



AutoTune

- [\[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\]](#)
- [AutoTune \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\] Pre-Arm \[\]](#)
- [\[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] / \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\]](#)
- [automatic-tuning-with-autotune](#)
- [Save Trim / Auto Trim](#)
- [calibrating-an-airspeed-sensor](#)
- [\[\] \[\] \[\] \[\]](#)
- [IMU FFT](#)
- [IMU \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] \[\] \[\]](#)
- [\[\] \[\] \[\] \[\] Input Shaping \[\]](#)
- [navigation-tuning](#)
- [new-roll-and-pitch-tuning](#)
- [QuikTune](#)
- [roll-pitch-controller-tuning](#)

- [\[\] \[\] \[\] \[\]](#) _____
- [tecs-total-energy-control-system-for-speed-height-tuning-guide](#)
- [\[\] \[\] \[\] \[\]](#) Throttle Boost [\[\]](#) _____
- [\[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\]](#) _____
- [\[\] \[\] \[\] \[\]](#) _____



-
- RTL / Land
- / /
-
-



Arm
Disarm



-  Pre-Arm  / 
-  / 
- 
- 

AutoTune

AutoTune

[illegible]

☐ AutoTune

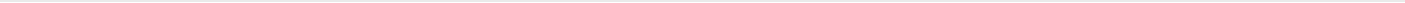
[illegible]

- 
-  AutoTune 
- 



□□□□ □□□□□□□□□□

- □□ /□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□
- GPS / □□□□ Home □□□□
- □□□□□□□□□□ RTL
- □□□□□□□□□□
- □□□□□□□□□□



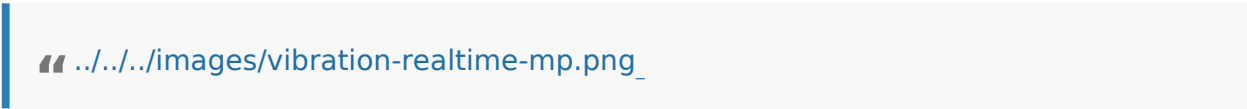


```
[copy  ? wiki  ="cop  , ,rover,dev,sub,planner"]
```

GPS
.
(
:



HUD "Vibe"



30m/s/s
. 30 /
60 /

Vibe

access-date=
() Vibe
30m/s/s

- ()
, **
- VIBE
X, Y Z
.
/
3DR IRIS
15m/s,
30m/s/s
30m/s/s()
- Clip0 Clip1 Clip2
,
(16G)
,
(< 100)
**

../../../../images/mp_vibe_dataflash_msg.png

../././images/mp_measuring_vibration_bad_vibes.png_

Malay AP .cpp calc clipping()

- IMU x y z
- 5hz, x,y z “accel vibe floor”
- accel vibe .
- , 2hz, (x y z) VIBE msg VibEX, Y Z

 , . 0, IMU

 , :

“ $X \oplus Y$, . /
 FC $X \text{ OR } Y$, FC
 FC
 FC Z , (bent label)
 ”















□ "□□" □

[illegible]

- `uint8_t` , `uint16_t`
- `uint8_t A[10]` `HHRS2.Roll`, `NKF1[0]`. `uint8_t` `NKF1[1]`. `uint8_t` `E[10]` `F2[10]` `AHRS2.Roll`, `XKF1[0]` (`uint8_t`). `uint8_t` `XKF1[1]`. `uint8_t` `EKF3[10]`

../../../../images/vibration-measuring-leans.png

IMU FFT

IMU FFT, ** "Raw IMU" **

IMU

Vibe ArduPilot IMU

- **LOG BITMA 1920K** IMU ,
- ()
- ** "a" "
- ()
- # 1 1.log
- IMU AccX AccY AccZ
- [DiagnosingWithLogs_Vibes](#)
- 3 3 -15 -5.
- ** **
- ()
- " " APM CPU ,



Pre-Arm

Pre-Arm Safety Checks

Pre-Arm Safety Checks

Pre-Arm Safety Checks

/HUD

- GPS /
- GPS
-
- SD

Pre-Arm Safety Checks



□□ □□□□□□□□□□□□□□
□□□□□□□

Copter □□□□□□

QuikTune□ AutoTune

- 1. □□□□□□□□□□□□□□
- 2. □□□□□ AutoTune □□□□
- 3. □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□



- 1. ☐ Pre-Arm ☐
- 2. ☐☐☐☐☐☐☐☐
- 3. ☐☐☐☐☐ AltHold / Loiter ☐☐☐☐☐
- 4. ☐☐☐☐☐☐☐☐
- 5. ☐☐☐☐ AutoTune ☐☐☐☐☐☐





Autoune Gramroll pitch Autoune

**ATC THR MI **MOT THST HOVER **

. 55.

P,D, 50% ,

- 1. 50%
- 2. D 10%
- 3. D 25%
- 4. ,P 50%
- 5. ,P 10%
- 6. P essa 25%

Each time the P term is changed set the I term equal to the P term. Those parameters can be changed on ground and preferably disarmed. A confident pilot could set them in flight with GCS or transmitter tuning knob (see Transmitter based tuning page for setup of this feature).

If using Transmitter based tuning, set the minimum value of the tuning range to the current safe value and the upper range to approximately 4 times the current value. Be careful not to move the slider before the parameter list is refreshed to recover the set value. Ensure the transmitter tuning is switched off before setting the parameter value or the tuning may immediately overwrite it.

Video of in-flight tuning

<https://youtu.be/NOQPrTdrQJM>

“	width:	100%
	url_parameters:	?start=145





Hover Throttle(
 "
). ##MOT THST # :
 , : :

**MOT THST Hover ** , flash , CTUN .
 THO . 0.2 0.6 , , .

, ** MOT HOVER LEARN ** 0.

[../images/throttle_mid_learning.png](#)





XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX,XXXXXX,XXXXXXXXXXXX,XX **ACROXX **
XXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXX,XXXXXXXXXXXXXXXXXXXXXXXXXXXX. XXXX (<
6XXXXXXXXXX)XXXXXXXXXX

XX,XXXXXXXXXXXXXXXXXXXX,XXX. XX ArduPilotXXXXXXXXXXXXXXXXXXXX
FENCES,XXXX,RTLXXXXXXXXXXXX. XXXXXXXXXXXXXXX **XX **



XXXX ArduCopter XX XXXX **XX XX,XXXXXXXXXXXX **
XX DSHOT ESCs
XXXXXXXXXX rpm,XXXXXX Notch XXXXXXXXXXXXXXX (XX XXXX).



XXXX GPSXXXXXXXXXXXX (ACRO, STABILILZE, ALTHOLD, etc.). XX,XXX GPS/Compass
XXXXXXXXXX (X LOITERX RTL)XXXXXXXXXXXX,
XXXXXXXXXXXXXXXXXXXXXXXXXXXX/XX RCXXX GPSXXX RFXXX



If you do include a GPS for the advance mode capabilities, then there are some potential consequences that can occur when switching to position holding/tracking modes after aggressive maneuvering in modes like ACRO. These are:

- **EKF Failsafes** occurring
- Large position errors, ie like switching to RTL and then overshooting HOME position by very large amount, or tracking waypoints in a mission with large positioning errors

automatic-tuning-with-autotune

■■■■ automatic-tuning-with-autotune ■

■■■■■

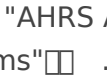
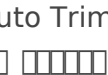
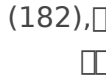
[illegible]

□□□□□□□□, □□□□□□□□, □□, **□□□□□□□□, **□□□□□□□□.

```

AHRS AutoTrim, , . a " AHRS
("182") . CH7 :
```

- ../images/MP_SaveTrim_Ch7PWMCheck.png

2.  RCX  "AHRs Auto Trim" (182),  Mission Planner  Aux  "Write Params"  RC  RC
3. Find a wind free environment with sufficient space to fly your copter without crashing into something.
4. Put the vehicle in manual Stabilize/AltHold mode or a position hold Loiter/PosHold mode (recommended).
5. Arm the vehicle and lift into a hover. The vehicle must be flying in order to enable autotrim. Flip the switch that you setup (CH7) to its High position (>1800us). "Autotrim running" message will be sent to the GCS.
6. If in a manual mode, fly your copter for about 25 seconds in a stable hover in position, keeping it from drifting with the pitch and roll sticks. As the trim is adjusted, less and less corrections with the sticks will be required. If using a position hold mode, take off and let the copter hover in place while not touching the sticks.
7. Once satisfied, flip the switch low, saving the trims, and land. "Trim save" message will be sent to the GCS.
8. Take off in AltHold mode and check if your copter is flying level now. If not repeat steps 4 - 7 again.



