

Mario Weidner  
Papstdorfer Straße 55, 01277 Dresden  
[www.mario-weidner.de](http://www.mario-weidner.de)

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

*Am Beispiel des Reglerdesigns von Alex Khoroshko*

# INHALTSVERZEICHNIS

## Inhalt

Multikopter	1
Reglertypen	2
Reglerdesign	3
PID-Regler	4
Analyse des Reglerdesigns der MultiWii-Software	7
Berechnung der Reglerparameter der MultiWii	11

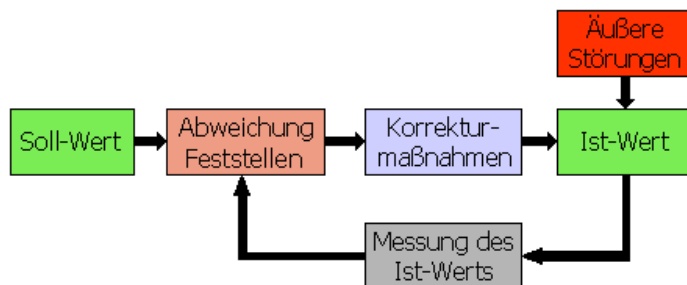
## Multikopter

### FLUGLAGEREGELUNG EINES QUADROKOPTERS

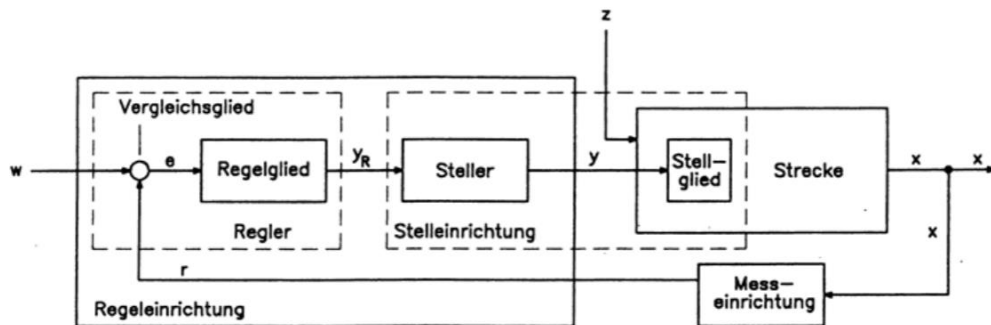
Da ein Quadrokofter von selbst keine stabile Fluglage herstellen kann wie ein Flächenmodell muss die Fluglage aktiv, d.h. durch die Rotoren erreicht werden. Dazu wird ein digitaler Flugcontroller (Flightcontroller, FC) benutzt, der die entsprechenden Daten an die Motoransteuerung (Brushless Controller, Speed Controller, ESC) sendet.

Auf dem FC befinden sich Sensoren, die die Fluglage (ACC) und die Beschleunigungen (GYRO) in den 3 Achsen x,y,z (Roll, Pitch und Yaw) und damit die Abweichung von einem gewünschten Wert ermitteln.

Die Regelung (Software) errechnet daraus die entsprechenden Steuerbefehle, für die Speed Controller.



Der Software-Regler (Controller) bekommt die Sollvorgabe  $w$  über die Fernbedienung übermittelt, berechnet die Abweichung  $e$  und wirkt dabei mit  $y$  auf die Regelstrecke (den Quadrokofter) ein.



Die Regelung ist als so genannter PID-Regler ausgeführt, der im Folgenden betrachtet wird.

Begriffe nach DIN 19226 Teil 2:

$w$  = Führungsgröße  
 $x$  = Regelgröße  
 $e$  = Regelabweichung  
 $y$  = Stellgröße

## Reglertypen

### PROPORTIONAL-REGLER

Der Ausgang des P-Reglers ist der Regelabweichung direkt proportional über einen Faktor  $K_p$ .

$$y(t) = K_p * e(t)$$

Vorteil: schnelle Reaktion

Nachteil: es bleibt immer eine Regelabweichung bestehen.

