6 stealthChop™

stealthChop is an extremely quiet mode of operation for stepper motors. It is based on a voltage mode PWM. In case of standstill and at low velocities, the motor is absolutely noiseless. Thus, stealthChop operated stepper motor applications are very suitable for indoor or home use. The motor operates absolutely free of vibration at low velocities. With stealthChop, the motor current is applied by driving a certain effective voltage into the coil, using a voltage mode PWM. With the enhanced stealthChop2, the driver automatically adapts to the application for best performance. No more configurations are required. Optional configuration allows for tuning the setting in special cases, or for storing initial values for the automatic adaptation algorithm. For high velocity drives spreadCycle should be considered in combination with stealthChop.

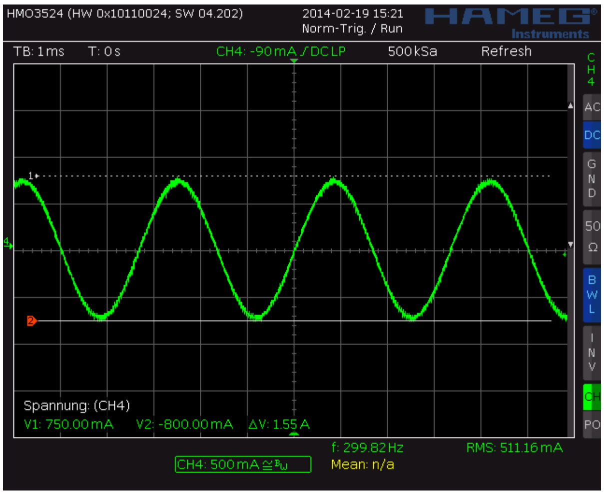

Figure 6.1 Motor coil sine wave current with stealthChop (measured with current probe)

6.1 Automatic Tuning

stealthChop2 integrates an automatic tuning procedure (AT), which adapts the most important operating parameters to the motor automatically. This way, stealthChop2 allows high motor dynamics and supports powering down the motor to very low currents. Just two steps have to be respected by the motion controller for best results: Start with the motor in standstill, but powered with nominal run current (AT#1). Move the motor at a medium velocity, e.g. as part of a homing procedure (AT#2). Figure 6.2 shows the tuning procedure.
Border conditions for AT#1 and AT#2 are shown in the following table:

AUTOMATIC TUNING TIMING AND BORDER CONDITIONS			
Step	**Parameter**	**Conditions**	**Required Duration**
AT#1	*PWM_ OFS_AUTO*	- Motor in standstill and actual current scale (*CS*) is identical to run current (*IRUN*). - If standstill reduction is enabled, an initial step pulse switches the drive back to run current, or set *IHOLD* to *IRUN*. - Pin VS at operating level. *Attention: Driver may reduce chopper frequency during AT#1. Use reduced standstill current IHOLD<IRUN to prevent extended periods of time at lower chopper frequency*	≤ 2^20+2*2^18 t_{CLK}, ≤ 130ms (with internal clock)
AT#2	*PWM_ GRAD_AUTO*	- Move motor at a velocity, where a significant amount of back EMF is generated and where the full run current can be reached. Conditions: - 1.5 * *PWM_OFS_AUTO* < *PWM_SCALE_SUM* < 4 * *PWM_OFS_AUTO* - *PWM_SCALE_SUM* < 255. *Hint:* A typical range is 60-300 RPM.	8 fullsteps are required for a change of +/-1. For a typical motor with *PWM_GRAD_AUTO* optimum at 50 or less, up to 400 fullsteps are required when starting from default value 0.

Determine best conditions for automatic tuning with the evaluation board.
Monitor *PWM_SCALE_AUTO* going down to zero during the constant velocity phase in AT#2 tuning. This indicates a successful tuning.

Attention:
Operating in stealthChop without proper tuning can lead to high motor currents during a deceleration ramp, especially with low resistive motors and fast deceleration settings. Follow the automatic tuning process and check optimum tuning conditions using the evaluation board. It is recommended to use an initial value for settings *PWM_OFS* and *PWM_GRAD* determined per motor type.
Protect the power stage and supply by additionally tuning the overcurrent protection.

Known Limitations:
Successful completion of AT#1 tuning phase is not safely detected by the TMC5161. It will require multiple motor start / stop events to safely detect completion.
Successful determination is mandatory for AT#2: Tuning of *PWM_GRAD* will not start when AT#1 has not completed.
Successful completion of AT#1 and AT#2 only can be checked by monitoring *PWM_SCALE_AUTO* approaching 0 during AT#2 motion.