You can manually set the trim by modifying the **AHRS_TRIM_X** and **AHRS_TRIM_Y**. Roll trim is **AHRS_TRIM_X**, Pitch trim is **AHRS_TRIM_Y**. Both values are in radians with left roll and forward pitch being negative numbers.



It is nearly impossible to get rid of all drift so that your copter remains completely motionless without any input. It may be preferable to yaw the copter 90 degrees and holding that orientation for about 6-7 seconds each time so that any external disturbances will get cancelled out, resulting in a better trim.

Save Trim



the above AHRS AutoTrim method is preferred since it does not involve moving transmitter trims after RC calibration.

Save trim involves essentially transferring your radio transmitter's trims into the autopilot's AHRS trim (the **AHRS_TRIM_X** and **AHRS_TRIM_Y** parameters), **the same values that the AHRS AutoTrim method above adjusts. Normally, once common-accelerometer-calibration is done, you never want to change the transmitter's trims, but in this case it is done temporarily.** The "Save Trim" function ("5") is assigned to any free RC channel, usually a switch based one, using an **common-auxiliary-functions**. For the example below, Channel 7 will be used.

1. Check that your CH7 switch goes above 1800 on the MissionPlanner's Setup > Mandatory Hardware > Radio Calibration screen

[../images/MP_SaveTrim_Ch7PWMCheck.png](#)

2. Set your chosen channel's `RCx_OPTION` to "Save Trim" (5) using Mission Planner's Aux Function screen and press the "Write Params" button. See **common-auxiliary-functions** for details on assigning options to any RC channel.

[../images/MP_SaveTrim_Ch7.png](#)

3. With your CH7 switch in the off (Low) position, fly your copter in Stabilize or AltHold mode and use your transmitter's roll and pitch trim to get it flying level

4. Land and put your throttle to zero
5. Release the roll and pitch sticks and switch the CH7 switch high for at least 1 second. The words "Trim saved" should appear in the MP's Flight Data screen's Messages tab
6. Reset your transmitters roll and pitch trims back to the center and fly again and it should fly level now. If it does not repeat steps 3, 4 & 5



Save Trim also works while ****Simple or Super Simple mode **** is enabled, but take care not to change the vehicle's heading between landing and switching CH7 high, since the trim will be saved relative to the vehicle's heading at that moment.

Desktop method

The trim can also be updated by setting the vehicle level, connecting to the Mission Planner (or perhaps other ground stations) and selecting Initial Setup, Mandatory Hardware, Accel Calibration and pushing the lower "Calibrate Level" button.

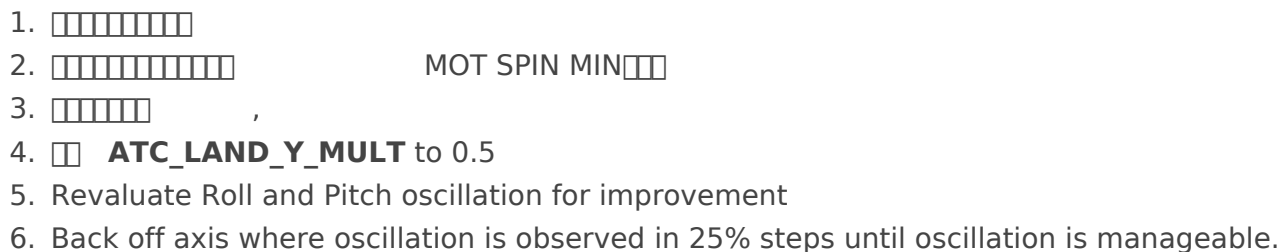
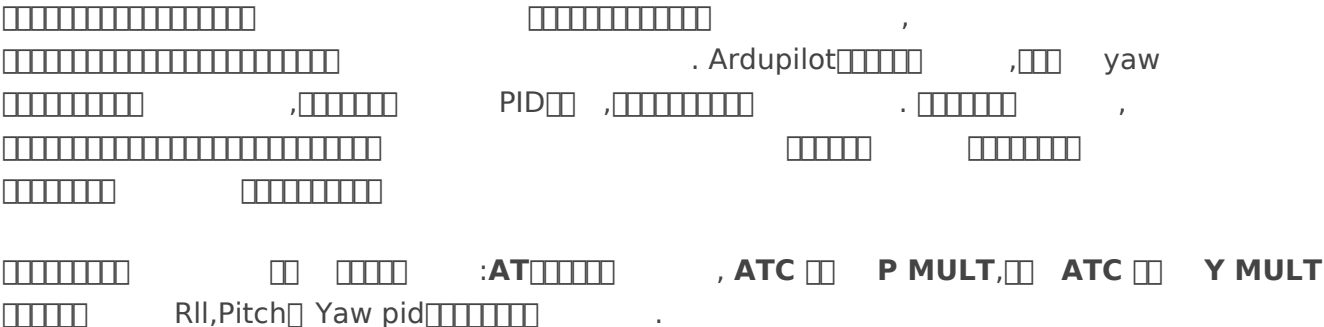
[../images/AccelCalibration_MP.png](#)

Please note though that making the HUD level while the vehicle is on the ground does not necessarily mean it won't drift horizontally while flying because of other small frame issues including the flight controller not being perfectly level on the frame, CG placement, an/ord slightly tilted motors.



calibrating-an-airspeed-sensor

calibrating-an-airspeed-sensor



Extreme care should be taken if **ATC_LAND_R_MULT** or **ATC_LAND_P_MULT** is reduced below 0.5.

IMU FFT

```
[ = "copter,plane"].
```

□□ FFT □□□□□□□□



 2MB  .  
 (GyroFFT)  ,  FFT Notch.

ArduPilot

- `**FFT ENBEL ** = 1` `FFT` . , `FFT` ,
FFT GCS , FFT ,
 FFT ,
`FFT` ,
`FFT` , CPU . **.
 , `cpu` , `FFT` **
`Notch` :
• `**INS HNTCH ENTCL** / **INS HNTC2 ENTCL ** = 1` = 1
• `**INS HNTCH MODE** / **INS HNTC2 MODE ** = 4` FFT
.
• Set `**INS_HNTCH_REF **` and/or `**INS_HNTC2_REF ** = 1` to set the harmonic notch reference value, which for FFT analysis generally means no scaling
• The harmonic notch reference frequency parameter, `**INS_HNTCH_FREQ **` and/or `**INS_HNTC2_FREQ **`, is used to indicate the lowest motor speed for which the ESC telemetry should be used to dynamically set the harmonic notch reference frequency.

For most uses with other FFT-related advanced parameters at their default, this is all that is required. The user can do optimization of the filtering setup by analyzing the test flight logs and adjusting notch bandwidth, if desired, by following the ****In-flight FFT Advanced Setup**** instructions.

Setting up the FFT parameters can be done automatically using the `RCx_OPTION` auxiliary function "162" on a transmitter switch. Set the function to a switch on the transmitter. Hover the vehicle, switch it on (high) for 30 seconds, and switch back low and land. The

parameters will have been set up and the switch function removed. NOTE: do not use this feature in firmware version 4.3!

□ □ □

Using In-Flight FFT can result in poorer performance than a properly set up **Throttle-Based** notch filter since the FFT computations take time and can lag the actual required center frequency. In-Flight FFT is useful when the rotor frequencies of the vehicle vary widely as in heavy lift vehicles operating with high and low loads. It can be useful in setting up **Throttle-Based** notch filters, however, see **In-flight FFT Advanced Setup** instructions for more information.

