15 dcStep

dcStep is an automatic commutation mode for the stepper motor. It allows the stepper to run with its target velocity as commanded by the Step signal, as long as it can cope with the load. In case the motor becomes overloaded, it slows down to a velocity, where the motor can still drive the load. This way, the stepper motor never stalls and can drive heavy loads as fast as possible. Its higher torque available at lower velocity, plus dynamic torque from its flywheel mass allow compensating for mechanical torque peaks. In case the motor becomes completely blocked, the stall flag becomes set.

15.1 User Benefits

Motor	–	never loses steps
Application	–	works as fast as possible
Acceleration	–	automatically as high as possible
Energy efficiency	–	highest at speed limit
Cheaper motor	–	does the job!

15.2 Designing-In dcStep

In a classical application, the operation area is limited by the maximum torque required at maximum application velocity. A safety margin of up to 50% torque is required, in order to compensate for unforeseen load peaks, torque loss due to resonance and aging of mechanical components. dcStep allows using up to the full available motor torque. Even higher short time dynamic loads can be overcome using motor and application flywheel mass without the danger of a motor stall. With dcStep the nominal application load can be extended to a higher torque only limited by the safety margin near the holding torque area (which is the highest torque the motor can provide). Additionally, maximum application velocity can be increased up to the actually reachable motor velocity.

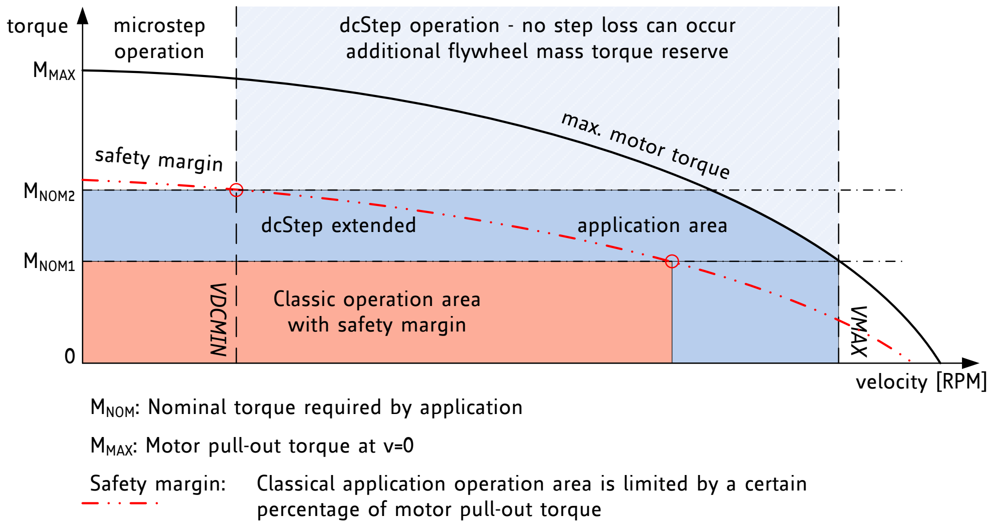

Figure 15.1 dcStep extended application operation area

Quick Start
For a quick start, see the Quick Configuration Guide in chapter 18.
For detail configuration procedure see Application Note AN003 - *dcStep*

15.3 Stall Detection in dcStep Mode

While dcStep is able to decelerate the motor upon overload, it cannot avoid a stall in every operation situation. Once the motor is blocked, or it becomes decelerated below a motor dependent minimum velocity where the motor operation cannot safely be detected any more, the motor may stall and loose steps. A stallGuard2 load value also is available during dcStep operation. The range of values is limited to 0 to 255, in certain situations up to 511 will be read out. In order to enable stallGuard, also set *TCOOLTHRS* corresponding to a velocity slightly above *VDCMIN* or up to *VMAX*.

Stall detection in this mode may trigger falsely due to resonances, when flywheel loads are loosely coupled to the motor axis.