### INTEGRAL-REGLER

Der Integralregler summiert die Regelabweichung über die Zeit fortwährend auf. Die Summe wird über den Parameter  $K_i$  gewichtet.

$$y(t) = K_i * \int_0^t e(T) dT$$

Vorteil: keine bleibende Regelabweichung

Nachteil: wirkt verzögert/langsam

### DIFFERENZIAL-REGLER

Der Differentialregler ermittelt die Änderung der Regelabweichung, die mit dem Parameter  $K_d$  gewichtet wird.

$$y(t) = K_d * \frac{de}{dt}$$

Vorteil: wirkt sehr schnell und vorherschauend

Nachteil: bleibende Regelabweichung und erkennt keine langsamen Regelabweichungen

### PID-REGLER

Der PID-Regler kombiniert die Vorteile der genannten Regler.

## Reglerdesign im Flugcontroller

In der MultiWii-Software sind zwei alternative PID-Regler implementiert. Im Folgenden wird das Reglerdesign nach AlexK betrachtet.

Aus ([www.rclineforum.de](http://www.rclineforum.de))

*Alex Khoroshko ist ein russischer Uni-Professor, der für ein Projekt in Sachen Steuerungstechnik auf MultiWii gestoßen ist. Bei seiner Analyse des Quellcodes kam er zum Ergebnis, dass die Regelungsschleife (PID-Controller) bei MultiWii eher suboptimal (oder falsch) implementiert ist. Dies führt dazu, dass der Copter ein gewisses Eigenleben hat und nicht unbedingt den Stick-Kommandos folgt. Er hat deshalb den PID-Controller völlig neu programmiert.*

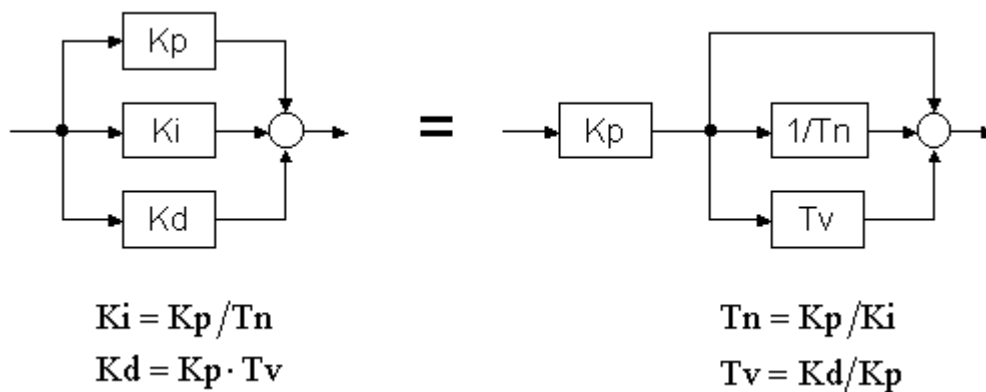
*Alex Khoroshko wurde von den Entwicklern von MultiWii ziemlich übel behandelt; die fühlten sich wohl vorgeführt. Er zog sich deshalb aus dem Projekt zurück ... schade, denn eigentlich hat jeder, der seine Implementierung getestet hat, dies für einen Quantensprung nach vorn gehalten. Die Entwickler von MWii haben sich dann entschlossen, den PID\_CONTROLLER 2 zwar im Quelltext zu belassen, aber für "unsupported" zu erklären.*

**Das PID-Reglerdesign nach AlexK fand seinen Weg in die aus dem MultiWii-Projekt abgeleiteten Projekte Baseflight und Cleanflight.**

## PID-Regler

Der PID-Regler besteht also aus einem P- einem I- und einem D-Regler.

### ERSATZDARSTELLUNG



Für den (analogen) PID-Regler gilt:

