

Betaflight BF4.x Tuning-Parameters



Inhaltsverzeichnis

- [Betaflight BF4.x Tuning-Parameters](#)
- [Inhaltsverzeichnis](#)
- [Historie](#)
- [Tuning-Parameter](#)
 - [Betaflight - Tuning-Tips](#)
- [DSHOT RPM Telemetrie-Daten](#)
 - [Allgemeines](#)
 - [IN](#)
 - [OUT](#)
 - [Parameter](#)
- [Gyro Filter](#)
 - [Allgemeines](#)
 - [IN](#)
 - [OUT](#)
 - [Gyro Filterarten](#)
 - [IN](#)
 - [OUT](#)
 - [DTerm => Dynamic D lowpass](#)
 - [Allgemein](#)
 - [Parameter](#)
 - [DTerm => Static D notch](#)
 - [AllgemeiHz](#)
 - [Parameter](#)
- [Feedforward](#)
 - [Allgemeines](#)
 - [IN](#)
 - [OUT](#)
 - [Parameter](#)
- [VBat](#)
 - [Allgemeines](#)
 - [IN](#)
 - [OUT](#)
 - [Parameter](#)
- [RC-Command](#)
 - [Allgemeines](#)
 - [IN](#)
 - [OUT](#)
 - [Parameter](#)
 - [rc_smoothing_auto_smoothness \(Default:10\)](#)
- [Setpoint](#)
 - [Allgemeines](#)

- [IN](#)
- [OUT](#)
- [Parameter](#)
- [ITerm Parameter](#)
 - [Allgemeines](#)

Historie

Version	Datum	Inhalt
0.1	August 2020	initial
0.2	August 2020	neu strukturiert

Tuning-Parameter

Nachfolgende eine Reihe der wichtigsten Tuning-Variablen.

Viel mehr Details findet man hier:

[BF4.2-Tuning-Notes](#)

ACHTUNG

Bei einem Update von BF <4.2 bitte **KEIN** Restore von alten Werten die durch `diff all` gespeichert wurden, importieren. Fangt bei **NULL** an

Betaflight - Tuning-Tips

Weitere Tuning-Tips findest du im BF-Wiki der jeweiligen Versionen:

- [BF4.2-Tuning-Notes](#)
- [BF4.1-Tuning-Notes](#)
- [BF4.0-Tuning-Notes](#)
- [BF4.0-Tuning-Notes](#)

DSHOT RPM Telemetrie-Daten

Allgemeines

Ab BF 4.0 werden Telemetriedaten des ESCs ausgelesen und analysiert. Diese Informationen sind elementar wichtig für viele nachgelagerte Filtertechniken und für den PID-Controller. Voraussetzung ist, dass man für den ESC ein DSHOT-Protokoll ausgewählt hat

IN

ESC -

Beachten:

die aktuelle Firmware des ESCs muss RPM-Telemetrie-Daten verarbeiten können.

OUT

RPM-Daten pro Motor

Parameter

Diese Parameter können nicht direkt beeinflusst werden. Zu beachten sind welches **DSHOT** Protokoll verwendet wird.

Bedenke: bei DSHOT300 und einer 8k PIDLoop erhältst du nur jede zweite PID-Loop Daten zugesendet. Daraus folgt, du solltest das passende **DSHOT-Protokoll** auf Deine PID-Loop auswählen

- **DSHOT150:** empfohlen bei 2k PIDLoop
- **DSHOT300:** empfohlen bei 4k PIDLoop
- **DSHOT600:** empfohlen bei 8k PIDLoop
- **DSHOT1200:** 8k PIDLoop

Gyro Filter

Allgemeines

Der Gyro ist das zentrale Bauelement auf dem FC und stellt die aktuellen **IST** Flugdaten zur Verfügung. Diese Daten werden dann bezogen auf die **SOLL** Daten (Die RC-Commands) verrechnet, gefiltert dem PIDController zur Verfügung gestellt. Anschließend gemixt und den Motoren als neue.

Jeder GYRO besitzt werkseitig einen internen LowPass-Filter. Je nach Bautyp des Gyros unterscheiden sich wie gut dieser interne Filter tatsächlich ist.