FFT Dynamic Harmonic Notch Frequency Tracking

FFT mode tracking sets the base frequency to the largest noise peak. Normally, when multiple harmonic notch filters are then enabled, the center frequency of each harmonic is locked to the base frequency of the first filter as an integer multiple, as determined by **INS_HNTCH_HMNCS**. **The bandwidth of the notch (and any enabled harmonic notches) will maintain a ratio set by the _FREQ/_BW params as the center frequency moves. Setting bit 1 of INS_HNTCH_OPTS, or INS_HNTC2_OPTS, will enable each harmonic filter to track the three largest noise peaks, individually.**

□ □ □

setting bit 1 of the notch options will also change the default value of **INS_HNTCH_HMNCS** to 1 instead of its normal 3. This is to maintain backwards compatibility with previous firmware versions. You can set **INS_HNTCH_HMNCS** back to 3, or whatever is desired, after setting bit 1 of **INS_HNTCH_HMNCS**

□ □ □

If you set bit 1 of **INS_HNTCH_OPTS**, or **INS_HNTC2_OPTS**, the bandwidth should not be half the frequency(default). It should be greatly reduced as more notches (3x more since now tracking three noise peaks) cause more phase lag (i.e. latency), thus the bandwidth

needs to be reduced to maintain a reasonable phase lag, else oscillation and a poorer tune will result despite the more accurate filtering compared to throttle-based filtering. The suggested starting point is setting the `_FREQ/_BW` params to 4/1 ratio instead of the default 2/1. This is because for notch filters, a wider bandwidth causes a greater phase lag per notch. You can also use the [Filter Tool](#) to check the phase lag for your chosen filtering settings. If your phase lag is higher than it was with the single peak filtering, then you can try reducing the bandwidth even further, balancing the phase lag and the amount of noise in the system.

FFT Options

There are two options that can be selected by setting the appropriate bit in the **FFT_OPTIONS** parameter that affect FFT operation:

Post Filter Chain FFT Analysis Window

Normally, the FFT analysis for adjusting the center frequency is done by measuring the noise directly at the output of the unfiltered gyro data. However, if bit 0 of **FFT_OPTIONS** is set, then the measurement window takes into account the effects of the low pass filter and any configured notch filter(s). This is useful if there is high-frequency noise, which impacts the control response less than lower-frequency noise due to the low pass at the end of the filter chain but may be targeted by the FFT measurement. Setting this bit will only track those lower frequency, and more critical, noise peaks.

Motor Noise Check

If bit 1 of **FFT_OPTIONS** is set and ESC motor rpm telemetry is available, then the measurement window for the FFT is centered about the motor(s) frequency as reported by ESC telemetry. This will generate a GCS warning message if noise from any motor greater than 40db is passing through the filter chain, and identify its level, motor number, and frequency. This bit must be used with bit 0 also set.

Typical Use

A typical use of the FFT notch filter is in addition to other dynamic harmonic notch filters (Throttle, ESC, or RPM-based). In these configurations, using the post-LPF FFT Window **FFT_OPTIONS** bit will

yield the best overall filtering results by positioning the FFT filter to target noise not filtered by the other notch filter and gyro LPF (**INS_GYRO_FILTER**).

Additional Information

For those interested in the details of how this feature works and tradeoffs in some of the advanced parameters, not normally adjusted by users, the ****In-Flight FFT: How it Works**** document describes the operation and these advanced parameters.



 *ArduPilot Copter*  *In-Flight FFT-Based Harmonic Notch Setup*                

IMU

--	--	--	--

```
[ = "copter,plane"].
```

--	--	--	--	--

Notch

Gyro

--	--

ArduPilot

 (, , slop),
 . 

```
[name] = " ". ArduPlane[ ] : [/ ] [name] = "copter". ArduCopter
[ ] : [/ ]
```

1. , , **2013-09-02**, ,
INS GYRO FILTER ().,

```
#. [REDACTED] INS_HNTCH_ENABLE ** and/or **INS_HNTC2_ENABLE **,
[site wiki="copter"] #. Attitude rate PID loop filtering of reference inputs, controlled by
**ATC_RAT_RLL_NTF, ATC_RAT_PIT_NTF and ATC_RAT_YAW_NTF. Attitude rate PID loop
filtering of error inputs is controlled by ATC_RAT_RLL_NEF, ATC_RAT_PIT_NEF and
ATC_RAT_YAW_NEF. For altitude acceleration control, use [PSC_ACCZ_NTF] and [PSC_ACCZ_NEF] on
firmware before 4.7.0, or PSC_D_ACC_NTF and PSC_D_ACC_NEF on firmware 4.7.0 and later.
[/site]
```

As discussed in **common-measuring-vibration** section, there are basically two classes of noise/vibrations: those generated within the bandwidth of the gyros/accelerometer sampling and noise above those frequencies which are "aliased" down to within that bandwidth and can cause the "leans". The aliased noise must be eliminated at the source with improved mounting or frame rigidity, but the above filters can deal with the other sources, typically generated from the motor/propellers at their rotation frequency and its harmonics.

For multicopters and QuadPlanes, virtually all vibrations originate from the motor's rotational frequency. For helicopters and planes, the vibrations are linked to the main rotor/prop speed.

ArduPilot has support for two dynamic harmonic notch filters whose filter frequency can be linked to the motor rotational frequency for motors, or the rotor speed for helicopters, and provides notches at a primary frequency and its harmonics. And support for eight static notch filters: four at the rate PID target inputs and four at the rate PID error inputs. These are useful for eliminating frame harmonic resonances.

While the lowpass filters can effectively diminish the impact of this noise, having low frequency set points creates a lot of phase lag and therefore reduces how aggressive the tune can be before oscillation occurs, which results in a poorer tune.

For the gyro based rate controllers, this reduces their ability to respond to fast disturbances. If the gyro lowpass filter can be set higher, the phase lag induced is lower and the tune can be more aggressive. But this allows more noise and vibration, effectively canceling that gain out. Enabling either one or both of the Harmonic Notch filters gives the ability to target the noise generated by the motors, allowing a higher frequency of the following low pass to be set and therefore allowing tighter tune.

Notch Filter Setup Overview

1. **Enable the notch filter.** This makes the mode-specific setup parameters visible after a reboot.
2. Select how the notch center frequency will be controlled. See **center_freq_control**.
3. If a static notch (not usually recommended), or **throttle-based** control is used, the dominant noise frequencies will need to be determined in order to setup the notch. See **notch_center_freq**.
4. Setup the selected center frequency control method using **INS_HNTCH_MODE ****. Then **setup its associated parameters by reading its associated page linked in this section **center_freq_control**.
5. After a test flight and log analysis with the notch enabled (See **notch_check**), the **number and placement of higher harmonic filters implemented can be adjusted **, the use of the **multi-notch options**, or even a second harmonic notch set (**INS_HNTC2_ENABLE**) can be configured to improve noise reduction effectiveness. The web based [Filter Review Tool](#) can be used to experiment with parameter changes based on the test flight log to determine best configurations without having to make iterative test flights.

Enabling the Notch Filters

The harmonic notch is enabled overall by setting **INS_HNTCH_ENABLE = 1** for the first notch, and, if needed, a second set of harmonic notches using **INS_HNTC2_ENABLE = 1**. After rebooting, all the relevant parameters will appear.

Various methods of dynamically adjusting the notch(s) center frequency to track motor speed under different thrust conditions is provided, ie Dynamic Harmonic Notch filtering.

Notch Filter Control Types

Key to the dynamic notch filter operation is control of its center frequency. There are five methods that can be used for doing this:

1. **INS_HNTCH_MODE ** = 0. **Static center frequency.** Dynamic notch frequency control is disabled. The center frequency is fixed and is static. Often used in Traditional Helicopters with external governors for rotor speed, either incorporated in the ESC or separate for ICE motors.
2. **INS_HNTCH_MODE ** = 1. (Default) **Throttle position based,** where the frequency at hover throttle is determined by analysis of logs, and then variation of throttle position above this is used to track the increase in noise frequency. See **throttle-based** for further setup details.



this type of notch filter cannot be used in Traditional Helicopters since throttle stick position does not relate to motor throttle, but rather collective and will result in a configuration error being declared at boot. The **Static center frequency** type is usually used since helis usually use constant motor speed.