$$y(t) = K_p * e(t) + K_i * \int e(t) dt + K_d * \frac{de}{dt}$$

mit

$$K_i = \frac{K_p}{T_n} \text{ und } K_d = K_p * T_v$$

$K_p$  = Proportionale Verstärkung

$K_i$  = Integrierbeiwert

$K_d$  = Differenzierbeiwert

$T_v$  = Vorhaltzeit

$T_n$  = Nachstellzeit

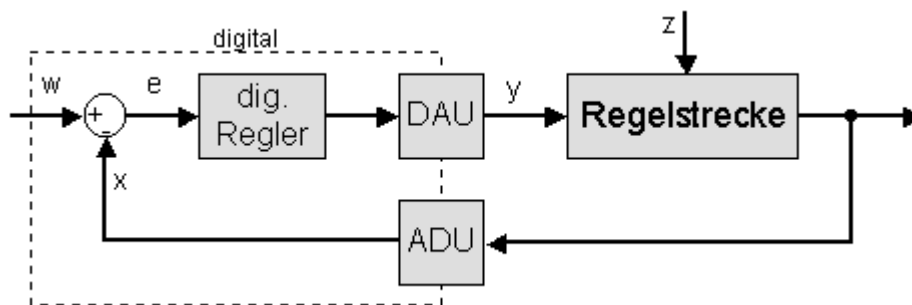
**Um den Regler zu dimensionieren und damit an die Regelstrecke anzupassen müssen die Parameter  $K_p$ ,  $K_i$ ,  $K_d$  bzw  $K_p$ ,  $T_n$ ,  $T_v$  bestimmt werden.**

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

## ANALOG VS. DIGITAL

Die genannten mathematischen Formeln gelten für analoge Regler, die z.B. mit Operationsverstärkern realisiert werden/wurden.

Durch die Realisierung mit Software auf einem Mikrokontroller gibt es diskrete Abtastzeitpunkte, an denen die Istwerte ermittelt und in digitale Werte gewandelt werden. Die Berechnung der Stellgröße und Wandlung in analoge Werte dauert eine bestimmte Zeit. Die Software durchläuft die Eingabe/Berechnung/Ausgabe in einer Schleife mit einer endlich kleinen Zykluszeit  $T_0$ .



Die Werte gelten also immer für einen Zyklus  $k$  und benutzen Werte aus den vorhergegangenen Zyklen  $k-1, k-2, \dots$

Für den Proportionalen Anteil des Reglers heisst das:

$$y(t) = K_p * e(t) \implies y(k) = K_p * e(k)$$

Für den Integralanteil ergibt sich die Summierung aus den jeweiligen Regelabweichungen der Zeitpunkte  $0 \dots k$  und der Zykluszeit  $T_0$ :

$$y(t) = K_i * \int_0^t e(T) dT \implies y(k) = K_i * (e(0) * T_0 + e(1) * T_0 + e(2) * T_0 + \dots + e(k-1) * T_0)$$

Und damit:

$$y(k) = K_i * T_0 * [e(0) + e(1) + e(2) + \dots + e(k-2) + e(k-1)]$$

bzw. durch  $K_i = K_p / T_n$ :

$$y(k) = \frac{K_p * T_0}{T_n} * [e(0) + e(1) + e(2) + \dots + e(k-2) + e(k-1)]$$

Für den D-Anteil erhalten wir:

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

$$y(t) = Kd * \frac{de}{dt} \implies y(k) = Kd * \frac{e(k) - e(k-1)}{T_0}$$

bzw. durch  $Kd = Kp * Tv$ :

$$y(k) = \frac{Kp * Tv}{T_0} * [e(k) - e(k-1)]$$

## REKURSIVER ANSATZ

Wenn man sich die Formel für  $y(k)$  anschaut, sieht man, dass für den I-Anteil über alle eingelesenen Werte der Regeldifferenz summiert wird. Dazu müsste man jeden einzelnen Wert für die Neuberechnung abspeichern. Einfacher ist es, auf den zuvor berechneten Wert  $y(k-1)$  der Stellgröße zurückzugreifen. Man spricht hier von einem rekursiven Algorithmus.

$$y(k) = \frac{Kp * T_0}{Tn} * [e(0) + e(1) + e(2) + \dots + e(k-2) + e(k-1)]$$

$$y(k-1) = \frac{Kp * T_0}{Tn} * [e(0) + e(1) + e(2) + \dots + e(k-2)]$$

Durch Differenzbildung fällt eine Menge weg:

$$y(k) - y(k-1) = \frac{Kp * T_0}{Tn} * [e(k-1)]$$

und es ergibt sich:

