITION_P`-Anteil optimiert werden. Da die Regelgröße den Sollwert in diesem Fall noch nicht erreicht, muss der Wert für `PID_POSITION_P` weiter erhöht werden.

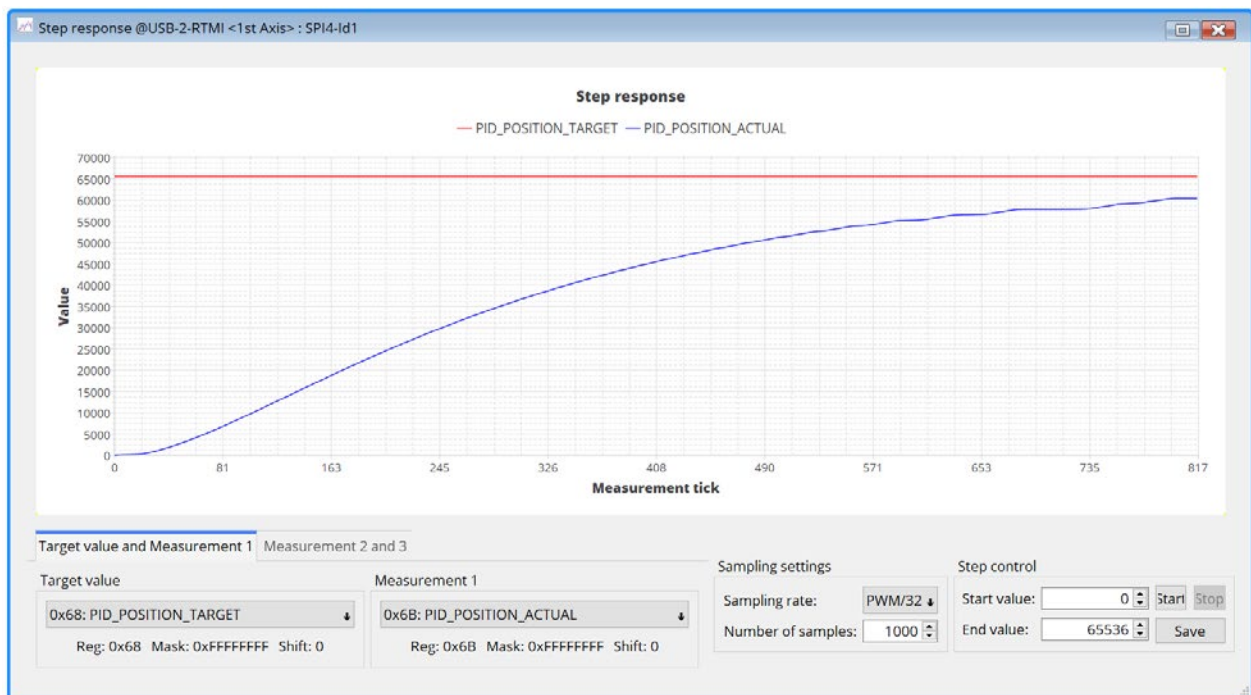


Abbildung 42: Sprungantwort mit `PID_POSITION_P = 10`

Durch schrittweises Erhöhen von `PID_POSITION_P` auf 100, erreicht die Regelgröße nach 343 Samples (0,439s) einen stationären Endwert von 65.536, womit auch in diesem Regelkreis stationäre Genauigkeit gegeben ist (Abbildung 43). Da die Reglerparameter für den Positionsregler ermittelt wurden, ist die Parameteroptimierung somit beendet.

