

[Guides](#)[Current](#)[Dynamic Idle](#)

Dynamic Idle

Introduction

Dynamic Idle improves how Betaflight controls the motors at the low end of the rpm range.

It requires an ESC that supports Bidirectional DShot telemetry - ie, a DShot ESC that can send RPM data to the flight controller.

Using this RPM data, the flight controller can dynamically alter motor drive to prevent any motor falling below a set minimum RPM. This improves handling and reduces the risk of desyncs. It also allows lower fixed idle settings, which improves straight line braking and hang time. Additionally, dynamic idle improves PID controller performance at zero throttle by permitting stronger braking of motors that are in positive airflow conditions.

NOTE 1: Dynamic Idle requires **fully functional BiDirectional DShot telemetry**.

NOTE 2: **Dynamic Idle is off by default**. To enable it, enable DShot Telemetry, and set `dyn_idle_min_rpm` to a suitable value in the CLI. For typical 5" quadcopters 30-40 (3000-4000 RPM) is a suitable starting point. Please read the tuning section, below.

NOTE 3: Dynamic Idle must not be used with 3D mode.

NOTE 4: **Transient throttle limit should be disabled while using Dynamic Idle**
(`set transient_throttle_limit = 0`)

How does Dynamic Idle work?

Let's first explain how idle is managed without dynamic idle.

Without dynamic idle, the lower limit of motor drive, under all conditions, is set by the `dshot_idle_value`. This defaults to 5.5%. When throttle stick is at zero, the motors always get 5.5% motor drive. The PIDs cannot reduce motor drive below 5.5%, either.

Imagine we are in a flat drop, or have done a quick 180 degree reversal from high forward speed. In both these situations, airflow is pushing into the props from below the quad (negative inflow). This can slow them down a lot. They may either turn so slowly that they can't start quickly when needed, or stop spinning completely, or even spin backwards. This can lead to poor handling or total loss of control (motor desync), often most obvious at the end of fast flips or rolls. To avoid these issues, we usually need to set the `dshot_idle_value` relatively high.

Also, because the motors are always getting a relatively high drive signal, even when it isn't really needed, we get a limit on our inverted hang time, the quad can be floaty, and we don't get such strong motor braking as we need when stopping flips and making quick moves, or when chopping throttle.

If we want to quickly stop a fast roll move, we need to slow down the driving motors. When we tell them to stop, they are moving quickly forward in the air. They have strongly positive airflow which keeps them spinning faster than they need to be. The PIDs would really like to send zero motor drive to them - maximal braking - but cannot send less than 5.5%, even though they are in no danger of stopping. Because they can't be slowed down as much as they could, our ability to stop the turn is not as good as it could be.

Finally, lets consider an inverted drop, with a yaw while inverted. While dropping cleanly, the motors will be at 5.5% motor drive, pushing downwards. Air is flowing into the props as we drop, and likely they are spinning faster than is needed. We would like them to spin more slowly, or resist spinning harder, to improve 'hang time' - but we can't, because DShot idle is a high enough number to prevent desyncs. Additionally, if we try to yaw while inverted, the PIDs can only make the motors go faster than idle, and speeding up the motors while inverted really limits hang time.

How does dynamic idle change things?

With dynamic idle enabled, the PIDs are allowed to send zero drive to the motors, so long as they don't spin too slowly. The actual RPM is continually monitored using Bidirectional DShot telemetry. Idle motor drive is dynamically adjusted to keep the slowest motor above the configured minimum RPM - even if its drive signal is zero.

Hence dynamic idle will greatly reduce the chance of a desync, and allow lower overall idle settings.

Because the motors won't slow down as much under strong reverse airflow conditions, they can start up quickly. This provides improved zero throttle turn responsiveness, greater stability against cross-axis wobble when stopping a flip quickly, and improved zero throttle stability in flat drops.

Under strongly positive inflow conditions, where the motors will be spinning faster than expected, they can now be braked much harder, since the PIDs can send as little as zero drive to slow them down, instead of the mandatory 5.5%. This is beneficial when quickly stopping a fast flip or roll, or when performing tricks during inverted drops.

These are the three key elements to dynamic idle for 4.2, but only the first two are relevant in 4.3:

1. We get the actual motor RPM via bidirectional DShot telemetry, and use this to prevent any motor falling below the set `dyn_idle_min_rpm` value.
2. We allow the PIDs to send motor drive to zero, provided that RPM is OK
3. In 4.2, the `dshot_idle_value` becomes our minimum throttle value. In 4.3, the `dshot_idle_value` is ignored.