Solution a):
Complete automatic tuning phase AT#1 process, by using a slow-motion sequence which leads to standstill detection in between of each two steps. Use a velocity of 8 (6 Hz) or lower and execute minimum 10 steps during AT#1 phase.

Solution b):
Store initial parameters for *PWM_GRAD_AUTO* for the application. Therefore, use the motor and operating conditions determined for the application and do a complete automatic tuning sequence (refer to *a)*). Store the resulting *PWM_GRAD_AUTO* value and use it for initialization of *PWM_GRAD*. With this, tuning of AT#2 phase is not mandatory in the application and can be skipped. Automatic tuning will further optimize settings during operation. Combine with a) if desired.

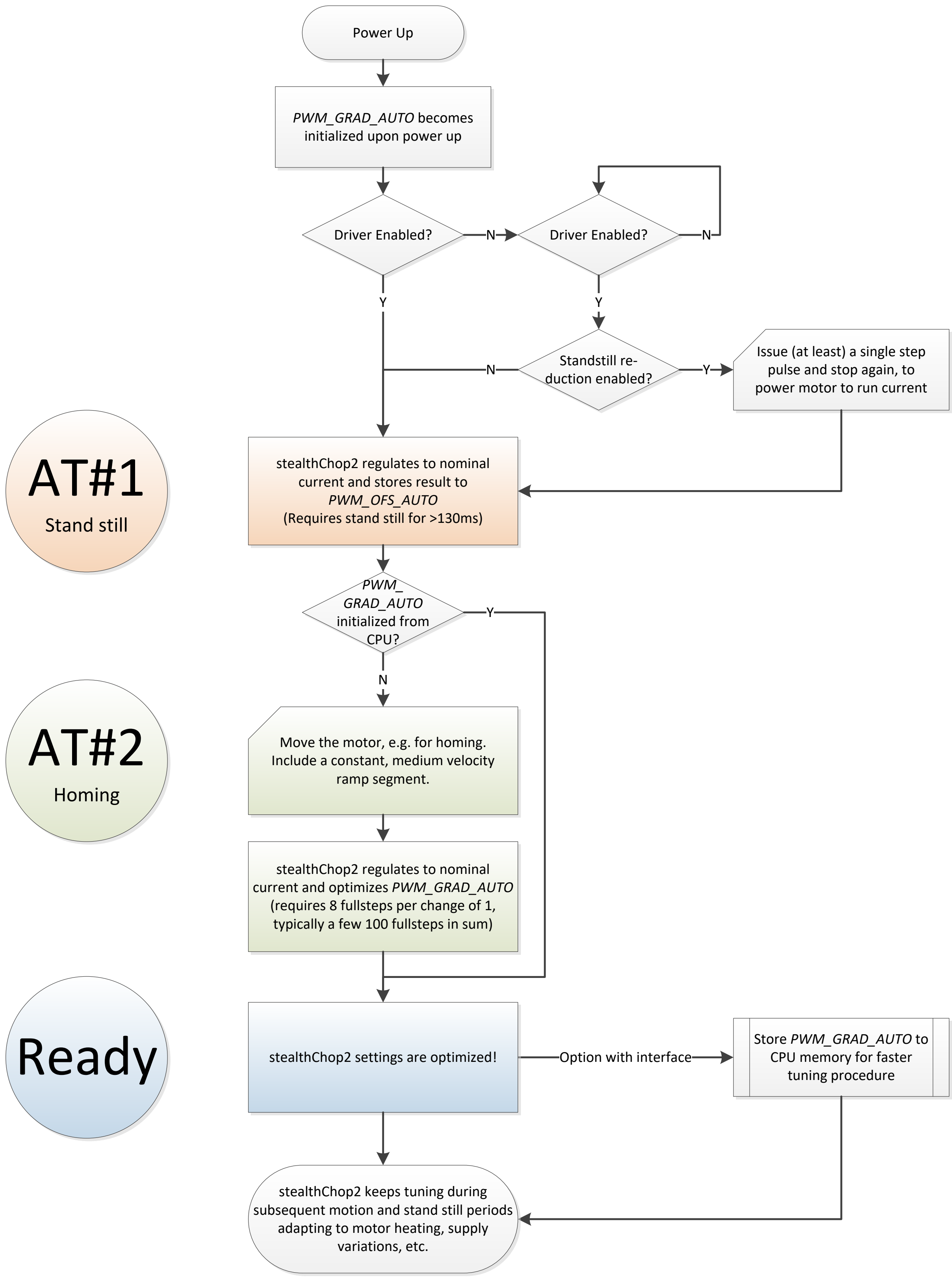

Figure 6.2 stealthChop2 automatic tuning procedure

Attention
Modifying *GLOBALSCALER* or VS voltage invalidates the result of the automatic tuning process. Motor current regulation cannot compensate significant changes until next AT#1 phase. Automatic tuning adapts to changed conditions whenever AT#1 and AT#2 conditions are fulfilled in the later operation.