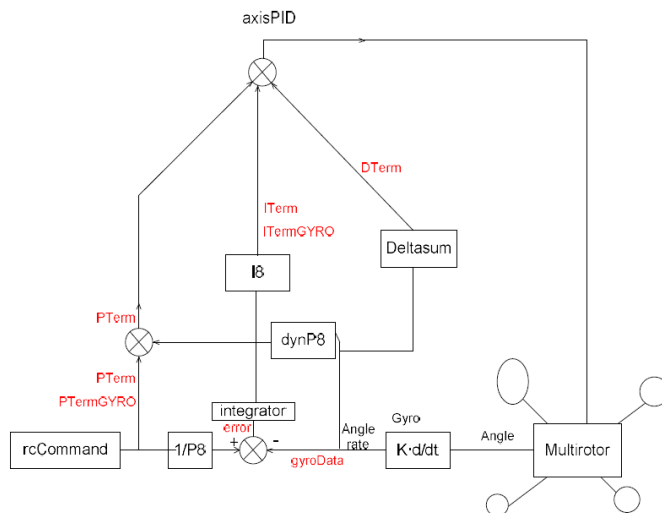
$$y(k) = y(k-1) + \frac{Kp * T_0}{Tn} * [e(k-1)]$$



# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

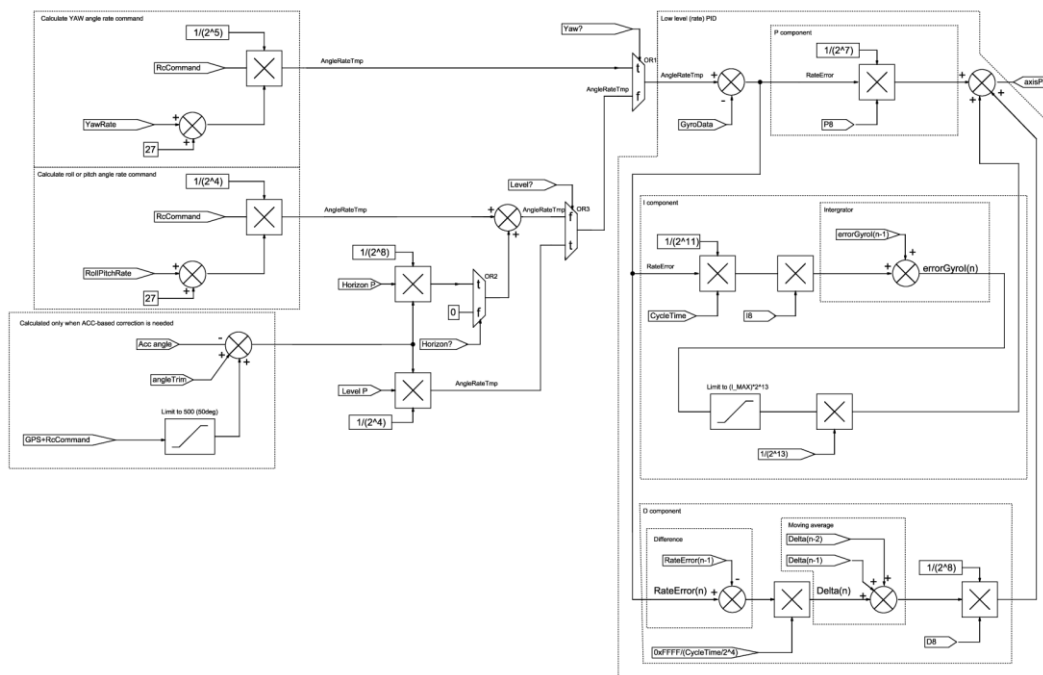
## Analyse des Reglerdesigns der MultiWii-Software