3. **INS_HNTCH_MODE ** = 2 (RPM Sensor 1) or 5(RPM Sensor2). **RPM sensor based,** where an external **RPM sensor **** is used to determine the motor frequency and hence primary vibration source's frequency for the notch. Often used in **Traditional Helicopters (See **Helicopters)** using the ArduPilot Head Speed Governor feature. See **RPM Sensor** for further setup instructions.
4. **INS_HNTCH_MODE ** = 3. **ESC Telemetry based,** where the ESC provides motor RPM information which is used to set the center frequency. This can also be used for the forward motor in fixed wing flight, if the forward motor(s) ESCs report RPM. This requires that your ESCs are configured correctly to support BLHeli telemetry via **a serial port**. See **ESC Telemetry** for further setup instructions. If **INS_HNTCH_OPTS**, or **INS_HNTC2_OPTS** if the second set of notches is enabled, has bit 1 set, then a set of notches for each motor will be created, tracking its RPM telemetry, otherwise, the average frequency of all motors will set the center frequency.
5. **INS_HNTCH_MODE ** = 4. **In-Flight FFT based,** where a running FFT is done in flight to determine the primary noise frequency and adjust the notch's center frequency to match. This probably the best mode if the autopilot is capable of running this feature. It requires that the autopilot firmware supports it (see **common-limited_firmware** for GyroFFT feature) and has sufficient cpu power (F7/H7 autopilots). See ****In-Flight FFT **** for

further setup instructions.

ESC RPM telemetry is generally the best source of frequency data because it automatically adapts to varying payloads and is the easiest method to set up. Its only disadvantage is that the ESCs must provide fast and reliable RPM information. If the RPM data has a slow update rate or is unreliable, then another method should be used. The second-best method is throttle-based notch filtering. It has no update lag whatsoever, but its configuration is more complex and requires correctly setting ****MOT_SPIN_MIN**** and ****MOT_THST_EXPO****. If either parameter is incorrectly set, the notch center frequency will not match the motor frequency and its effectiveness will be reduced. Throttle-based notch filtering also requires retuning whenever the payload changes, because the ****INS_HNTCH_REF**** and/or ****INS_HNTC2_REF**** parameter must be changed.

All of the above are repeated, independently, for the second notch and are prefaced with `INS_HNTC2_` instead of `INS_HNTCH_`. The following will explain setup for the first set of notches.

□ □ □

only one filter can be mode 4(FFT).

Determining Notch Filter Center Frequency

Before actually setting up a dynamic notch filter, the frequencies that are desired to be rejected must first be determined. This is crucial if a static notch or **common-throttle-based-notch** is used. While the other methods do not require this knowledge before setting up their parameters, it can still be worthwhile as a comparison point for the post filter activation analysis of the filter's effectiveness.

Once the noise frequency is determined, the notch filter(s) can be further setup. Historically, **common-imu-batchsampling** has been used for this (also for slow cpu's like F4-based autopilots), logging short bursts of raw IMU data for spectral analysis.

As of firmware version 4.5 and later, a better method has been developed using continuous raw IMU data, if the autopilot is H7-based, and using a new web based tool. ****This method is now preferred****.

□ □ □

if you see two noise peaks close together, it may be that your copter has significant yaw imbalance. See the ****Yaw Imbalance section****.

Number of Harmonics Filtered

- Set ****INS_HNTCH_HMNCS**** to enable multiple harmonic filters centered at up to 16x the base frequency (multiples of the center frequency) notches. For ESC ****INS_HNTCH_MODE**** tracking, each motor will get a set of these notches. If an octocopter sets up three harmonics, this results in $8 \times 3 = 24$ notch filters. Enabling triple notches (see below) would result in 72 filters! This would most certainly cause excessive cpu loading and performance issues. Other modes only provide a single set of harmonic notches.

Always enable only the number of harmonic notch filters actually required and be especially aware of what is being enabled if using ESC (****INS_HNTCH_MODE**** = 3) tracking mode. Enabling too many will result in running out of CPU cycles with unpredictable results. Three harmonics is usually safe.

Checking Notch Filter Effectiveness

Once the notch filter(s) are setup, the effectiveness of them can be checked by again measuring the frequency spectrum of the output of the filters (which are the new inputs to the IMU sensors). Refer back to the **common-imu-batchsampling** or **common-raw-imu-logging** page for this.

Multi Notch

The software notch filters used are very "spikey" being relatively narrow but good at attenuation at their center. On larger copters the noise profile of the motors is quite dirty covering a broader range of frequencies than can be covered by a single notch filter. In order to address this situation it is possible to configure the harmonic notches as multiple notches that gives a wider spread of significant attenuation. The configuration is controlled by the ****INS_HNTCH_OPTS**** parameter. This is a bitmask parameter and multiple options are possible at the same time, but using bit 0, 1, and bit 4 at the same time should be avoided. Use only one of those in a given configuration.

INS_HNTCH_OPTS Bit Action	
0	Double overlapping Notches

**INS_HNTCH_OPTS ** Bit Action	
1	MultiSource: if using FFT Mode, the three largest noise sources will have a notch assigned. If ESC Telemetry Mode, then each motor will have a notch assigned at its RPM. If throttle mode, then each motor will have a notch assigned at its throttle input rather than the average overall throttle.
2	Updates the filters at the loop rate. This is cpu intensive, but tracks noise variations faster. Only valid if frequency source updates at loop rate, ie Bi-Directional DShot telemetry.
3	Enables notches on every IMU instead of just the primary. This is cpu intensive, but allows better lane switching decisions in noisy situations and for debugging. Not recommended for F4 boards.
4	Triple overlapping Notches
5	If the RPM source fails (Sensor,ESC telem,FFT) the minimum frequency (FFT_MINHZ , INS_HNTCH_FREQ , or INS_HNTC2_FREQ) will be used for notch center frequency

⏏ ⏏

double notch option is no longer recommended since the triple notch option has been added. With a double notch, the maximum attenuation is either side of the center frequency, so on smaller aircraft with a very pronounced peak their use is usually counter productive.

⏏ ⏏

Each notch has some CPU cost so if you configure multiple notches you can end up with many notches on your aircraft. For example, triple single (no harmonics) notches, using ESC telemetry will result in 3 notches per motor or 12 total notches. For example, with F4 cpus this should be acceptable, but enabling a second group of triple notches with **INS_HNTC2_ENABLE** or multiple harmonic notches, could cause problems.





--	--	--	--	--	--	--	--


$$[\begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \square \\ \hline \end{array}] = [\begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \square \\ \hline \end{array}].$$

-

Concepts

The two key controls for transmitter based tuning are:

- A "tuning knob", setup on your transmitter to a convenient knob or slider and linked to an RC channel that your board receives
- An optional "selector switch" for controlling advanced features of the tune. This should be linked to a two position switch mapped to a RC channel that your board receives. A spring loaded switch (such as the trainer switch) works particularly well.

In addition to those input controls the following concepts are useful in understanding the tuning process:

- A "parameter set" is a group of related parameters that can be tuned one after the other in a single flight
- The "mid-point" is point on the tuning knobs range where it produces a PWM value half way between minimum and maximum

Setting up for tuning

To setup your vehicle for tuning you need to set the following parameters:

- **TUNE_CHAN**: the RC input channel associated with your chosen tuning knob
- **TUNE_CHAN_MIN**: the minimum PWM values produced on **TUNE_CHAN**
- **TUNE_CHAN_MAX**: the maximum PWM values produced on **TUNE_CHAN**
- **TUNE_PARAM**: the parameter or set of parameters you will be tuning
- **TUNE_SELECTOR**: the RC input channel associated with your chosen selector switch (optional)
- **TUNE_RANGE**: the scaling range which the tuning knob covers

The **TUNE_PARAM** parameter selects the parameter or set of parameters you will be tuning. Values of **TUNE_PARAM** less than 100 correspond to individual tunable parameters whereas values of 101 or higher correspond to sets of related parameters that can be tuned one after the other in a flight.



Most of the tuneable parameters apply to the VTOL operation of QuadPlanes. Only **TUNE_PARAM** values in the 50-57 and 111-113 range are for fixed wing operation.

Use your ground stations parameter interface to see the full list of tunable parameters and parameter sets available for **TUNE_PARAM**. For Plane most of the parameters are associated with tuning the QuadPlane VTOL motors as those are the most difficult to tune. You can also tune some fixed wing parameters, although most people find the automatic tuning with AUTOTUNE mode is the best option for fixed wing flight.

If you want to tune a set of parameters (by choosing a **TUNE_PARAM** value over 100) you must have a selector switch configured with the **TUNE_SELECTOR** parameter.

Using the tuning knob

The basic operation of the tuning knob is very simple. If the tuning knob is at the bottom of its range then the parameter being tuned is set to the initial value divided by the **TUNE_RANGE**. If the tuning knob is at the top of its range then the parameter being tuned will be set to the initial value multiplied by **TUNE_RANGE**.