6.2 stealthChop Options

In order to match the motor current to a certain level, the effective PWM voltage becomes scaled depending on the actual motor velocity. Several additional factors influence the required voltage level to drive the motor at the target current: The motor resistance, its back EMF (i.e. directly proportional to its velocity) as well as the actual level of the supply voltage. Two modes of PWM regulation are provided: The automatic tuning mode (AT) using current feedback (*pwm_autoscale* = 1, *pwm_autograd* = 1) and a feed forward velocity controlled mode (*pwm_autoscale* = 0). The feed forward velocity controlled mode will not react to a change of the supply voltage or to events like a motor stall, but it provides very stable amplitude. It does not use nor require any means of current measurement. This is perfect when motor type and supply voltage are well known. Therefore we recommend the automatic mode, unless current regulation is not satisfying in the given operating conditions.

It is recommended to operate in automatic tuning mode.

Non-automatic mode (*pwm_autoscale*=0) should be taken into account only with well-known motor and operating conditions. In this case, careful programming via the interface is required. The operating parameters *PWM_GRAD* and *PWM_OFS* can be determined in automatic tuning mode initially.

The stealthChop PWM frequency can be chosen in four steps in order to adapt the frequency divider to the frequency of the clock source. A setting in the range of 20-50kHz is good for most applications. It balances low current ripple and good higher velocity performance vs. dynamic power dissipation.

CHOICE OF PWM FREQUENCY FOR STEALTHCHOP				
Clock frequency f_{CLK}	**PWM_FREQ=%00** f_{PWM}=2/1024 f_{CLK}	**PWM_FREQ=%01** f_{PWM}=2/683 f_{CLK}	**PWM_FREQ=%10** f_{PWM}=2/512 f_{CLK}	**PWM_FREQ=%11** f_{PWM}=2/410 f_{CLK}
18MHz	35.2kHz	52.7kHz	70.3kHz	87.8kHz
16MHz	31.3kHz	46.9kHz	62.5kHz	78.0kHz
12MHz (internal)	23.4kHz	35.1kHz	46.9kHz	58.5kHz
10MHz	19.5kHz	29.3kHz	39.1kHz	48.8kHz
8MHz	15.6kHz	23.4kHz	31.2kHz	39.0kHz

Table 6.1 Choice of PWM frequency – green / light green: recommended

6.3 stealthChop Current Regulator

In stealthChop voltage PWM mode, the autoscaling function (*pwm_autoscale* = 1, *pwm_autograd* = 1) regulates the motor current to the desired current setting. Automatic scaling is used as part of the automatic tuning process (AT), and for subsequent tracking of changes within the motor parameters. The driver measures the motor current during the chopper on time and uses a proportional regulator to regulate *PWM_SCALE_AUTO* in order match the motor current to the target current. *PWM_REG* is the proportionality coefficient for this regulator. Basically, the proportionality coefficient should be as small as possible in order to get a stable and soft regulation behavior, but it must be large enough to allow the driver to quickly react to changes caused by variation of the motor target current (e.g. change of VREF). During initial tuning step AT#2, *PWM_REG* also compensates for the change of motor velocity. Therefore, a high acceleration during AT#2 will require a higher setting of *PWM_REG*. With careful selection of homing velocity and acceleration, a minimum setting of the regulation gradient often is sufficient (*PWM_REG*=1). *PWM_REG* setting should be optimized for the fastest required acceleration and deceleration ramp (compare Figure 6.3 and Figure 6.4). The quality of the setting *PWM_REG* in phase AT#2 and the finished automatic tuning procedure (or non-automatic settings for *PWM_OFS* and *PWM_GRAD*) can be examined when monitoring motor current during an acceleration phase Figure 6.5.