Setup - enabling dynamic idle

- Enable DShot Telemetry
- set `dyn_idle_min_rpm` above zero, eg between 30 and 40, to enable Dynamic Idle.

In Betaflight 4.3, that's all you need to do. 4.3 users should follow the instructions in the 4.3 tuning guide, and not the instructions below.

In 4.2, the Dynamic Idle control code was not so effective, and the following (complicated) setup procedure was recommended:

- Take props off !!
- Set your normal DShot Idle value in the configuration page of the configurator
- Go to the motors tab
- Re-check that you really did take props off !! :-)
- Connect a Lipo at around 3.8 - 3.9V
- Enable the motors
- Use the master slider to adjust motor drive; for example, if your DShot Idle value is 5%, set the motor drive to 1050
- read the RPM you get at this idle.
- set your dynamic idle minimum RPM value to something a bit below the idle RPM. Note that 20 means 2000 RPM.

For example, if the idle RPM at 1050 was say 2,000, take two zeros off to get 20, then take off 20% to get 16; set your dynamic idle minimum (`dyn_idle_min_rpm` in

CLI) to 16.

This setting method puts a kind of throttle up value of 5.5% above a minimum rpm which acts as a lower 'back-stop'.

If you already are running a known good DShot idle value, start off with that, and set the `dyn_idle_min_rpm` value set 20% lower as described above.

Tuning

Most of the benefits of dynamic idle happen 'out of the box' as soon as dynamic idle is enabled at the correct rpm value. Turn responsiveness, drop stability, and desync reduction should improve immediately.

Higher minimum RPM values - 30-40 - with lower DShot Idle percent values will keep the motors spinning more quickly when the throttle is cut, or in strongly positive inflow states. This will lead to more rapid spool up when needed, better zero throttle stability, and crisper flips - at the cost of less effective braking and a more 'floaty' feel, and reduced inverted hang time.

Lower idle values (both reduced together) will lead to improved inverted hang time, at the cost of greater risk of de-sync, reduced stability when chopping throttle hard after a punch, when blipping up from idle, and any time you are at idle, eg mid-flip, flat drops, etc. Because the motors will idle more slowly, they may have difficulty re-starting after being commanded to idle, eg at the end of a flip, causing instability and bounce back.

Every quad is different, and the purpose we put them to varies a great deal. Unless your requirements or build are unusual, the default idle value is usually quite good.

Low authority quads (larger props, weaker motors, ducted cine quads, endurance quads) will typically do better with higher idle values.

High authority quads often fly best with idle values around 4-4.5%.

I'm after lots of inverted hang time in 4.2

To get longer inverted hang time, go for the lowest possible values of both DShot idle percentage and `dyn_idle_min_rpm`. Keep `dyn_idle_min_rpm` adjusted relative to DShot as described above (20% lower than the idle rpm).

REMEMBER: both values must be reduced!

You can go as low as you like, eventually you will get annoying instability with throttle chops, or desyncs.

If you were running DShot idle of 5% and `dyn_idle_min_rpm` of 20 was 20% below rpm at 5%, you could try DShot idle of 4% and `dyn_idle_min_rpm` of 16. If that was OK, then 3.5% and 14, even 3% and 12.

When hard flip stops or throttle chops get ragged, or you start tumbling out of the sky under reverse airflow due to a desync, that's too low. You can typically get well below the default DShot idle value.

Really low idle settings may maximise inverted time but will adversely affect stability as described above.

Is this useful for whoops?

Yes, for sure. 4.3 is better. Whoops have low authority at idle that is often improved significantly with the combination of Thrust Linear and Dynamic Idle.

Whoops may require higher than default idle values to keep their motors spinning reliably.

I'm a very high turn rate Acro LOS pilot

When targeting very high maximum turn rates - 1800 deg/s for example - desyncs can be more of a problem than for quads with lower target rates.

For maximum acrobatic and hang performance in a high power to weight LOS quad, the goal would be the lowest idle values that didn't cause desyncs or other adverse effects like excessively slow startup after flips.

Note that if the rpm from `dyn_idle_min_rpm` is higher than that of DShot idle, hovers may not be smooth, since the dynamic idle controller may hunt around a little, but desyncs are less likely.