So with a default value for **TUNE_RANGE** of 2 you will be able to change the parameter in a range from half its initial value to 2x the initial value. This is a good range for many tuning tasks.

Activating the tuning knob

When you first start tuning a parameter you will find the tuning knob is not yet active. This is because the knob does not activate until it passes the "mid-point value", defined as half way between **TUNE_CHAN_MIN** and **TUNE_CHAN_MAX**. Activating the tuning knob in this way ensures that you don't accidentally take off with a large change in tuning value. You are guaranteed to start the tune with a value very close to your current value for the parameter.

When the tuning knob activates by reaching the mid-point the buzzer on the flight board will give a quick "bup-bip" sound to indicate that tuning has been activated.

Re-centering the tuning knob

It is quite common to find that the **TUNE_RANGE** is not wide enough to move the tuning value to the ideal point for your vehicle. For example, you may have started the tune with a P gain for some axis of 0.7, and when you move the tuning knob up all the way the vehicle still hasn't started oscillating. In that case the tuning value will have reached 1.4 and you need some more range.

To get more range you can use the selector switch to re-center the tuning knob around the current value. Toggle the selector switch briefly high then low and the center-value will change to whatever the tuning knob is set to. When you re-center the tuning knob will de-activate again until you move it to the mid-point position. This prevents you from getting a jump in the tuning value when you re-center.

Tuning multiple parameters

You can tune multiple parameters in one flight by setting **TUNE_PARAM** to one of the "tuning set" parameters. For example, if you set **TUNE_PARAM** to 101 then you will have 4 different stages to your tune:

- Roll: Rate D
- Roll: Rate P and Rate I
- Pitch: Rate D
- Pitch: Rate P and Rate I

notice that Roll: Rate P and Rate I is actually two parameters in one, controlling both the P gain and the I gain for roll. This follows the normal advice for MultiCopters that you should keep the P and I values equal when doing a manual tune.

When you choose a tuning set with **TUNE_PARAM** then you will initially be tuning the first parameter in the set. Once you have adjusted that parameter as much as you need to you can move to the next parameter in the set by holding the selector switch for more than 2 seconds. It is suggested that you count to 3 to ensure you are over 2 seconds.

Holding the selector switch for more than 2 seconds will switch you to the next parameter and will also change the tuning knob back to its "wait for mid-point" state on the new parameter. The buzzer on the board will give a loud BEEP sequence to indicate which parameter in the set you have changed to. For the first parameter in the set, you will get one loud BEEP. For the second parameter, you will get two loud BEEPs and so on.

When you have cycled through all of the parameters in the tuning set you have chosen it will wrap back around to the first parameter in the set.

Saving the tuning results

When you are happy with the tune you can save the result by holding the selector switch for more than 5 seconds. After 5 seconds the board will make a rapid bup-bip-bup-bip sound to indicate that the save is complete. If you leave the selector switch in the high position then tuning will remain disabled after the save.

Reverting the tune

If you are not happy with your tuning results or the vehicle becomes unstable you should change flight mode. Any change of flight mode will immediately revert all of the parameters you are tuning to the last saved value. However, you can prevent this reversion upon flight mode change by setting the **TUNE_MODE_REVERT** parameter to 0.

The tuning process

The tuning system is designed to make it easy to quickly get a reasonable manual tune on a vehicle in one flight. The most common use for this type of tuning will be in adjusting the rate roll and pitch PID gains. The tuning procedure outlined below is for that particular case.

Setting up

To setup for tuning your rate PIDs you should set **TUNE_CHAN** to your tuning channel, **TUNE_SELECTOR** to your selector switch and **TUNE_PARAM** to 101 (which is the "rate roll and pitch PIDs tuning set").

Then takeoff and switch the vehicle to a comfortable flight mode for rate tuning. For a QuadPlane QHOVER or QLOITER are the best choices.

The first parameter you will be tuning will be RateRollID. To tune that parameter (and the other parameters in the rate roll/pitch set) you should follow this process:

- move the tuning knob to the mid-point to activate the knob. You will hear a rapid bup-bip from the board to indicate the tuning knob is activated.
- start raising the tuning knob slowly, stopping immediately if the vehicle starts to oscillate. While you raise the gain you should give some small roll inputs on the sticks.
- if you get to the top of the tuning knob range and the vehicle has not yet started to oscillate then use the selector switch to re-center the range, then move the tuning knob to the mid-point to reactivate the tuning knob
- as soon as you see oscillation you should immediately lower the tuning knob to the point where the oscillation just stops
- once the oscillation stops then re-center the tuning knob using the selector switch, then move it to the mid-point to activate it, and then lower the tuning knob all the way down. That will move the tuning value to half of the value that just stopped the oscillation (assuming a **TUNE_RANGE** of 2). Moving to half of that value will give you enough margin in your tune to ensure your vehicle can handle a wide range of flight conditions.

Once you have completed the above process for the first parameter then you can move to the 2nd parameter by holding the selector switch for a count of 3. You will hear a BEEP BEEP sound from the vehicle indicating that you have moved to parameter 2, which is the RateRollPI parameters. You should then repeat exactly the same tuning process with that parameter.

Keep tuning each parameter in turn using the above process until you are happy with all of them and then save your new tuning parameters by holding the selector switch for more than five seconds. You will know the 5 seconds is up when you hear the distinctive rapid bup-bip-bup-bip sound from the buzzer.

At that point, you can land the vehicle, or just enjoy flying it.

The first time you do a full tune in this way it will probably take about five minutes of flight time to do a tune. With some practice, you can do a full tune in a bit over a minute.

[/site] [site wiki="copter"]

First you should set `RCx_OPTION` = 219, where `x` is a free RC channel that will be used for transmitter tuning.

□ □ □

prior to version 4.6, Channel 6 was hard coded to be the "tuning" channel and setting `RCx_OPTION` = 219 was not required to be set.

□ □ □

a second tuning channel can also be setup using `RCx_OPTION` = 220, and the `TUNE2_xx` parameters to allow two sets of tuning parameters to be adjusted in the same flight session.

The **TUNE** parameter determines which parameter is being tuned.

The **TUNE_MAX** parameter determines the maximum value of the parameter when the channel is at `RCx_MAX`, while the **TUNE_MIN** parameter determines the value when tuning RC channel is at `RCx_MIN`.

TUNE Values

** Traditional Heli Only

These values can be either set manually or using Mission Planner

Setting with Mission Planner

Rate Roll P and Rate Pitch P will be used in the following example procedure

[../images/RollPitchTuning.png](#)

1. Connect your autopilot to Mission Planner
2. From parameter list, assign channel ☐ to transmitter tuning with `RCx_OPTION` = 219.
3. On Mission Planner, select CONFIG>>Extended Tuning
4. Set the TUNE drop down box option to "Rate Roll/Pitch kP"
5. Set Min to 0.08, Max to 0.20 (most copters ideal gain is within this range although from a small number of copter the Max can be as high as 0.25)
6. Push the "Write Params" button
7. Turn your transmitter's CHx tuning knob to the minimum position, press the "Refresh Params" button and ensure that the Rate Roll P and Rate Pitch P values become 0.08 (or something very close)
8. Turn the CHx knob to its maximum position, press "Refresh Params" and ensure the Rate Roll P moves to 0.20
9. Move the CHx knob back to the middle
10. Arm and fly your copter in Stabilize mode adjusting the CHx knob until you get a copter that is responsive but not wobbly
11. After the flight, disconnect your LiPo battery and reconnect the autopilot to the mission planner
12. With the CHx knob in the position that gave the best performance, return to the Copter Pids screen and push the "Refresh Params" button
13. In the Rate Roll P and Rate Pitch P fields re-type the value that you see but just slightly modified so that the mission planner recognizes that it's changed and resends to the autopilot (Note: if you re-type exactly the same number as what appears in Rate Roll P it won't be updated). So for example, if the Rate Roll P appears as "0.1213" make it "0.1200"
14. Set CHx Opt back to "None" and push "Write Params"
15. Push the Disconnect button on the top right, and the Connect
16. Ensure that the Rate Roll P value is the value that you retyped in step #13

□ □

While you are moving the tuning knob the values update at 3 times per second. The need to press the Refresh button in the mission planner in steps #7 and #8 above is just because the Copter is not sending the updates to the mission planner in real-time.