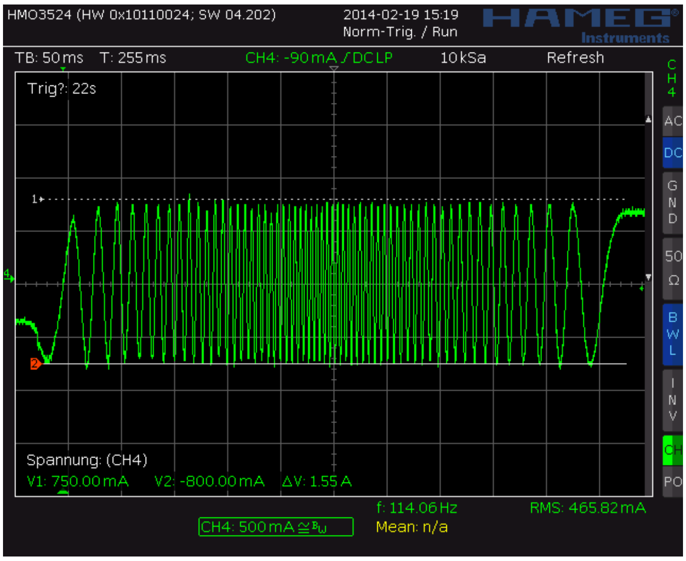

Figure 6.3 Scope shot: good setting for PWM_REG

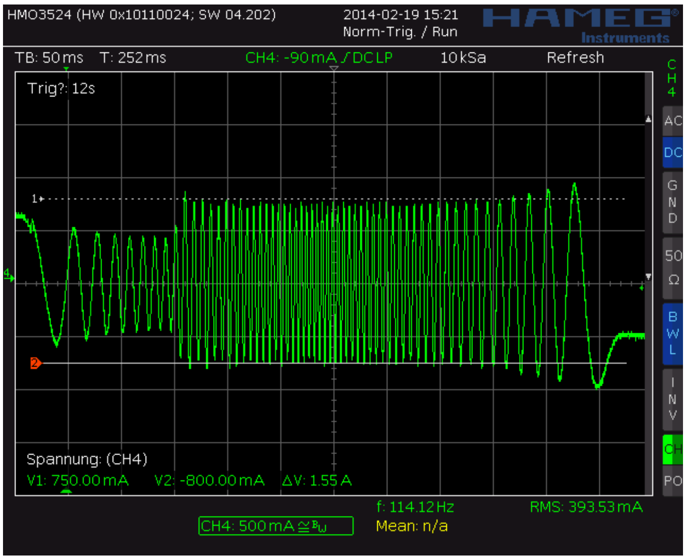

Figure 6.4 Scope shot: too small setting for PWM_REG during AT#2

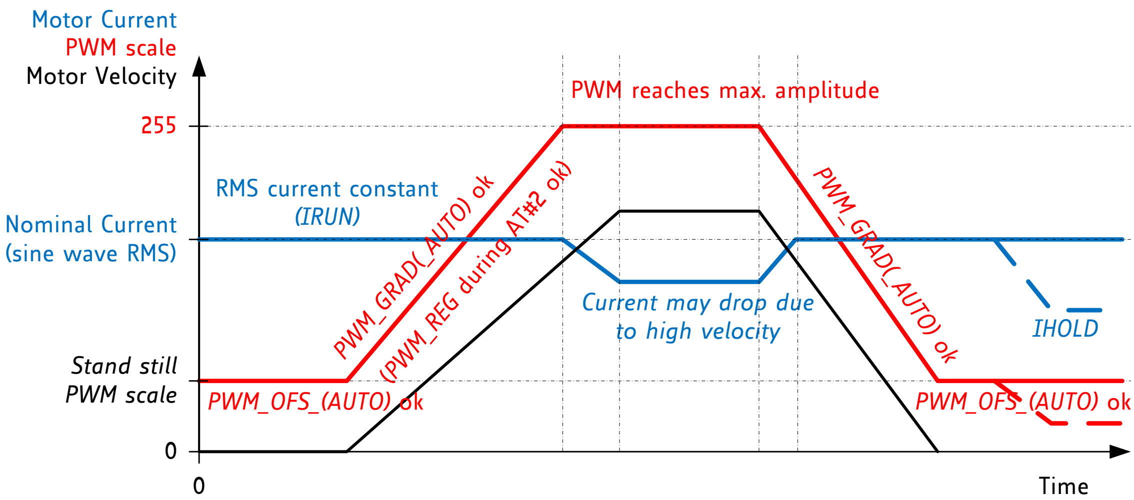

Figure 6.5 Successfully determined PWM_GRAD(_AUTO) and PWM_OFS(_AUTO)

Quick Start
For a quick start, see the Quick Configuration Guide in chapter 18.

6.3.1 Lower Current Limit

The stealthChop current regulator imposes a lower limit for motor current regulation. As the coil current can be measured in the shunt resistor during chopper on phase only, a minimum chopper duty cycle allowing coil current regulation is given by the blank time as set by *TBL* and by the chopper frequency setting. Therefore, the motor specific minimum coil current in stealthChop autoscaling mode rises with the supply voltage and with the chopper frequency. A lower blanking time allows a lower current limit. It is important for the correct determination of *PWM_OFS_AUTO*, that in AT#1 the run current set by the sense resistor, *GLOBALSCALER* and *IRUN* is well within the regulation range. Lower currents (e.g. for standstill power down) are automatically realized based on *PWM_OFS_AUTO* and *PWM_GRAD_AUTO* respectively based on *PWM_OFS* and *PWM_GRAD* with non-automatic current scaling. The freewheeling option allows going to zero motor current.