Die Gyro-Filter Parameter umfassen folgende Filterarten

Bei den weiteren beschriebenen Gyro Filtern wird nicht nochmals auf IN/OUT eingegangen.

IN

gyro_scaled Daten direkt aus dem Gyro.

OUT

Bereitstellung der Daten für nachgelagerte **DTerm-Filter** und als Mix-Daten für den **P-Controller** => **Vorab-Fehler P-Berechnung**

Gyro Filterarten

Es wird unterschieden zwischen den Gyro-RPM Filtern, einem Dynamic NotchFilter und Static Notchfilter 1+2, einem Dynamische LowPassfilter und einem statischen Lowpassfilter.

- Gyro-RPM-Notch Filter

```
gyro_rpm_notch_harmonics=3
gyro_rpm_notch_q=500
gyro_rpm_notch_min=100
```

- Dynamic-Notch Filter

```
dyn_notch_min_hz
dyn_notch_max_hz
dyn_notch_width_percent
dyn_notch_q
```

- Static Gyro-Notch Filter (1 und 2)

```
gyro_notch1_hz
gyro_notch1_cutoff
gyro_notch2_hz
gyro_notch2_cutoff
```

- Dynamic Gyro LowPass Filter

```
gyro_lowpass_type
gyro_lowpass_hz
dyn_lpf_gyro_min_hz
dyn_lpf_gyro_max_hz
```

- Static Gyro LowPass Filter

```
gyro_lowpass2_type
gyro_lowpass2_hz
```

Gyro Filter => GYRO-RPM Notch Filter

Allgemeines

Der Gyro-RPM Notch Filter nutzt die vom ESC zurückgegeben RPM-Daten und liegt als erste Filterstufe direkt hinter dem `gyro_scaled` Daten.

[BF 4.1/4.2 Bidirectional DSHOT and RPM Filter Guide](#)

Parameter

Parameter	BF	Default	Bezeichnung
<code>gyro_rpm_notch_harmonics</code>		3	<p>Schwingungen treten in wiederkehrenden Amplituden auf. Eine harmonische Schwingung kann durch eine Sinusfunktion beschrieben werden (https://de.wikipedia.org/wiki/Schwingung#Harmonische_Schwingung).</p> <p>Das bedeutet, dass eine Vibration sich alle xHz wiederholt!</p> <p>Betaflight generiert somit pro Motor 3 (Anzahl Harmonics) Notch-Filter, somit werden alle Motordaten durch 12 individuellen RPM-Notch-Filter analysiert und schon vorgefiltert. Diesen Sachverhalt kann man in einer Blackboxauswertung sehr gut sehen (FFT-Spektrogramm).</p>
<code>gyro_rpm_notch_q</code>		500	<p>Der Q-Faktor des Notchfilters gibt die Breite der Kerbe (Notch=Kerbe) an. Je größer die Zahl (max 1000) umso schmaler wird der Notchfilter. Der Q-Faktor wird auch als Güte-Faktor bezeichnet. Je höher die Güte-Faktor (Q-Faktor) desto geringer die Dämpfung, desto schmaler der Notch-Filter. Kleine Q-Faktoren vergrößern den RPM-Filter Delay - was unerwünscht ist</p>

Parameter	BF	Default	Bezeichnung
<code>gyro_rpm_notch_min</code>		100	Beschreibt die untere Grenzfrequenz des Notch-Filters

Gyro Filter => Dynamic-Notch Filter

Allgemeines

Dieser Filter ist dem RPM-Filter nachgelagert und filtern nochmals bestimmte Frequenzen aus. Ohne RPM-Filter wird der Filter als Doppel-Notch Filter betrieben, dies wird über `dyn_notch_width_percent` > 0 definiert. Ist dieser Wert 0, wird nur ein Notch-Filter erzeugt.

Das besondere an dynamic-notch-Filter ist, dass sie dynamisch sich an der aktuellen *RPM* des Systems orientieren und so laufend rund um den höchsten Frequenzbereich der Motor-Vibrationen arbeiten.

Grundsätzlich gilt, dynamische Notch-Filter haben eine geringere Latenzzeit als statische Notch-Filter. Bei einem gut abgestimmten Copter können andere Low-Pass Filter deaktiviert werden.