### REGLERTYP 1 (NICHT BETRACHTET)



Siehe <http://www.multiwii.com/forum/viewtopic.php?f=8&t=3671>

### REGLER NACH ALEXX

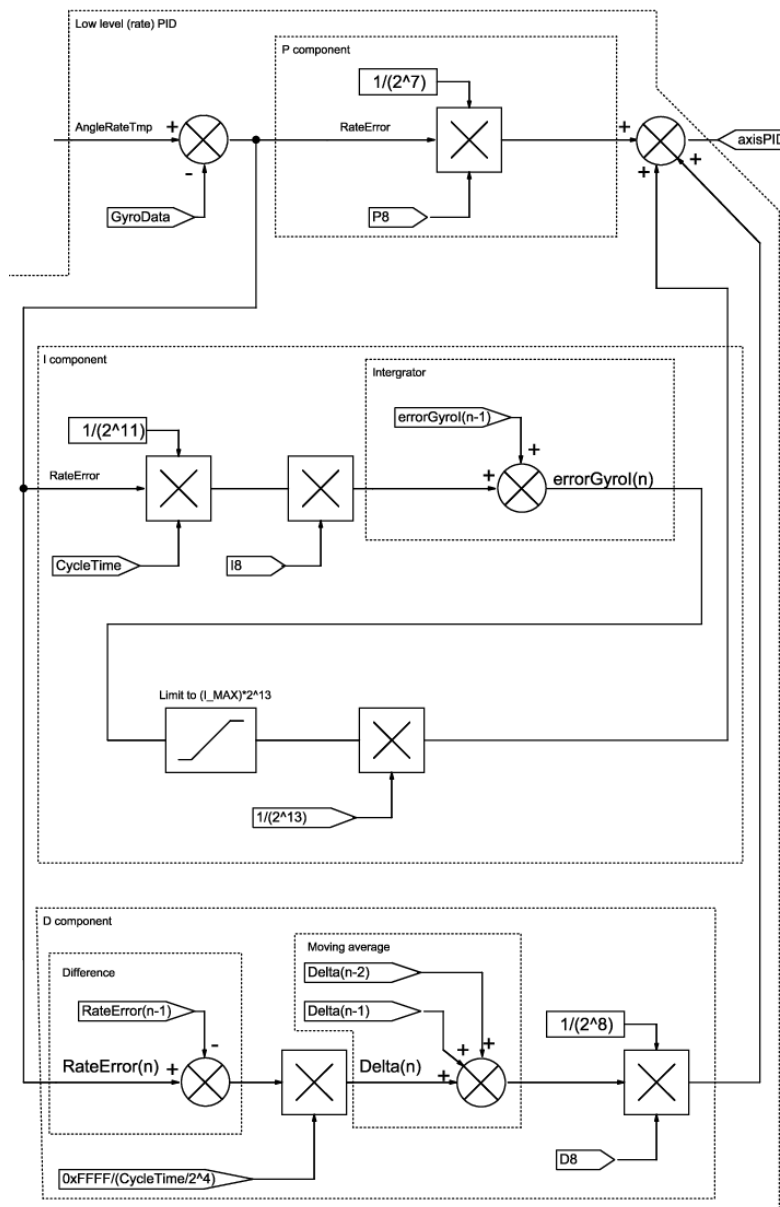


# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

## ABBILDUNG SCHEMATISCH

Die Führungsgröße  $w$  heisst hier *AngleRateTmp*. und *AxisPID* ist die Ausgabe der Stellgröße  $y=P+I+D$ .

Um die auftretenden Werte im Bereich der verwendeten Variablen zu belassen bzw. um schnelle Ganzzahlberechnungen nutzen zu können sind im Algorithmus einige Divisionen enthalten, die sich auf die einzugebenden Reglerparameter auswirken.



# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

## ABBILDUNG IM QUELLCODE

Der P-Anteil ist simpel:

```
// -----calculate P component
PTerm = (RateError * cfg.P8[axis]) >> 7;
```

Das entspricht der Formel:

$$y(k) = Kp * e(k) \text{ mit } Kp = \frac{P8}{2^7}$$

Wobei P8 der über die Konfigurationsoberfläche einzugebende P-Faktor ist.

Der I-Anteil enthält zusätzlich einen Begrenzer.

```
// -----calculate I component
// Time correction (to avoid different I scaling for different builds based on average cycle
// time) is normalized to cycle time = 2048.
errorGyroI[axis] = errorGyroI[axis] + ((RateError * cycleTime) >> 11) * cfg.I8[axis];

errorGyroI[axis] = constrain(errorGyroI[axis], (int32_t)-GYRO_I_MAX << 13,
(int32_t)+GYRO_I_MAX << 13);
ITerm = errorGyroI[axis] >> 13;
```

Das entspricht der Formel:

$$y(k) = y(k-1) + \frac{Kp * T_0}{Tn} * [e(k-1)] \text{ mit } \frac{Kp * T_0}{Tn} = I8 * \frac{T_z}{2^{11}} * \frac{1}{2^{13}}$$

Mit  $T_z$  = Zykluszeit in  $\mu s$  also  $T_z = T_0 * 10^6$  und  $Kp = P8 / 2^7$  ergibt sich dann:

$$I8 = \frac{P8}{Tn} * \frac{2^{17}}{10^6}$$

Wobei I8 der über die Konfigurationsoberfläche einzugebende I-Faktor ist.