Lower motor coil current limit for stealthChop2 automatic tuning:

$$I_{Lower\ Limit} = t_{BLANK} * f_{PWM} * \frac{V_M}{R_{COIL}}$$

With V_M the motor supply voltage and R_{COIL} the motor coil resistance.
$I_{Lower\ Limit}$ can be treated as a thumb value for the minimum nominal *IRUN* motor current setting. In case the lower current limit is not sufficient to reach the desired setting, the driver will retry with a lower chopper frequency in step AT#1, only.
f_{PWM} is the chopper frequency as determined by setting *PWM_FREQ*. In AT#1, the driver tries a lower, (roughly half frequency), in case it cannot reach the current. The frequency will remain active in standstill, while currentscale *CS=IRUN*. With automatic standstill reduction, this is a short moment.

EXAMPLE:

A motor has a coil resistance of 5Ω, the supply voltage is 24V. With *TBL*=%01 and *PWM_FREQ*=%00, t_{BLANK} is 24 clock cycles, f_{PWM} is 2/(1024 clock cycles):

$$I_{Lower\ Limit} = 24\ t_{CLK} * \frac{2}{1024\ t_{CLK}} * \frac{24V}{5\Omega} = \frac{24}{512} * \frac{24V}{5\Omega} = 225mA$$

This means, the motor target current for automatic tuning must be 225mA or more, taking into account all relevant settings. This lower current limit also applies for modification of the motor current via the *GLOBALSCALER*.

Attention
For automatic tuning, a lower coil current limit applies. The motor current in automatic tuning phase AT#1 must exceed this lower limit. $I_{LOWER\ LIMIT}$ can be calculated or measured using a current probe. Setting the motor run-current or hold-current below the lower current limit during operation by modifying *IRUN* and *IHOLD* is possible after successful automatic tuning.

The lower current limit also limits the capability of the driver to respond to changes of *GLOBALSCALER*.

6.4 Velocity Based Scaling

Velocity based scaling scales the stealthChop amplitude based on the time between each two steps, i.e. based on *TSTEP*, measured in clock cycles. This concept basically does not require a current measurement, because no regulation loop is necessary. A pure velocity based scaling is available via programming, only, when setting *pwm_autoscale* = 0. The basic idea is to have a linear approximation of the voltage required to drive the target current into the motor. The stepper motor has a certain coil resistance and thus needs a certain voltage amplitude to yield a target current based on the basic formula I=U/R. With R being the coil resistance, U the supply voltage scaled by the PWM value, the current I results. The initial value for *PWM_OFS* can be calculated:

$$PWM\_OFS = \frac{374 * R_{COIL} * I_{COIL}}{V_M}$$

With V_M the motor supply voltage and I_{COIL} the target RMS current

The effective PWM voltage U_{PWM} (1/SQRT(2) x peak value) results considering the 8 bit resolution and 248 sine wave peak for the actual PWM amplitude shown as *PWM_SCALE*:

$$U_{PWM} = V_M * \frac{PWM\_SCALE}{256} * \frac{248}{256} * \frac{1}{\sqrt{2}} = V_M * \frac{PWM\_SCALE}{374}$$

With rising motor velocity, the motor generates an increasing back EMF voltage. The back EMF voltage is proportional to the motor velocity. It reduces the PWM voltage effective at the coil resistance and thus current decreases. The TMC2160 provides a second velocity dependent factor (*PWM_GRAD*) to compensate for this. The overall effective PWM amplitude (*PWM_SCALE_SUM*) in this mode automatically is calculated in dependence of the microstep frequency as:

$$PWM\_SCALE\_SUM = PWM\_OFS + PWM\_GRAD * 256 * \frac{f_{STEP}}{f_{CLK}}$$

With f_{STEP} being the microstep frequency for 256 microstep resolution equivalent
and f_{CLK} the clock frequency supplied to the driver or the actual internal frequency

As a first approximation, the back EMF subtracts from the supply voltage and thus the effective current amplitude decreases. This way, a first approximation for *PWM_GRAD* setting can be calculated:

$$PWM\_GRAD = C_{BEMF}\left[\frac{V}{\frac{rad}{s}}\right] * 2\pi * \frac{f_{CLK} * 1.46}{V_M * MSPR}$$

C_{BEMF} is the back EMF constant of the motor in Volts per radian/second.
MSPR is the number of microsteps per rotation, e.g. 51200 = 256µsteps multiplied by 200 fullsteps for a 1.8° motor.