I'm a race pilot

Mostly a high minimum RPM of 3500 - 4000 works best. The DShot Idle value can be set low, eg 2-3%

Racers typically use low throttle very little of the time, so you might wonder why you'd bother with something that only influences idle behavior.

There are two reasons.

First, braking. Sure we want to go fast, but we also want to slow down. If we can set idle lower than usual, we can brake harder. Dynamic idle lets us do that. We can go full throttle, and when we cut throttle, we will bleed off speed more quickly, making

the next turn easier. We will also be able to drop into dive gates more aggressively on cutting throttle.

Second, turn performance high speed. Typically at high speed we have strong positive inflow. We can turn faster with dynamic idle active because under positive inflow conditions it can brake motors down to zero drive, helping turn in more quickly.

When flying race courses that involve high speed entries to maneuvers such as split-S or tight hairpins, dynamic idle may allow more consistent 'reverse throw' and improved control/propwash management, allowing more accurate rhythm-based turn entry and vertical maneuvers (e.g. repetitive dive/antigravity gates or inverted dive entries).

Optimal dynamic idle for experienced racers, who keep throttle on most of the time, may be with slightly reduced idle values. Something like your DShot idle reduced to say 4 - 4.5%, and `dyn_idle_min_rpm` 20% below that, maybe. The downside of lower idle values is that if you do make a mistake and do a zero throttle reversal, it may be uglier.

The technical stuff

When we first arm, until Airmode is activated, Dynamic Idle is constrained to a fixed 4% maximum increase in motor drive, for safety reasons. If we have set DShot idle to a very low value, like 1%, we may see a slightly lower, and very stable, RPM before Airmode activates - a value which may be less than our configured minimum RPM. Once Airmode is active, Dynamic Idle becomes fully effective, that is removed, and the RPM should not fall below the target RPM while in flight.

Dynamic idle adjustments are not instantaneous. The correction factor uses a special PID controller where the adjustment is proportional to the difference between the needed and actual motor accelerations. Feedback delays have some potential for instability when extreme values are used.

The stability of the dynamic idle controller can be tested by hovering with very low dynamic idle. The quad will rely entirely on dynamic idle to maintain stable hover RPM. On unusual builds, it might be unstable so take care. You would do this kind of testing only to check that the dynamic idle system was working well. Usually it is not needed.

The secondary settings are best left at defaults.

`idle_adjustment_speed` is a gain multiplier on the crude difference between actual and target rpm. Default is 50.

`idle_p` sets the gain of the controller. Higher values will cause corrections to be more aggressive.

These two parameters should not need adjusting in most cases. To confirm that they are appropriate for your quad, enable the RPM debug, fly straight vertically up at speed, cut throttle quickly to zero, and drop straight flat down under zero throttle. If the motor traces and rpm traces are smooth, all is good. If they are wobbling, the values may need to be tweaked a bit. My race quads work very well with `idle_p` a bit higher, at 80, but increasing `idle_adjustment_speed` as well made the rpm traces a bit sawtoothed.

`idle_pid_limit` constrains the maximum allowed error correction. If set too low, reverse airflow may still cause a desync because the amount of correction may be inadequate.

`idle_max_increase` is a limit on the maximum allowed percentage increase in idle above zero, expressed as percent *10. Default of 150 means 15% motor drive is the maximum increase that can be applied to prevent a desync. Higher values run the risk of driving the other motors very hard in the event of a prolonged desync.

The Dynamic Idle debug returns:

0 : motorRangeMinIncrease _ 1000 1 : targetRpsChangeRate (simple RPM error _ `idle_adjustment_speed`) 2 : error (amount of error to fix) 3 : minRps (lowest current motor rpm, in revolutions per second * 10)

So, for example, a minRps value of 500 corresponds to $50.0 * 60 = 3000\text{rpm}$

Youtube

 [Joshua Bardwell](#)

 [Ivan Efimov](#)

Community

 [Discord](#)

 [IntoFPV](#)

Links

 [Oscar Liang](#)

 [VitroidFPV](#)

Feeds

 [RSS](#)

 [Atom](#)

 [JSON](#)



Copyright © 2025 All rights reserved Team Betaflight

Built with Docusaurus

made with ❤️ by **VitroidFPV** and **un!t**