Der D-Anteil enthält zusätzlich eine Berechnung des gleitenden Durchschnitts über die 3 letzten Werte.

```
//-----calculate D-term
delta = RateError - lastError[axis]; // 16 bits is ok here, the dif between 2 consecutive
gyro reads is limited to 800
lastError[axis] = RateError;

// Correct difference by cycle time.
// would be scaled by different dt each time. Division by dT fixes that.
delta = (delta * ((uint16_t)0xFFFF / (cycleTime >> 4))) >> 6;

// add moving average here to reduce noise
deltaSum = delta1[axis] + delta2[axis] + delta;
```

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

```
delta2[axis] = delta1[axis];  
delta1[axis] = delta;  
DTerm = (deltaSum * cfg.D8[axis]) >> 8;
```

Das entspricht der Formel:

$$y(k) = \frac{K_p * T_v}{T_0} * [e(k) - e(k-1)] \text{ mit } \frac{K_p * T_v}{T_0} = D8 * \frac{1}{2^8} * 3 * \frac{1}{2^6} * \frac{2^{16}}{T_z^4}$$

ausmultipliziert ergibt sich:

$$K_p \frac{T_v}{T_0} = D8 * \frac{3 * 2^6}{T_z}$$

mit  $T_z$  = Zykluszeit in  $\mu s$  also  $T_z = T_0 * 10^6$  und  $K_p = P8 / 2^7$  ergibt sich dann:

$$D8 = P8 * T_v * \frac{10^6}{3 * 2^{13}}$$

Damit haben wir die Möglichkeit die in der Software zu verwendenden Parameter zu berechnen, wenn wir die theoretischen Parameter des PID-Regler kennen.

$$P8 = K_p * 2^7$$

$$I8 = \frac{P8}{T_n} * \frac{2^{17}}{10^6}$$

$$D8 = P8 * T_v * \frac{10^6}{3 * 2^{13}}$$

## Einstellung der Reglerparameter der MultiWii

### GÄNGIGE METHODEN IN DER PRAXIS

#### Dimensionierung durch Probieren (Empirisches Einstellen)

Man beginnt mit unkritischen Einstellungen. Die Parameter für den I- und D-Anteil werden auf Null gesetzt und der Proportionalanteil von kleinen Werten beginnend solange erhöht, bis das System anfängt zu schwingen. Dann wird der Wert etwas zurückgenommen, so dass die Schwingung aufhört. Danach erhöht man auf die gleiche Weise den Integralanteil und zum Schluss den D-Anteil.

#### Dimensionierung nach Einstellregeln

Zur Dimensionierung von Reglerparametern gibt es zahlreiche Einstellregeln. Die bekannteste ist die Methode des Stabilitätsrandes nach Ziegler-Nichols von 1942!!

### EINSTELLUNG NACH DER STABILITÄTSRANDMETHODE VON ZIEGLER-NICHOLS

Der Regelkreis wird mit Hilfe eines proportionalen Reglers geschlossen ( $K_i=0$ ;  $K_d=0$ ) und die Reglerverstärkung ( $K_p$ ) solange erhöht, bis der Ausgang des Regelkreises bei konstantem Eingang eine Dauerschwingung mit der Periode  $T_{krit}$  bei der Reglerverstärkung  $K_{pkrit}$  ausführt.

Aus der kritischen Verstärkung und der Periodendauer können die Reglerparameter bestimmt werden.

$$K_p = 0.6 K_{pkrit}$$

$$T_n = 0.5 T_{krit}$$

$$T_v = 1/8 * T_{krit}$$

### ERMITTLUNG DER PARAMETER

Die PID-Parameter des Quadropters für die Roll- und Pitch-Achse werden auf Null gesetzt.

Die weiteren Tests erfordern, dass der Kopter zwischen zwei Schnüren o.ä. horizontal entlang der Roll-Achse aufgehängt wird oder festgehalten wird (**Vorsicht Verletzungsgefahr!!**)

Der P-Parameter wird nun solange in kleinen Schritten erhöht, bis der Kopter anfängt, eine konstante Schwingung auszuführen. In diesem Zustand wird der P-Wert festgehalten →  $K_{pkrit}$

Die Periodendauer bzw. die Frequenz der Schwingung ist zu ermitteln z.B. über eine Videoaufnahme mit Auszählen der Schwingungen in einer bestimmten bekannten Zeitdauer →  $T_{krit} = 1/f_{krit}$ .