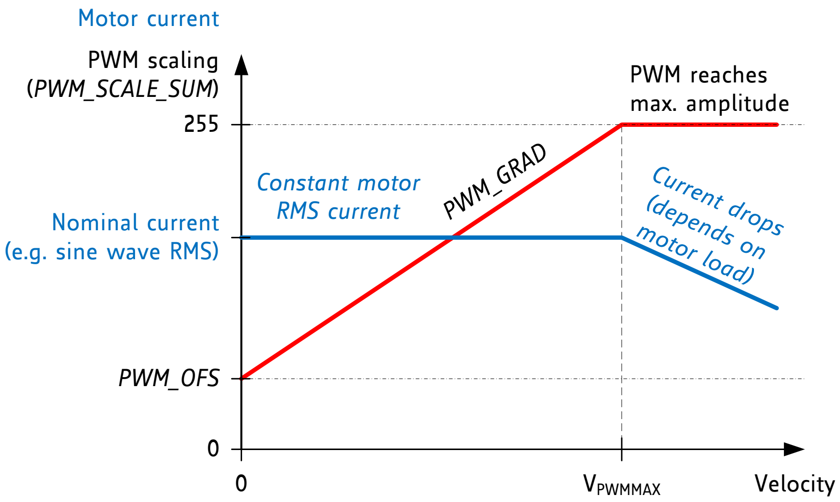

Figure 6.6 Velocity based PWM scaling (pwm_autoscale=0)

> *Hint*
> The values for *PWM_OFS* and *PWM_GRAD* can easily be optimized by tracing the motor current with a current probe on the oscilloscope. Alternatively, automatic tuning determines these values and they can be read out from *PWM_OFS_AUTO* and *PWM_GRAD_AUTO*.

UNDERSTANDING THE BACK EMF CONSTANT OF A MOTOR

The back EMF constant is the voltage a motor generates when turned with a certain velocity. Often motor datasheets do not specify this value, as it can be deducted from motor torque and coil current rating. Within SI units, the numeric value of the back EMF constant C_{BEMF} has the same numeric value as the numeric value of the torque constant. For example, a motor with a torque constant of 1 Nm/A would have a C_{BEMF} of 1V/rad/s. Turning such a motor with 1 rps (1 rps = 1 revolution per second = 6.28 rad/s) generates a back EMF voltage of 6.28V. Thus, the back EMF constant can be calculated as:

$$C_{BEMF}\left[\frac{V}{rad/s}\right] = \frac{HoldingTorque[Nm]}{2 * I_{COILNOM}[A]}$$

$I_{COILNOM}$ is the motor's rated phase current for the specified holding torque
HoldingTorque is the motor specific holding torque, i.e. the torque reached at $I_{COILNOM}$ on both coils. The torque unit is [Nm] where 1Nm = 100Ncm = 1000mNm.
The voltage is valid as RMS voltage per coil, thus the nominal current is multiplied by 2 in this formula, since the nominal current assumes a full step position, with two coils operating.

6.5 Combining stealthChop and spreadCycle

For applications requiring high velocity motion, spreadCycle may bring more stable operation in the upper velocity range. To combine no-noise operation with highest dynamic performance, the TMC2160 allows combining stealthChop and spreadCycle based on a velocity threshold (Figure 6.7). With this, stealthChop is only active at low velocities.