Parameter

Parameter	BF	Default	Bezeichnung
<code>dyn_notch_min_hz</code>			Beschreibt die untere Grenzfrequenz dieses Notch-Filters in Hz
<code>dyn_notch_max_hz</code>			Beschreibt die obere Grenzfrequenz dieses Notch-Filters in Hz
<code>dyn_notch_width_percent</code>			Beschreibt (wenn > 0), wie weit beide Notch-Filter voneinander getrennt sind. Der Prozentsatz berechnet sich aus der Breite des Notch-Filters.
<code>dyn_notch_q</code>			Q-Faktor des Notch-Filters. (siehe hierzu Beschreibung weiter oben)

Gyro Filter => Static Gyro-Notch Filter (1 und 2)

Allgemeines

Zwei statische Notch-Filter für ein bestimmtes Frequenzband. Dieses Frequenzband wird während des Fluges nicht mehr angepasst (statisch).

Parameter

Parameter	BF	Default	Bezeichnung
<code>gyro_notch1_hz</code>			Center-Frequenz des Notch-Filters
<code>gyro_notch1_cutoff</code>		todo	
<code>gyro_notch2_hz</code>			Center-Frequenz des Notch-Filters
<code>gyro_notch2_cutoff</code>		todo	

Gyro Filter => Dynamic Gyro LowPass Filter

Allgemeines

Die Auswahl den Dyn-Notchfilters Frequenzbereiches kann über drei Auswahlmöglichkeiten voreingestellt werden

- **LOW** : `dyn_lpf_gyro_max_hz` liegt bei 334hz oder ist 0 (deaktiviert)

- **MEDIUM** : `dyn_lpf_gyro_max_hz` liegt bei 610hz
- **HIGH** : `dyn_lpf_gyro_max_hz` liegt bei > 610hz

Die durchschnittlichen Werte für optimale Werte für diese Ranges liegen

- **LOW** : 80-330hz (für Copter mit niedrigen Drehzahlen oder wenn Resonanzen in niedrigen Frequenzen auftreten)
- **MEDIUM** : 140-550hz (für gut eingestellte 5" Copter)
- **HIGH** : 230-800hz (für Copter mit hohen Drehzahlen 2,5" - 3")

Ab BF 4.0 wird zusätzlicher `dyn_notch_min_hz` Parameter zur Verfügung gestellt. Dieser Wert fängt den Bereich unterhalb des Dyn-LPF ab und hat seinen Default bei 150Hz.

Um 100Hz Peaks heraus zu filtern muss **LOW** aktiviert werden und der

Parameter

Parameter	BF	Default	Bezeichnung
<code>dyn_notch_min_hz</code>			
<code>gyro_lowpass_type</code>			LOW/MEDIUM/HIGH (siehe Beschreibung)
<code>gyro_lowpass_hz</code>			Static Gyro LPF, sind <code>dyn_lpf</code> gesetzt, dann ist der Static LPF deaktiviert
<code>dyn_lpf_gyro_min_hz</code>			untere Grenzfrequenz es DynLPF
<code>dyn_lpf_gyro_max_hz</code>			obere Grenzfrequenz des DynLPF

Gyro Filter => Static Gyro LowPass Filter

Allgemeines

IN

Throttle-Daten

Parameter

Parameter	BF	Default	Bezeichnung
<code>gyro_lowpass2_type</code>	4.0	PT1/BIQUAD	
<code>gyro_lowpass2_hz</code>	4.0		unter Grenzfrequenz des LPF in Hz, wenn auf 0, dann ist der LPF deaktiviert

DTerm Filter

Allgemeines

Der DTerm-Filter besitzt eine Reihe von Parametern die dazu genutzt werden, das DTerm-Ausgangssignal zu bearbeiten und von Störungen (Vibrations-Frequenzen zu befreien). **Wichtig::** Der DTerm des PID-Controllers verstärkt Vibrationen, daher ist es wichtig, dass dieses Signal möglichst frei von Störungs- / Vibrationssignalen ist.