[/site]

□□□□□□□□□□□□□□□□



Autotune. Autotune ,
Autotune . :

- 1. AltHold STABILIZE
- 2. . , , , , 4
- 3. ,
- 4.

,
(** **)

, : **ATC RATE FF ENAB ** 0

- 1. AltHold STABILIZE
- 2.
- 3.
- 4. 100%

, , (** **) .

Set **ATC_RATE_FF_ENAB** to 1 after the tests are complete.



, PID (LOOP), PID
 1/loop 400 , 2.5ms ArduCopter 4.7 , PID
 4KHz, 2.5uS ,
 () ,

CPU H , F4

PID , (LOOP):



:

- :
 - 0: 1
 - 1: CPU (), CPU .
 - 2: , , CPU
 - 3: Enable fixed fast rate. This is useful for testing but not usually used.
- FSTRATE_DIV**: this defaults to 1 and represents the minimum fast rate gyro divisor. So a value of 2 will yield a rate loop of gyro rate / 2 (**INS_GYRO_RATE** / 2). This allows a fixed fast rate that is lower than the gyro rate. This parameter is can be modified in-flight.

The easiest way to try this feature is to set **FSTRATE_ENABLE** = 1 (Dynamic rate). If your flight controller is struggling it will reduce the attitude loop rate until normality is restored. **FSTRATE_ENABLE** = 2 (Fixed Rate) is not recommended until you have seen reasonable performance with dynamic rates.

One way to gain more CPU is to switch off additional IMUs using **INS_ENABLE_MASK** or reduce **INS_GYRO_RATE**. The main CPU load is from the INS and attitude control threads.

When this feature is enabled, several other settings are affected:

- **SERVO_DSHOT_RATE** - this represents a multiplier of the main loop rate to output to the motors. When the fast rate loop is enabled this is normally set to 1, however if the calculated rate will not exceed the gyro rate, higher values are allowed. For instance if **INS_GYRO_RATE** = 2(4KHz) and **FSTRATE_DIV** = 4 (1KHz) then values of **SERVO_DSHOT_RATE** up to 4(4x loop rate) are allowed.
- Fast attitude and PID logging enabled in **LOG_BITMASK** - setting these bits usually logs at the main loop rate, however if fast attitude is enabled these will be logged at the minimum of the fast attitude rate and 1kHz
- **SCHED_LOOP_RATE** - since all the heavy lifting of attitude control is being done in a separate thread, **SCHED_LOOP_RATE** can be run at normal values - for instance 400Hz or even less.

Tuning Setup

Its extremely important that the **Notch filtering** **** be setup as well as the **tuning-process-instructions** be followed before tuning with this feature enabled.

- **INS_GYRO_FILTER** - as high as you can get without cooking your motors. Often copters have resonance around 120Hz so it is hard to get higher than this without notching out frame resonance.
- each axis' `ATC_RAT_xxx_FLTD` should be **INS_GYRO_FILTER**/2 on roll and pitch and **INS_GYRO_FILTER**/4 on yaw. This is because D is more active, lets more noise through and is more likely to cook your motors. Setting it above $0.75 * \text{INS_GYRO_FILTER}$ is not recommended.
- each axis' `ATC_RAT_xxx_FLTT` should be at the highest rate you want the EKF or your fingers to control the copter. Values of 30Hz should be fine, but higher has diminishing returns but does not affect the tune.





10个字节 (10个GPS, 10个) 10个字节。
 10个GPS 10个字节, 10个 ** GPS 10个 **

- 10个, 10个 10个
- GPS 10个, 10个 (10个)



10个字节 10个字节 10个 HDOP, 10个
 10个 10个 10个 10个 10个, 10个 10个, 10个 10个 1000个



** 10个 ** 10个 GPS, 10个 10个, 10个 10个



** 10个 ** 10个 10个 10个。
 10个 10个 10个, 10个, 10个 10个
 10个 10个



10个 **AltHold ** 10个 **sonar lidar** 10个, 10个
 10个, 10个 url= (10个)。 10个



- Disable the GPS by setting **GPS1_TYPE** to 0
- Configure the ****battery failsafe**** to "Land" by setting ****BATT_FS_LOW_ACT**** = 1
- Configure the ****radio failsafe**** to "Enabled always Land" by setting ****FS_THR_ENABLE**** = 3
- Disable the ****fence**** by setting ****FENCE_ENABLE**** = 0
- Use a downward facing sonar or lidar (if available)

Safe Indoor Flying Don'ts

- Don't fly in small confined spaces. Use common sense, flying inside a warehouse with a high roof = OK, bedroom = not OK.
- Don't use Auto* modes

* Autonomous and semi-autonomous modes require a position estimate. This can come from GPS or from ****Non-GPS Navigation**** sources such as optical flow, external navigation, or motion capture systems. In ArduPilot 4.7 and later, the EKF origin can be automatically saved and restored across power cycles — see ****Persistent Origin Storage**** for details.

**** Non-autonomous modes include ****Stabilize**** and ****AltHold******





Sung) . , ,
 ,  ,  ,  .
 ,  .

- [illegible]

- [illegible]

- **MOT SPIN MIN (()). Court. **

2. , , , , , , .

3.  ,  .
 . Alt Hold  ,


- [illegible]

- **** ATC THR MIX MAN**** 0.1
- ****MOT THST Hover **** 0.25()

- ****MOT THST Hover**** $\square$ 0.25()

5.  ( **common-radio-control-calibration**).
6. Configure an Emergency Stop Motors switch and test it (see ****Auxiliary Functions ****).
7. Do tuning flights in low-wind condition and normal weather (no rain and between 15°C/59°F and 25°C/77°F).
8. Practice STABILIZE flight in simulator or on a low-end drone first, you should be confident to be able to takeoff and land with your untuned aircraft.

6. Configure an Emergency Stop Motors switch and test it (see ****Auxiliary Functions****).

7. Do tuning flights in low-wind condition and normal weather (no rain and between 15°C/59°F and 25°C/77°F).

- Practice STABILIZE flight in simulator or on a low-end drone first, you should be confident to be able to takeoff and land with your untuned aircraft.

First Flight

The first take off is the most dangerous time for any multicopter. Care must be taken to ensure the aircraft is not destroyed in the first seconds of flight and nobody is injured.

- **Ensure that all spectators are at a safe distance.**
- **Ensure the pilot is at a safe distance and position.**
- The pilot should refresh themselves on the method used to disarm the aircraft (using ****Auxiliary Functions **** for Motor Interlock or Arm/Disarm may be beneficial).

This flight will allow to setup your aircraft in a "flyable for tuning" state.

1. Ensure the aircraft is in STABILIZE mode
2. Arm the aircraft
3. Immediately disarm the aircraft to ensure your disarm procedure is correct
4. Arm the aircraft
5. Slowly increase the throttle looking for signs of oscillation. (long or flexible landing gear may cause some landing gear oscillation that will only go away after the aircraft leaves the ground)
6. As soon as the aircraft lifts off the ground immediately put the aircraft back down as gently as possible
7. Disarm the aircraft
8. Evaluate what you observed to decide if you need to make adjustments to the tuning parameters or if it is safe to take off again
9. Arm and increase the throttle to initiate a takeoff
10. Hover at approximately 1m altitude and apply small (5 degrees) control inputs into roll and pitch
11. Immediately land if any oscillation is observed

Next section will explain how to remove the oscillations.

Initial aircraft tune

The first priority when tuning an multirotor aircraft is to establish a stable tune, free of oscillations, that can be used to do further tests.

1. Arm the aircraft in STABILIZE
2. Increase the throttle slowly until the aircraft leaves the ground
3. If the aircraft starts to oscillate immediately abort the takeoff and/or land the aircraft
4. Reduce all the following parameters by 50%

1. ****ATC_RAT_PIT_P ****
2. ****ATC_RAT_PIT_I ****
3. ****ATC_RAT_PIT_D ****
4. ****ATC_RAT_RLL_P ****
5. ****ATC_RAT_RLL_I ****
6. ****ATC_RAT_RLL_D ****

This process is repeated until the aircraft can hover without oscillations being detectable visually or audibly.

If the aircraft has very long or flexible landing gear then you may need to leave the ground before ground resonance stops.