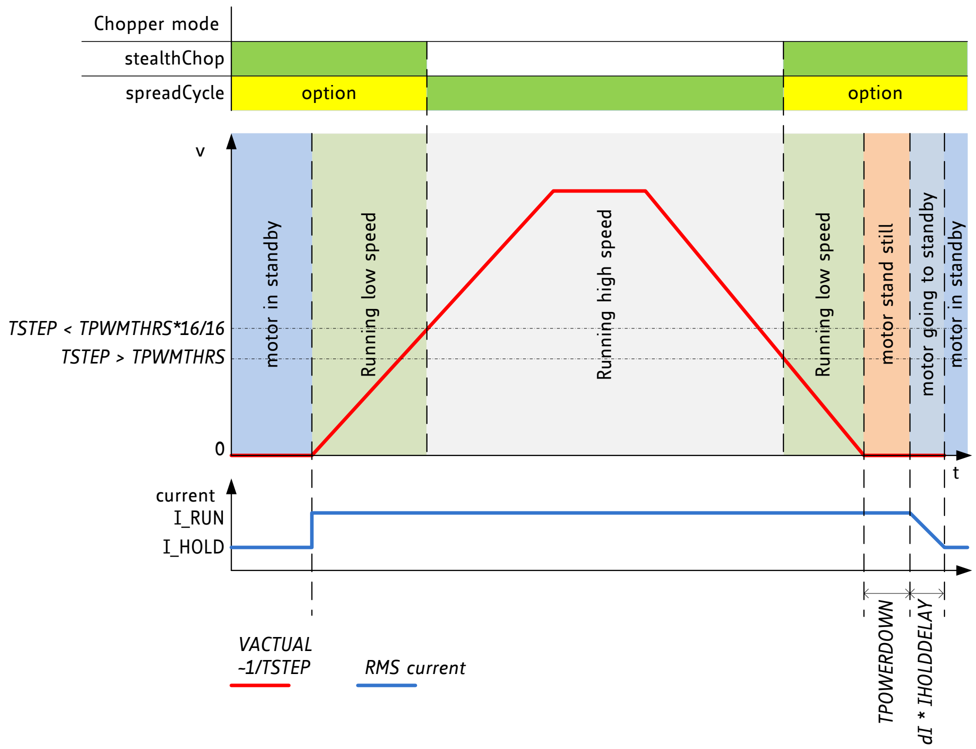

Figure 6.7 TPWMTHRS for optional switching to spreadCycle

As a first step, both chopper principles should be parameterized and optimized individually. In a next step, a transfer velocity has to be fixed. For example, stealthChop operation is used for precise low speed positioning, while spreadCycle shall be used for highly dynamic motion. *TPWMTHRS* determines the transition velocity. Read out *TSTEP* when moving at the desired velocity and program the resulting value to *TPWMTHRS*. Use a low transfer velocity to avoid a jerk at the switching point.

A jerk occurs when switching at higher velocities, because the back-EMF of the motor (which rises with the velocity) causes a phase shift of up to 90° between motor voltage and motor current. So when switching at higher velocities between voltage PWM and current PWM mode, this jerk will occur with increased intensity. A high jerk may even produce a temporary overcurrent condition (depending on the motor coil resistance). At low velocities (e.g. 1 to a few 10 RPM), it can be completely neglected for most motors. Therefore, consider the switching jerk when choosing *TPWMTHRS*. Set *TPWMTHRS* zero if you want to work with stealthChop only.

When enabling the stealthChop mode the first time using automatic current regulation, the motor must be at stand still in order to allow a proper current regulation. When the drive switches to stealthChop at a higher velocity, stealthChop logic stores the last current regulation setting until the motor returns to a lower velocity again. This way, the regulation has a known starting point when returning to a lower velocity, where stealthChop becomes re-enabled. Therefore, neither the velocity threshold nor the supply voltage must be considerably changed during the phase while the chopper is switched to a different mode, because otherwise the motor might lose steps or the instantaneous current might be too high or too low.

A motor stall or a sudden change in the motor velocity may lead to the driver detecting a short circuit or to a state of automatic current regulation, from which it cannot recover. Clear the error flags and restart the motor from zero velocity to recover from this situation.

Hint
Start the motor from standstill when switching on stealthChop the first time and keep it stopped for at least 128 chopper periods to allow stealthChop to do initial standstill current control.

6.6 Flags in stealthChop

As stealthChop uses voltage mode driving, status flags based on current measurement respond slower, respectively the driver reacts delayed to sudden changes of back EMF, like on a motor stall.

Attention
A motor stall, or abrupt stop of the motion during operation in stealthChop can lead to a overcurrent condition. Depending on the previous motor velocity, and on the coil resistance of the motor, it significantly increases motor current for a time of several 10ms. With low velocities, where the back EMF is just a fraction of the supply voltage, there is no danger of triggering the short detection.

Hint
Tune low side driver overcurrent detection to safely trigger upon motor stall, when using stealthChop. This will avoid high peak current draw from the power supply.

6.6.1 Open Load Flags

In stealthChop mode, status information is different from the cycle-by-cycle regulated spreadCycle mode. OLA and OLB show if the current regulation sees that the nominal current can be reached on both coils.

- A flickering OLA or OLB can result from asymmetries in the sense resistors or in the motor coils.
- An interrupted motor coil leads to a continuously active open load flag for the coil.
- One or both flags are active, if the current regulation did not succeed in scaling up to the full target current within the last few fullsteps (because no motor is attached or a high velocity exceeds the PWM limit).

If desired, do an on-demand open load test using the spreadCycle chopper, as it delivers the safest result. With stealthChop, *PWM_SCALE_SUM* can be checked to detect the correct coil resistance.