D verstärkt höhere Frequenzen, der D-Anteil wird aber dringend benötigt um Vibrationen zwischen 30-80hz (Z.B. Propwash) auszugleichen. Das bedeutet wir benötigen soviel wie möglich D-Anteil bis 100hz und so wenig wie möglich über 100hz. DTerm-Filter sollten immer in der ersten Stufe als BIQUAD und in der zweiten Stufe als PT eingestellt werden.

DTerm Filter Daten sind zeitabhängig (d/dt)

Folgende DTerm-Filter werden genutzt:

- **Dynamic D lowpass**

```
dterm_lowpass_type
dyn_lpf_dterm_min_hz
dyn_lpf_dterm_max_hz
dyn_lpf_dterm_curve_expo
```

- **Static D lowpass**

```
dterm_lowpass2_type
dterm_lowpass2_hz
```

- **Static D notch**

```
dterm_notch_hz
dterm_notch_cutoff
```

IN

Daten kommen aus den **Gyro Filter** Berechnungen

OUT

Daten gehen direkt an den **D-Controller**

DTerm => Dynamic D lowpass

Allgemein

Dynamischer Lowpass filter für den DTerm

Parameter

Parameter	BF	Default	Bezeichnung
dterm_lowpass_type	4.0		PT1 / BIQUAD, sollte auf BIQUAD für LPF 1 stehen, LPF2 = PT1
dyn_lpf_dterm_min_hz	4.0		LowPass min Hz
dterm_lowpass2_hz	4.0		LowPass2 min Hz

DTerm => Static D notch

AllgemeinHz

Parameter

Parameter	BF	Default	Bezeichnung
dterm_notch_hz			

Parameter	BF	Default	Bezeichnung
<code>dterm_notch_cutoff</code>			

Feedforward

Allgemeines

Feedforward ist dem PID-Controller nachgelagert und unabhängig vom PID. FF verstärkt bzw. wirkt auf Deine Stickbewegung und hilft den Motoren schneller zu reagieren.

Mehr Infos zu Feedforward

- [Feedforward BF 4.1](#)
- [Feedforward 2.0 BF 4.2](#)

IN

OUT

Parameter

Parameter	BF	Default	Bezeichnung
<code>ff_boost</code>	4.1	15	Der Booster verstärkt den gesamten FF-Wert aber zu einem sehr frühen Zeitpunkt und verringert damit ein Delay
<code>ff_interpolate_sp</code>	4.1	2 (Average_2)	siehe Anhang der Tabelle
<code>ff_spike_limit</code>	4.1	50	Es wird eine effiziente verzögerungsfreie Dämpfungsmethode verwendet, die Erhöhung des boosts durch Spikes verringert bzw. vermeidet. Liegt der normale Boost-Wert unter dem limit wird er durchgelassen, alles weitere, hohe Boosts die durch Spikes verursacht werden, werden gedämpft.
<code>ff_max_rate_limit</code>	4.1	100	<code>ff_max_rate_limit</code> unterbricht den Feedforward, wenn die Geschwindigkeit mit dem der Stick bewegt wird wahrscheinlich sein Ende des mechanischen Verfahrensweges erreicht. Dadurch wird ein Überspringen gerade bei Beginn von Flips reduziert.
<code>ff_smooth_factor</code>	4.2	37	Glättungsfaktor für einkommende Signale. Funktioniert wie ein LowPassfilter. 0 = keine Glättung, höhere Werte wie der Defaultwert, erhöhen auch die Latenzzeit und wirkt dem eigentlich FF-Forward entgegen

**** `ff_interpolate_sp` Ausprägung****

- OFF
- ON
- AVERAGE-2 : passt für die meisten Copter & Freestyler
- AVERAGE-3 :
- AVERAGE-4 :

VBat

Allgemeines

Ab BF 4.2 wir mit VBAT-SAG-Kompensation

[Weitere Informationen:](#)

IN

OUT

Parameter

Parameter	BF	Default	Bezeichnung
vbat_sag_compensation	4.2	100	100% Kompensation der Batterieentladung
vbat_pid_gain	?	OFF	alte Version sollte immer OFF sein

RC-Command

Allgemeines

IN

Receiver-Daten Signal

OUT

Daten werden mit den eingestellten **Rates** verrechnet und gelten dann als das angewendete **RCCommand-Eingangs-Signal**

Parameter

Parameter	BF	Default	Bezeichnung
rc_interpolation			
rc_interp			
rc_inter_ch			
rc_inter_int			

rc_smoothing_auto_smoothness (Default:10)

rc-smoothing-auto-smoothness setzt wie glatt der die RC-Signale sein sollen.