Be aware that in this state the aircraft may be very slow to respond to large control inputs and disturbances. The pilot should be extremely careful to put minimal stick inputs into the aircraft to avoid the possibility of a crash.

Test AltHold

This test will allow to test the altitude controller and ensure the stability of your aircraft.

1. Check **MOT_HOVER_LEARN** is set to 2. This will allow the controller to learn by itself the correct hover value when flying.
2. Take off in STABILIZE and increase altitude to 5m. Switch to AltHold and be ready to switch back to STABILIZE. If the aircraft is hovering at a very low hover throttle value you may hear a reasonably fast oscillation in the motors. Ensure the aircraft has spent at least 30 seconds in hover to let the hover throttle parameter converge to the correct value. Land and disarm the aircraft.
3. Set these parameters on ground and preferably disarm (A confident pilot could set them in flight with GCS or CH6 tuning knob):

- **PSC_D_ACC_I** to $0.2 \times \text{MOT_THST_HOVER}$
- **PSC_D_ACC_P** to $0.1 \times \text{MOT_THST_HOVER}$



In Copter 4.6 and earlier these parameters were named **PSC_ACCZ_I** and **PSC_ACCZ_P** and were scaled 10x larger:

- **PSC_D_ACC_I** to $2 \times \text{MOT_THST_HOVER}$
- **PSC_D_ACC_P** to MOT_THST_HOVER

if AltHold starts to oscillate up and down the position and velocity controllers may need to be reduced by 50%. These values are: **PSC_D_POS_P** and **PSC_D_VEL_P**.

After you have a hover without oscillations the next step is to get a good notch filter setup to reduce noise to the PID controllers. A good set of notch filtering parameters is critical to a good tune.

To get a notch filter setup you need to hover your vehicle for at least 30 seconds with no pilot input and with **INS_LOG_BAT_MASK** set to 1. This will enable FFT logging which will guide the correct setup of the notch filters. You should then carefully read the **common-imu-notch-filtering** documentation and setup a harmonic notch to remove the noise from your gyros.

Eliminating noise with the notch filters will dramatically improve the quality of your tune.





Input Shaping

XXXXXXXXXX

XXXXXX

. XXXXXXXXXXXXXXXX

,

XXXXXXXXXXXXXXXXXXXX

.

XXXXXXXXXX

:

- **Y Rate:** XXXXXX (XX /X)
- **ATC ACC P** XX : Pitch XXXX
- **ATC ACC R MAX** XX XX
- **ATC ACC Y** XX : Yaw XXXX
- XXXX **ATC ANGLE MAX:**XXXX
- ****ATC_ANG_LIM_TC ****: Aircraft smoothing time
- ****ATC_INPUT_TC ****: Time to achieve 63% seady state pitch and roll attitude. Multiply by 3 to determine time to achieve steady state attitude.
- ****PILOT_Y_RATE_TC ****: Time to achieve 63% steady state yaw rate. Multiply by 3 to determine time to achieve steady state rate.

Autotune will set the **ATC_ACC_P_MAX**, **ATC_ACC_R_MAX** and **ATC_ACC_Y_MAX** parameters to their maximum based on measurements done during the Autotune tests. These values should not be increased beyond what Autotune suggests without careful testing. In most cases pilots will want to reduce these values significantly.

For aircraft designed to carry large directly mounted payloads, the maximum values of **ATC_ACC_P_MAX**, **ATC_ACC_R_MAX** and **ATC_ACC_Y_MAX** should be reduced based on the minimum and maximum takeoff weight (TOW):

- **ATC_ACC_P_MAX** x (min_TOW / max_TOW)
- **ATC_ACC_R_MAX** x (min_TOW / max_TOW)
- **ATC_ACC_Y_MAX** x (min_TOW / max_TOW)

PILOT_Y_RATE should be set to be approximately 0.5 x **ATC_ACC_Y_MAX** to ensure that the aircraft can achieve full yaw rate in approximately half a second.

****ATC_ANG_LIM_TC **** may be increased to provide a very smooth feeling on the sticks at the expense of a slower reaction time.

Aerobatic aircraft should keep the **ATC_ACC_P_MAX**, **ATC_ACC_R_MAX** and **ATC_ACC_Y_MAX** provided by autotune and reduce ****ATC_ANG_LIM_TC **** to achieve the stick feel desired by the pilot. For pilots wanting to fly ACRO the following input shaping parameters can be used to tune the feel of ACRO:

- ****ACRO_BAL_PITCH ****

- ****ACRO_BAL_ROLL ****
- ****ACRO_RP_EXPO ****
- ****ACRO_RP_RATE ****
- ****ACRO_RP_RATE_TC ****
- ****ACRO_THR_MID ****
- ****ACRO_TRAINER ****
- ****ACRO_Y_EXPO ****
- ****ACRO_Y_RATE ****
- **ACRO_Y_RATE_TC**

The full list of input shaping parameters are:

- ****ACRO_BAL_PITCH ****
- ****ACRO_BAL_ROLL ****
- ****ACRO_RP_EXPO ****
- ****ACRO_RP_RATE ****
- ****ACRO_RP_RATE_TC ****
- ****ACRO_THR_MID ****
- ****ACRO_TRAINER ****
- ****ACRO_Y_EXPO ****
- ****ACRO_Y_RATE ****
- ****ACRO_Y_RATE_TC ****
- **ATC_ACC_P_MAX**
- **ATC_ACC_R_MAX**
- **ATC_ACC_Y_MAX**
- **ATC_ANGLE_MAX**
- ****ATC_INPUT_TC ****
- ****ATC_ANG_LIM_TC ****
- ****ATC_RATE_P_MAX ****
- ****ATC_RATE_R_MAX ****
- ****ATC_RATE_Y_MAX ****
- ****ATC_RATE_WPY_MAX ****
- **PILOT_ACC_Z**
- **PILOT_SPD_DN**
- **PILOT_SPD_UP**
- ****PILOT_THR_BHV ****
- ****PILOT_THR_FILT ****
- **PILOT_TKO_ALT_M**
- ****PILOT_Y_RATE ****
- ****PILOT_Y_RATE_TC ****
- **LOIT_ACC_MAX_M**
- ****LOIT_ANG_MAX ****
- **LOIT_BRK_ACC_M**
- ****LOIT_BRK_DELAY ****
- **LOIT_BRK_JRK_M**
- **LOIT_SPEED_MS**



navigation-tuning



navigation-tuning



new-roll-and-pitch-tuning

new-roll-and-pitch-tuning

QuikTune

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#) [http](#)

[http](#) [http](#)

“ □□ : 100% ”

VTOL Lua , copter .

 , . 29. 60%,
 ,  .

`00000000`, `00000000`:`000000` `00000000`,
`00000000`, `00` - `0000` `00000000000000000000` QuickTune `00`
ac `0000` if the vehicle can't do an initial hover stably. Then setup the filters and retune
for best results.

The script attempts to tune all these parameters (in the given order)

- **ATC_RAT_RLL_D**
- **ATC_RAT_RLL_P** and **ATC_RAT_RLL_I**
- **ATC_RAT_PIT_D**
- **ATC_RAT_PIT_P** and **ATC_RAT_PIT_I**
- **ATC_RAT_YAW_D**
- **ATC_RAT_YAW_P** and **ATC_RAT_YAW_I**

The advantage over **AutoTune **** is that **QuikTune** is safer because the vehicle does not need to move or twitch. The disadvantage is that **QuikTune** cannot find the vehicle's maximum rotational accelerations (e.g. ****ATC ACC R MAX, ATC ACC P MAX,**

Installing the Script

- Set `**SCR_ENABLE** = 1` to enable scripting and then reboot the autopilot
- Download [VTOL-quicktune.lua](#) to your PC
- Copy the script to your autopilot's SD card's APM/scripts directory. If using MP it may be easiest to use the Config, MAVFtp screen
[../images/quiktune-mp-mavftp.png](#)
- Reboot the autopilot and set `QUIK_ENABLE = 1`
- If an RC switch will be used to start/stop the tune set `RCx_OPTION = 300` where "x" is the RC input channel number. Alternatively set one of Mission Planner's Aux Function tab's rows to "Scripting1"
[../images/quiktune-mp-auxfunction.png](#)