Parameter	Description	Range	Comment
vhighfs & *vhighchm*	These chopper configuration flags in *CHOPCONF* need to be set for dcStep operation. As soon as *VDCMIN* becomes exceeded, the chopper becomes switched to fullstepping.	0 / 1	set to 1 for dcStep
TOFF	dcStep often benefits from an increased off time value in *CHOPCONF*. Settings >2 should be preferred.	2… 15	Settings 8…15 do not make any difference to setting 8 for dcStep operation.
VDCMIN	This is the lower threshold for dcStep operation when using internal ramp generator. Below this threshold, the motor operates in normal microstep mode. In dcStep operation, the motor operates at minimum *VDCMIN*, even when it is completely blocked. Tune together with *DC_TIME* setting. Activation of stealthChop also disables dcStep.	0… 2^22	0: Disable dcStep Set to the lower velocity limit for dcStep operation.
DC_TIME	This setting controls the reference pulse width for dcStep load measurement. It must be optimized for robust operation with maximum motor torque. A higher value allows higher torque and higher velocity, a lower value allows operation down to a lower velocity as set by *VDCMIN*. Check best setting under nominal operation conditions, and re-check under extreme operating conditions (e.g. lowest operation supply voltage, highest motor temperature, and highest supply voltage, lowest motor temperature).	0… 1023	Lower limit for the setting is: t_{BLANK} (as defined by *TBL*) in clock cycles + *n* with *n* in the range 1 to 100 (for a typical motor)
DC_SG	This setting controls stall detection in dcStep mode. Increase for higher sensitivity. A stall can be used as an error condition by issuing a hard stop for the motor. Stop the motor upon an impulse on the stall output (configure DIAG0 or DIAG1 to signal a stall). This way the motor will be stopped once it stalls.	0… 255	Set slightly higher than *DC_TIME* / 16

15.4 dcStep with STEP/DIR Interface

The TMC2160 provides two ways to use dcStep when interfaced to an external motion controller. The first way gives direct control of the dcStep step execution to the external motion controller, which must react to motor overload and is allowed to override a blocked motor situation. The second way assumes that the external motion controller cannot directly react to dcStep signals. The TMC2160 automatically reduces the motor velocity or stops the motor upon overload. In order to allow the motion controller to react to the reduced real motor velocity in this mode, the counter *LOST_STEPS* gives the number of steps which have been commanded, but not taken by the motor controller. The motion controller can later on read out *LOST_STEPS* and drive any missing number of steps. In case of a blocked motor it tries moving it with the minimum velocity as programmed by *VDCMIN*.

Enabling dcStep automatically sets the chopper to constant TOFF mode with slow decay only. This way, no re-configuration is required when switching from microstepping mode to dcStep and back.

dcStep operation is controlled by three pins in STEP and DIR mode:

- DCEN – Forces the driver to dcStep operation if high. A velocity-based activation of dcStep is controlled by *TPWMTHRS* when using stealthChop operation for low velocity settings. In this case, dcStep is disabled while in stealthChop mode, i.e. at velocities below the stealthChop switching velocity.
- DCO – Informs the motion controller when motor is not ready to take a new step (low level). The motion controller shall react by delaying the next step until DCO becomes high. The sequencer can buffer up to the effective number of microsteps per fullstep to allow the motion controller to react to assertion of DCO. In case the motor is blocked this wait-situation can be terminated after a timeout by providing a long > 1024 clock STEP input, or via the internal *VDCMIN* setting.
- DCIN – Commands the driver to wait with step execution and to disable DCO. This input can be used for synchronization of multiple drivers operating with dcStep.

15.4.1 Using *LOST_STEPS* for dcStep Operation

This is the simplest possibility to integrate dcStep with an external motion controller: The external motion controller enables dcStep using DCEN or the internal velocity threshold. The TMC2160 tries to follow the steps. In case it needs to slow down the motor, it counts the difference between incoming steps on the STEP signal and steps going to the motor. The motion controller can read out the difference and compensate for the difference after the motion or on a cyclic basis. Figure 15.2 shows the principle (simplified).