Größere Werte erhöhen die Glättung vergrößern aber das RC-Delay. 10 ist optimal für die meisten allgemeinen Flüge. Racer bevorzugen 8 oder sogar 5. Das RC-Delay nimmt zwar ab, dafür können die Motor-Signale etwas unruhiger werden.

Setpoint

Allgemeines

In der weiteren Verarbeitung der Eingangssignale werden diese als **Setpoint** bezeichnet und spiegeln das RC-Signal wieder allerdings durch eine Reihe von Parametern geglättet

IN

Aufbereitetes RC-Command Signal

OUT

Daten die mittels **setpoint_smoothing** nochmals geglättet werden werden an

- **Vorab-Fehler P-Berechnung**
- **d/dt**

Parameter

Setpoint/Setpoint smoothing beinhaltet eine Reihe von Parametern die das eigentlich Signal nochmals aufbereiten.

Parameter	BF	Default	Bezeichnung
rc_smoothing_type			
rc_smoothing_auto_smoothness		10	setzt wie glatt der die RC-Signale sein sollen. Größere Werte erhöhen die Glättung vergrößern aber das RC-Delay. 10 ist optimal für die meisten allgemeinen Flüge. Racer bevorzugen 8 oder sogar 5. Das RC-Delay nimmt zwar ab, dafür können die Motor-Signale etwas unruhiger werden
rc_smoothing_input_Hz			
rc_smoothing_input_type			

Iterm Parameter

Allgemeines

Iterm Parameter dienen dazu das I-Signal die PID-Controllers entweder vor der Bearbeitung von Iterm oder nach Iterm zu beeinflussen.

Insbesondere sollen hier Peaks im Iterm eliminiert werden.

Die Nachfolgende Tabelle beinhaltet zwei zusätzliche Spalten **IN** und **OUT** sie bezeichnen woher die Daten kommen (**IN**) und wer sie verwertet (**OUT**)

Parameter	BF	IN	OUT	Default	Bezeichnung
item_windup		MIXER	Iterm		item_windup ist eine alte Methode zur Unterdrückung der iTerm-Akkumulation, wenn das Motordifferential einen benutzerdefinierten Schwellenwert überschreitet. In BF 4.2 wirkt item_windup nur noch auf YAW
item_relax		ITERM	PID_SUM		item_relax hat item_windup zur Vermeidung von I-Anhäufungen auf Mini-Quads weitgehend ersetzt. item_relax ist hauptsächlich zur Vermeidung von bounce-backs
item_relax_type		ITERM	PID_SUM		

Parameter	BF	IN	OUT	Default	Bezeichnung
<code>item_relax_cutoff</code>		ITERM	PID_SUM		<p>Wenn der Pilot eine Änderung der Drehgeschwindigkeit anfordert, die für das Quad zu schnell ist, eilt das Gyrosignal dem Sollwert (SetPoint) hinterher und daraus entsteht ein mehr oder minder großes Fehlersignal. Der I-Term versucht nun diesen Fehler zu akkumulieren (aufsummieren) und versucht diese zu korrigieren. <code>item_relax</code> versucht nun diese Akkumulation zu kontrollieren. Reicht der <code>item_relax</code> nicht aus, sammeln sich die ITerm Fehler immer mehr an. Stoppt nun der Pilot seine Stickbewegung (z.B. in einem Flip), dann wird all der angesammelte ITerm-Fehler eine Gegenbewegung des Copters verursachen (BounceBack), der dann langsam ausklingen wird bis er wieder auf 0 ist. <code>item_relax</code> für Flip & Rolls und <code>item_windup</code> für YAW, versuchen diese Bouncebacks zu kontrollieren und abzumildern. <code>item_relax_cutoff</code> begrenzt die ITermAkkumulation.</p>
<code>item_rotation</code>		GYRO-Filter	ITerm		