Running QuikTune

- Wait for a calm day and go to an open area with good GPS reception
- Connect with a ground station (e.g Mission Planner or QGC) and ensure that the Messages tab can be seen. This is where output from the tune will appear
- Move the RC switch to the low position OR push MP's Aux Function's "Low" button
- Arm and takeoff in Loiter mode and climb to a height of about 3m
- Begin the tune by moving the RC switch to the middle position OR push MP's Aux Function's "Mid" button
- Monitor the progress of the tune using the GCS's Messages tab
- If necessary reposition the vehicle using the RC transmitter. This will temporarily pause tuning and restore the original gains. Tuning will resume a few seconds after the RC sticks are returned to their center position
- If the vehicle begins oscillating violently cancel the tune by moving the RC switch to the low position OR push MP's Aux Function's "Low" button
- Once the tune has completed accept the new gains by moving the RC aux switch to the high position OR push MP's Aux Function's "High" button
- Land and disarm the vehicle

Advanced Configuration

The full list of available [parameter settings are here](#)



roll-pitch-controller-tuning

roll-pitch-controller-tuning

Next setup the thrust expo. If you are setting up a professional aircraft then you should invest in a thrust stand so you can accurately measure the true thrust for your motor/ESC/propeller combination as you vary the throttle. Then you will adjust the expo value along with the endpoints (given by motors setup below) so that the thrust between the endpoints is as linear as possible. Do not trust manufacturer data for the thrust curve as they are frequently inaccurate. See **motor-thrust-scaling** for details on thrust scaling.

If you are setting up a hobby grade vehicle then you can use the graph below to estimate the correct **MOT_THST_EXPO** value for your aircraft.

- **MOT_THST_EXPO**: 0.55 for 5 inch props, 0.65 for 10 inch props, 0.75 for 20 inch props (or larger). This parameter should be derived by thrust stand measurements for best results (don't trust manufacturer data).

[../..../images/tuning-process-instructions-1.hires.png](#)



Do not set an expo above 0.75 unless you have thrust stand data for your own motor, ESC and propeller combination showing that a higher value is needed. The graph is an approximation for hobby grade hardware, and guessing at a high expo can make the vehicle harder to tune rather than easier.

Motors setup

The motor parameters define the PWM output range sent to the ESCs. This is critical to ensure that the entire range of throttle values used in flight is within the linear range of your propulsion system.

Parameters used to define the output range sent to the ESC.

- **MOT_PWM_MAX**: Check ESC manual for fixed range or 2000us
- **MOT_PWM_MIN**: Check ESC manual for fixed range or 1000us
- **MOT_SPIN_ARM**: use the **motor test feature** to determine a value which will reliably start the motors spinning at a low rpm as an indication of the armed state.
- **MOT_SPIN_MAX**: 0.95
- **MOT_SPIN_MIN**: use the **motor test feature** and motor test data, if available, to set the lower range of linear thrust. The default value is usually adequate for hobby uses.
- **MOT_THST_HOVER**: 0.25 or below the expected actual hover thrust percentage (lower is safe)

PID Controller Initial Setup

The settings below are meant to get your PID controller acceleration and filter settings into the right approximate range for your vehicle. These parameters are critical to the tuning process. The PID controller default values for axis P/D/I values are usually safe for first test hovers of most vehicles.

- ****INS_ACCEL_FILTER****: 10Hz
- ****INS_GYRO_FILTER****: 80Hz for 5 inch props, 40Hz for 10 inch props, 20Hz for 20 inch props (or larger)
- **ATC_ACC_P_MAX**: 1100 for 10 inch props, 500 for 20 inch props, 200 for 30 inch props
- **ATC_ACC_R_MAX**: 1100 for 10 inch props, 500 for 20 inch props, 200 for 30 inch props
- **ATC_ACC_Y_MAX**: 200 for 10 inch props, 100 for 20 inch props, 90 for 30 inch props
- ****ATC_RAT_PIT_FLTD****: ****INS_GYRO_FILTER**** / 2
- ****ATC_RAT_PIT_FLTT****: ****INS_GYRO_FILTER**** / 2
- ****ATC_RAT_RLL_FLTD****: ****INS_GYRO_FILTER**** / 2
- ****ATC_RAT_RLL_FLTT****: ****INS_GYRO_FILTER**** / 2
- ****ATC_RAT_YAW_FLTE****: 2
- ****ATC_RAT_YAW_FLTT****: ****INS_GYRO_FILTER**** / 2

[../images/tuning-process-instructions-2.hires.png](#) [../images/tuning-process-instructions-3.hires.png](#)
[../images/tuning-process-instructions-4.hires.png](#)

Mission Planner Helper

A tab under SETUP/Mandatory Hardware/Initial Parameter Setup is provided in Mission Planner to setup the above parameters easily.

[../../../../images/mp-initial-copter-param-setup.png](#)



tecs-total-energy-control-system-for-speed-height-tuning-guide



tecs-total-energy-control-system-for-speed-height-tuning-guide





Throttle Boost

ArduPilot uses a throttle boost feature to help maintain a constant throttle position during "punch-outs" or "throttle chops". This feature is controlled by the ATC THR G parameter. When enabled, the throttle will automatically increase to a predefined level when the throttle position drops below a certain threshold. This is useful for maintaining a consistent throttle position during maneuvers that require a sudden increase in throttle, such as a punch-out or a throttle chop. The ATC THR G parameter is set in the ArduPilot configuration menu. The default value is 0, which means the throttle boost is disabled. Setting it to 1 enables the throttle boost. The ATC THR G parameter is also used to control the throttle boost during a punch-out or throttle chop. When the throttle position drops below the threshold, the throttle will automatically increase to the level specified by the ATC THR G parameter. This helps to maintain a consistent throttle position during these maneuvers. The ATC THR G parameter is also used to control the throttle boost during a punch-out or throttle chop. When the throttle position drops below the threshold, the throttle will automatically increase to the level specified by the ATC THR G parameter. This helps to maintain a consistent throttle position during these maneuvers.



yaw [] , [] , [] [] ,
[]

[] . [] ,
[] [] / []



GCS [] [] , [] , []
[] , [] :

` [ ] (3) `

[] . [] ArduCopter [] ,
yaw [] . [] , [] , []

[] , [] .
[] .

[] , [] [] ,
[] .



yaw [] yaw [] , [] yaw [] . [] yaw []
[] [] , [] yaw
[] [] [] :

`Yaw Imball 87 %` %`

[] , [] . [] yaw [] , []
flash [] ,Yaw [] PWM [] ấn [] .
[] , [] :

[../././images/yaw_imbalance_log.png](#)

This should be fixed in hardware. The most common cause is the motors not being vertical on a circular arm. If the imbalance still persists the motors can be tilted slightly such that the thrust angle assists the yaw for its rotation direction. Some vehicles can be extremely sensitive to motors thrust vector.

If the warning is only seen in aggressive yaw maneuvers, the warning threshold can be increased by raising **ATC_RAT_YAW_IMAX**. However, it may also be worth revisiting the yaw tune.

Both the thrust loss and yaw imbalance warnings can be disabled with the **FLIGHT_OPTIONS** parameter. This should only be done after extensive log review and testing to verify the warnings are not picking up genuine issues.



XXXXXXXXXX :

1. XXXX ,XXXXXXXXXX
2. **XXXX **XX ,XXXXXXXXXX .
3. ** XXXXXXXX **
4. XX XX nochXXX ,XX XX - XXXX
5. XXXXXXXX XX /XXX **qui**XX (XXXX AUTOTUNEXXXX) !
6. XXXX .
7. XX **Input Shaping** parameters to obtain the desired "feel"

The initial tune of the aircraft should be done **in the aircraft's most agile configuration**. This generally means that the aircraft will be at its minimum take off weight with fully charged batteries.

XX X

Following each of the above steps will usually result in a safe tuning process and adequate tune for most users. Alternatively consider using the [Methodic Configurator](#) for step-by-step setup

Advanced Tuning

ArduCopter has an extremely flexible controller design that has been used with great results on aircraft from 100g to 500kg. There are a number of difficult control problems that provide a greater depth of understanding that can be provided here. Some of these issues include:

- High gyro noise levels
- Flexible airframes
- Soft vibration dampers
- Large payloads on flexible or loose mounts
- Rate limited actuators
- Non-Linear actuators
- Extremely aggressive or dynamic flight