In case the motor driver needs to postpone steps due to detection of a mechanical overload in dcStep, and the motion controller does not react to this by pausing the step generation, *LOST_STEPS* becomes incremented or decremented (depending on the direction set by DIR) with each step which is not taken. This way, the number of lost steps can be read out and executed later on or be appended to the motion. As the driver needs to slow down the motor while the overload situation persists, the application will benefit from a high microstepping resolution, because it allows more seamless acceleration or deceleration in dcStep operation. In case the application is completely blocked, *VDCMIN* sets a lower limit to the step execution. If the motor velocity falls below this limit, however an unknown number of steps is lost and the motor position is not exactly known any more. DCIN allows for step synchronization of two drivers: it stops the execution of steps if low and sets DCO low.

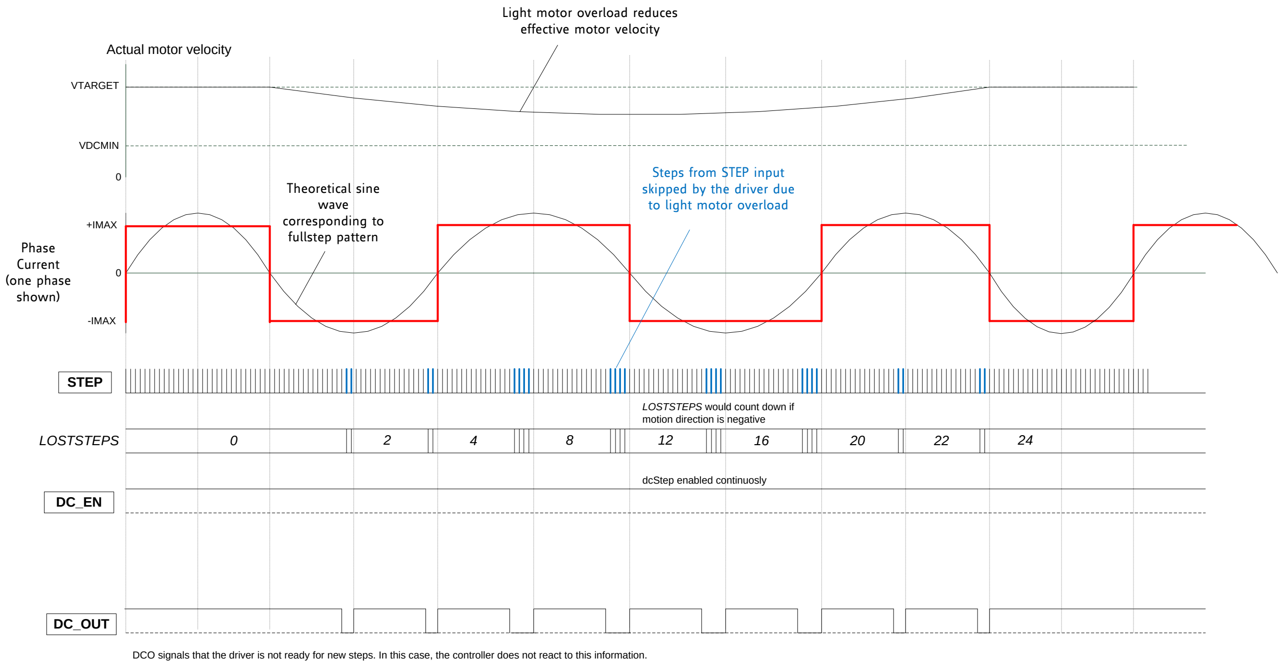

Figure 15.2 Motor moving slower than STEP input due to light overload. LOSTSTEPS incremented

15.4.2 DCO Interface to Motion Controller

In STEP/DIR mode, DCEN enables dcStep. It is up to the external motion controller to enable dcStep either, once a minimum step velocity is exceeded within the motion ramp, or to use the automatic threshold *VDCMIN* for dcStep enable.

The STEP/DIR interface works in microstep resolution, even if the internal step execution is based on fullstep. This way, no switching to a different mode of operation is required within the motion controller. The dcStep output DCO signals if the motor is ready for the next step based on the dcStep measurement of the motor. If the motor has not yet mechanically taken the last step, this step cannot be executed, and the driver stops automatically before execution of the next fullstep. This situation is signaled by DCO. The external motion controller shall stop step generation if DCOUT is low and wait until it becomes high again. Figure 15.4 shows this principle. The driver buffers steps during the waiting period up to the number of microstep setting minus one. In case, DCOUT does not go high within the lower step limit time e.g. due to a severe motor overload, a step can be enforced: override the stop status by a long STEP pulse with min. 1024 system clocks length. When using internal clock, a pulse length of minimum 125µs is recommended.