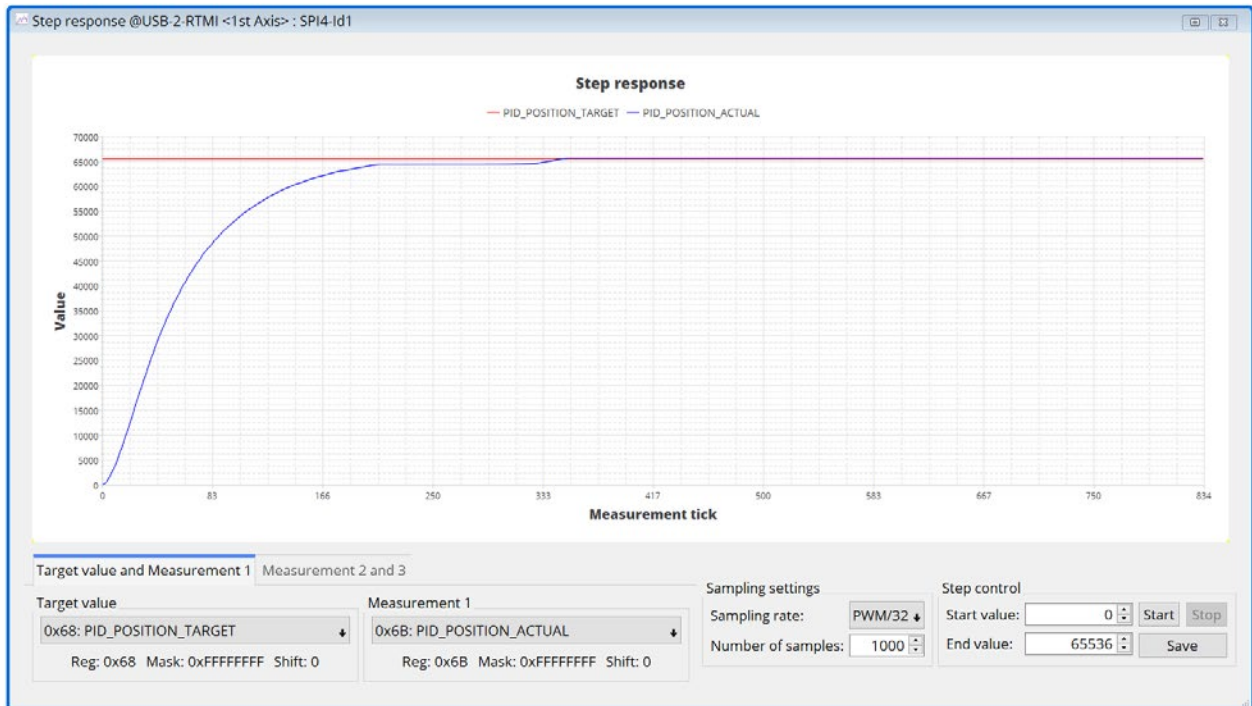


Abbildung 43: Erreichen des stationären Endwerts von 65.536 mit `PID_POSITION_P = 100`

Zusammenfassend sind alle optimierten Reglerparameter aus dem *PI control* Fenster und die eingestellten Limits in Abbildung 44 aufgeführt. Die Reglerparameter können über die Schaltfläche Export abgespeichert und daraufhin bei einem Neustart des Systems anhand der Schaltfläche *Import* hochgeladen werden.

Zusammenfassung der ermittelten und optimierten Reglerparameter und Limit-Einstellungen

Stromregler →

Geschwindigkeitsregler →

Positionsregler →

Eingestellte Limits

Adr	Name	Value
0x56	PID_TORQUE_I	2 500
	PID_TORQUE_P	1 200
0x54	PID_FLUX_I	3 800
	PID_FLUX_P	1 800

Adr	Name	Value
0x58	PID_VELOCITY_I	670
	PID_VELOCITY_P	800

Adr	Name	Value
0x5A	PID_POSITION_I	0
	PID_POSITION_P	100

Adr	Name	Value
0x5D	PIDOUT_UQ_UD_LIMITS	32 767
0x5E	PID_TORQUE_FLUX_LIMITS	10 609
0x5F	PID_ACCELERATION_LIMIT	2 000
0x60	PID_VELOCITY_LIMIT	4 000
0x61	PID_POSITION_LIMIT_LOW	-2 147 483 647
0x62	PID_POSITION_LIMIT_HIGH	2 147 483 647

Abbildung 44: Optimierte Reglerparameter und eingestellte Limits

**[1] TRINAMIC Motion Control GmbH & CO. KG, AN53: TMC4671 PI Tuning
Benutzerhandbuch V1.3., 17.12.2020**

https://www.trinamic.com/fileadmin/assets/Support/AppNotes/AN053_TMC4671-PI_Tuning.pdf

Downloads:

TMC4671-LA datasheet:

https://shop.mev-elektronik.com/wp-content/uploads/TMC4671-LA_datasheet.pdf

TMC4671-ES datasheet:

https://shop.mev-elektronik.com/wp-content/uploads/TMC4671-ES_datasheet.pdf

AN 005u: PCB Footprint Guidelines QFN76:

https://shop.mev-elektronik.com/wp-content/uploads/AN005u_IC_Package_PCB_Footprint_Guidelines_QFN76_rev1.00.pdf

AN025: DC Motor Control with TMC4671:

https://shop.mev-elektronik.com/wp-content/uploads/AN025_DC_Motor_Control_with_TMC4671.pdf

AN053:_TMC4671-PI_Tuning:

https://shop.mev-elektronik.com/wp-content/uploads/AN053_TMC4671-PI_Tuning.pdf

AN 064: How to Run a Linear Motor With TMC4671:

https://shop.mev-elektronik.com/wp-content/uploads/AN064_How_to_run_a_LinearMotor_with_TMC4671.pdf

Zugehörige Videos:

TMC4671 Disc Demo:

<https://www.youtube.com/watch?v=-hhFyXe7TFc>

Trinamic TMC4671 Servo Controller with Field Oriented Control / FOC:

<https://www.youtube.com/watch?v=-hhFyXe7TFc>

TMC4671 Product Training Module

<https://www.youtube.com/watch?v=DjoWdWBKudw>

How to set up the TMC4671 FOC Servo Controller:

<https://www.youtube.com/watch?v=g2BHEdvW9bU>

What is FOC? (Field Oriented Control) and why you should use it!

<https://www.youtube.com/watch?v=Nhy6g9wGHow>