6.6.2 PWM_SCALE_SUM Informs about the Motor State

Information about the motor state is available with automatic scaling by reading out *PWM_SCALE_SUM*. As this parameter reflects the actual voltage required to drive the target current into the motor, it depends on several factors: motor load, coil resistance, supply voltage, and current setting. Therefore, an evaluation of the *PWM_SCALE_SUM* value allows checking the motor operation point. When reaching the limit (255), the current regulator cannot sustain the full motor current, e.g. due to a drop in supply volage.

6.7 Freewheeling and Passive Braking

stealthChop provides different options for motor standstill. These options can be enabled by setting the standstill current *IHOLD* to zero and choosing the desired option using the *FREEWHEEL* setting. The desired option becomes enabled after a time period specified by *TPOWERDOWN* and *IHOLD_DELAY*. Current regulation becomes frozen once the motor target current is at zero current in order to ensure a quick startup. With the freewheeling options, both freewheeling and passive braking can be realized. Passive braking is an effective eddy current motor braking, which consumes a minimum of energy, because no active current is driven into the coils. However, passive braking will allow slow turning of the motor when a continuous torque is applied.

Hint
Operate the motor within your application when exploring stealthChop. Motor performance often is better with a mechanical load, because it prevents the motor from stalling due mechanical oscillations which can occur without load.

PARAMETERS RELATED TO STEALTHCHOP			
Parameter	**Description**	**Setting**	**Comment**
en_spread_cycle	General disable for use of stealthChop (register *GCONF*). The input SPREAD is XORed to this flag.	1	Do not use stealthChop
		0	stealthChop enabled
TPWMTHRS	Specifies the upper velocity for operation in stealthChop. Entry the *TSTEP* reading (time between two microsteps) when operating at the desired threshold velocity.	0 … 1048575	stealthChop is disabled if *TSTEP* falls *TPWMTHRS*
PWM_LIM	Limiting value for limiting the current jerk when switching from spreadCycle to stealthChop. Reduce the value to yield a lower current jerk.	0 … 15	Upper four bits of 8 bit amplitude limit (Default=12)
pwm_autoscale	Enable automatic current scaling using current measurement. If off, use forward controlled velocity-based mode.	0	Forward controlled mode
		1	Automatic scaling with current regulator
pwm_autograd	Enable automatic tuning of *PWM_GRAD_AUTO*	0	disable, use *PWM_GRAD* from register instead
		1	enable
PWM_FREQ	PWM frequency selection. Use the lowest setting giving good results. The frequency measured at each of the chopper outputs is half of the effective chopper frequency f_{PWM}.	0	f_{PWM}=2/1024 f_{CLK}
		1	f_{PWM}=2/683 f_{CLK}
		2	f_{PWM}=2/512 f_{CLK}
		3	f_{PWM}=2/410 f_{CLK}
PWM_REG	User defined PWM amplitude regulation loop P-coefficient. A higher value leads to a higher adaptation speed when *pwm_autoscale*=1.	1 … 15	Results in 0.5 to 7.5 steps for *PWM_SCALE_AUTO* regulator per fullstep
PWM_OFS	User defined PWM amplitude (offset) for velocity based scaling and initialization value for automatic tuning of *PWM_OFFS_AUTO*.	0 … 255	*PWM_OFS*=0 disables linear current scaling based on current setting
PWM_GRAD	User defined PWM amplitude (gradient) for velocity based scaling and initialization value for automatic tuning of *PWM_GRAD_AUTO*.	0 … 255	
FREEWHEEL	Stand still option when motor current setting is zero (*I_HOLD*=0). Only available with stealthChop enabled. The freewheeling option makes the motor easy movable, while both coil short options realize a passive brake.	0	Normal operation
		1	Freewheeling
		2	Coil short via LS drivers
		3	Coil short cia HS drivers
PWM_SCALE_AUTO	Read back of the actual stealthChop voltage PWM scaling correction as determined by the current regulator. Shall regulate close to 0 during tuning.	-255 … 255	(read only) Scaling value becomes frozen when operating in spreadCycle
PWM_GRAD_AUTO *PWM_OFS_AUTO*	Allow monitoring of the automatic tuning and determination of initial values for *PWM_OFS* and *PWM_GRAD*.	0 … 255	(read only)
TOFF	General enable for the motor driver, the actual value does not influence stealthChop	0	Driver off
		1 … 15	Driver enabled
TBL	Comparator *blank time*. This time needs to safely cover the switching event and the duration of the ringing on the sense resistor. Choose a setting of 1 or 2 for typical applications. For higher capacitive loads, 3 may be required. Lower settings allow stealthChop to regulate down to lower coil current values.	0	16 t_{CLK}
		1	24 t_{CLK}
		2	36 t_{CLK}
		3	54 t_{CLK}