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

## UMRECHNUNG IN DIE REGLERPARAMETER DES FLUGCONTROLLERS

Die intern in der Software verwendeten Reglerparameter P8, I8 werden in der Konfigurationsoberfläche nicht direkt angezeigt sondern mit einem Faktor versehen. Es gilt daher:

$$P = P8 / 10$$

$$I = I8 / 1000$$

$$D = D8$$

Der kritische Wert für P kann wie beschrieben ermittelt werden. Damit ergibt sich  $K_{pkrit}$  als zehnfaches des ermittelten P-Wertes.  $T_{krit}$  wurde auch wie beschrieben ermittelt.

Damit ergibt sich nach Ziegler-Nichols mit  $K_p = 0.6 K_{pkrit}$ :

$$P8 = 0.6 * P_{pkrit} * 10 \text{ (angezeigt und eingegeben werden muss } P8/10 \text{ !!)}$$

Mit  $T_n = 1/2 T_{krit}$  ergibt sich:

$$I8 = \frac{P8}{T_{krit}} * \frac{2^{17}}{10^6} \text{ oder } I8 = P8 * f_{krit} * \frac{2^{17}}{10^6}$$

(angezeigt und eingegeben werden muss I8/1000 !!)

Mit  $T_V = 1/8 T_{krit} = T_{krit} / 2^3$  ergibt sich:

$$D8 = P8 * T_{krit} * \frac{10^6}{3 * 2^{16}} \text{ oder } D8 = \frac{P8}{f_{krit}} * \frac{10^6}{3 * 2^{16}}$$

Zusammengefasst ist in der Konfigurationsoberfläche direkt einzustellen:

$$P = 0.6 * P_{pkrit}$$

$$I = \frac{6 * P_{pkrit}}{T_{krit}} * \frac{2^{17}}{10^9} \text{ oder } I = 6 * P_{pkrit} * f_{krit} * \frac{2^{17}}{10^9}$$

$$D = 6 * P_{pkrit} * T_{krit} * \frac{10^6}{3 * 2^{16}} \text{ oder } D = \frac{6 * P_{pkrit}}{f_{krit}} * \frac{10^6}{3 * 2^{16}}$$

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

## VERGLEICH MIT WERTEN AUS DER PRAXIS

Für meinen Quadrokoopter mit  $P_{krit}=6$  und  $f_{krit}=3\text{Hz}$  berechnen sich die Parameter wie folgt:

$$P = 0.6 * P_{pkrit} = 3.6$$

$$I = 6 * P_{pkrit} * f_{krit} * \frac{2^{18}}{10^9} = 6 * 6 * 3 * \frac{2^{17}}{10^9} = 0.014$$

$$D = \frac{6 * P_{pkrit}}{f_{krit}} * \frac{10^6}{3 * 2^{16}} = \frac{6 * 6}{3} * \frac{10^6}{3 * 2^{16}} = 61$$

**Der Praxistest mit diesen Werten bewies sehr gute Flugeigenschaften!**

Die im Quellcode verankerten Vorgabewerte für die Einstellung des PID-Reglers sind übrigens wie folgt:

Software	P (Roll/Pitch)	I (Roll/Pitch)	D (Roll/Pitch)
MultiWii 3.2 pid_controller=2	2.8	0.010	70
Baseflight (pid_controller=1)	4	0.030	23
Cleanflight (pid_controller=1)	4	0.030	23

# REGLERPARAMETRISIERUNG FÜR EINEN QUADROKOPTER

## QUELLEN:

<http://staff.ltam.lu/feljc/school/asser t3/Digitalregler.pdf>

[http://www.kahlert.com/web/download/pid\\_einstellregeln.pdf](http://www.kahlert.com/web/download/pid_einstellregeln.pdf)

<http://de.wikipedia.org/w/index.php?title=Regler&redirect=no#PID-Regler>

<http://rn-wissen.de/wiki/index.php/Regelungstechnik>

## weiterführende Literatur:

Ziegler, J. G. / Nichols, N. B.: Optimum Settings for Automated Controllers. In: Transactions of the American Society of Mechanical Engineers, 1942. S. 759-768.