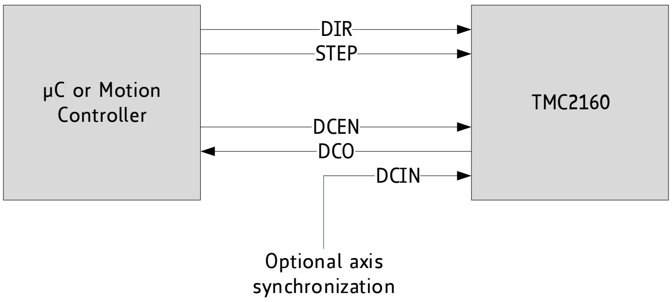

Figure 15.3 Full signal interconnection for dcStep

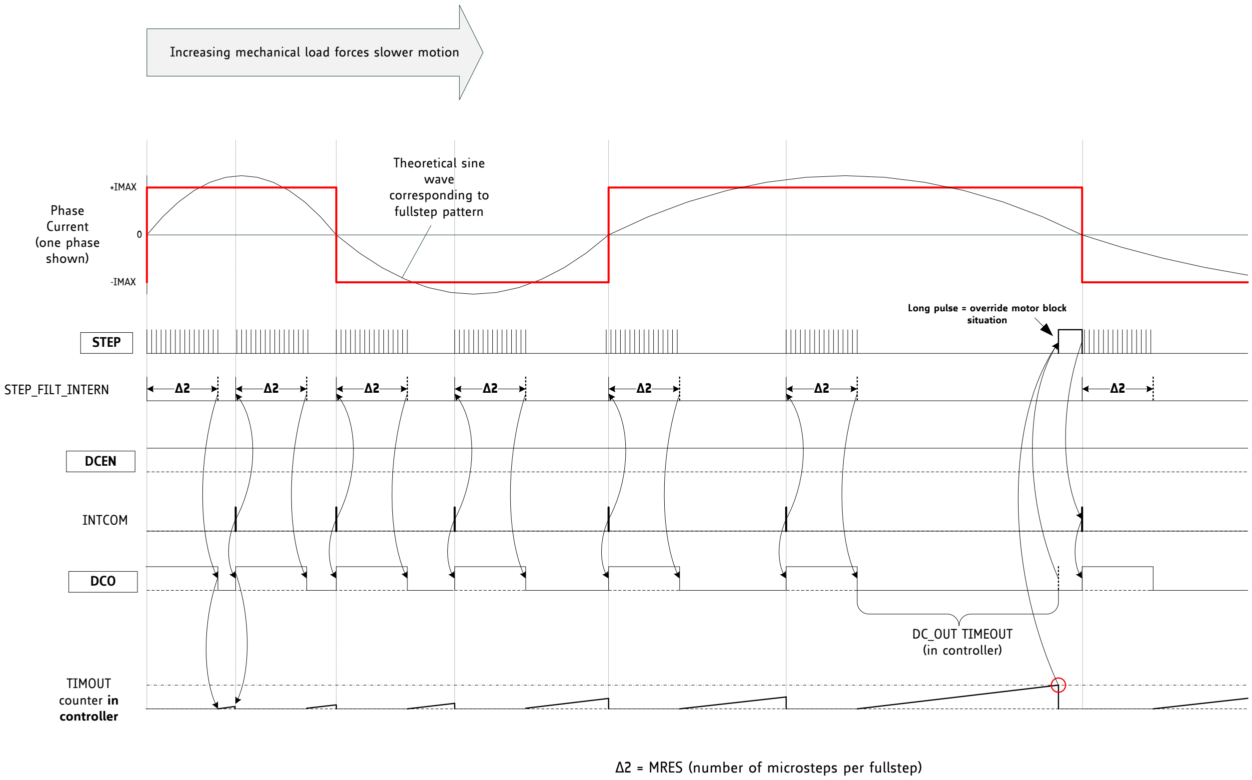

Figure 15.4 DCO Interface to motion controller – step generator stops when DCO is asserted