

## **ADDENDUM:**

### **The following changes will be advertized in the next new version**

Content of the changes: (TOTAL-2)

---

Change #-1

Date May 2008

Subject: Wrong calculation in the Command Reference Manual, Chapter EF [N] - Encoder Filter Frequency.

Eng.: Alexander Yankelev (MTCR # 00-100-11)

Location: <http://www.elmomc.com/support/manuals/MAN-SIMCR.pdf>

(Page 3-45)

**New Change:**

### **EF[N] - Encoder Filter Frequency**

**Purpose:**

Filters encoder signal in order to improve its noise immunity. Because the logic of the quadrature decoder must sense transitions, the inputs are first run through a glitch filter. This filter has a digital delay line that samples four time points on the signal and verifies that a majority of the samples are at a new state before outputting the new state to the internal logic. The sample rate of this delay line is programmable, to adapt to a variety of signal bandwidths.

When an analog encoder is used, the basic signal, before interpolation, is filtered using the same method.

EF[N] sets the sample rate of the corresponding digital glitch filter: EF[1] for the main encoder and EF[2] for the auxiliary encoder. A counter increases or decreases to the value of EF[N]. When the count reaches the specified value, the counter is reset and the filter takes a new sample of the raw A, B, Index and Home input signals. If EF[N] is zero the digital filter is bypassed.

If EF[N] is large, the encoder reading noise immunity will be better, but true fast transitions (occurring by fast speed) may be dismissed as false. A number that is too small may cause the counting of noise pulses.

A good value for the required delay of the encoder filter is  $\frac{1}{4}$  of the minimum time expected between transitions.

**Example:**

Suppose that the maximum speed of a motor is 10,000 rpm and that the motor is equipped with an encoder with 1000 lines (4000 counts/rev with resolution multiplication). The expected minimum encoder transition time is:

$$\frac{60 \text{ sec/min}}{4000 \text{ cpr} * 10,000 \text{ rpm}} = 1.5 \mu\text{sec}$$

The encoder pulse time is calculated as  $\frac{1.5 \mu\text{sec}}{4} = 375 \text{ nsec}$  and because of CPU clock frequency, minimum required encoder signal stable time should be set to 400 nsec (i.e. EF[] = 3)

The minimum required encoder signal stable time should be set to about:

$$\frac{1.5 \mu\text{sec}}{4} = 400 \text{ nsec}$$

The ranges of encoder frequency filtering are as follows:

EF[1]/EF[2]	0	1	2	...	K	127
Filter time	25 nsec	200 nsec	300 nsec	...	100*(K+1) nsec	12.8 μsec



When the interpolator is used, the time difference between consecutive signal changes may be shorter. This is dictated by the interpolator's specifications and not from the motor speed.

Let use an interpolated encoder for the above example. Suppose that the maximum speed of a motor remains 10,000 rpm and that the motor is equipped with an interpolated encoder with 2048 analog sine/cosine periods per mechanical revolution. In addition, the interpolation x32 (CA[31]=5) is used and feedback resolution is formed as  $2048 * 2^5 = 65536 \text{ counts/rev}$ .

The expected minimum encoder (analog sine/cosine signals without interpolation) transition time is:

$$\frac{60 \text{ sec/min}}{2048 * 4 * 10,000 \text{ rpm}} = 732.4 \text{ nsec}$$

The encoder pulse time (sine/cosine signal cutting by hardware comparator)

is calculated as  $\frac{732.4 \text{ nsec}}{4} = 183.11 \text{ nsec}$  and because of CPU clock

frequency, minimum required encoder signal stable time should be set to 200 nsec (i.e. EF[] = 1)

<b>Attributes:</b>	<b>Type:</b>	Parameter, Integer
	<b>Source:</b>	Program, RS-232, CANopen
	<b>Restrictions:</b>	None
	<b>Default values:</b>	0 (RS), Non-volatile
	<b>Range:</b>	[0...127]
	<b>Index range:</b>	[1, 2]
	<b>Unit modes:</b>	All
	<b>Activation:</b>	Immediate

**Change #-2**

**Date** JULY 2008

**Subject:** Command Reference Manual isn't described what ID and IQ mean.

**Eng.:** Moshe Barazani / Ilia Volgust (MTCR # 00-100-24)

**Location:** <http://www.elmomc.com/support/manuals/MAN-SIMCR.pdf>

(Page 3-66)

**New Change:**

## **IQ, ID - Read Active Current and Reactive Current**

**Purpose:**

**Gets the active (IQ) and the reactive (ID) components** of the motor current, in amperes.

A brushless motor carries alternating currents in its phases. The alternating currents in the motor phases create a rotating magnetic field, which can be projected in two directions. The first magnetic field component is aligned with the magnetic direction of the rotor; it produces no mechanical torque. The other magnetic field component is perpendicular to the magnetic direction of the rotor and produces all the mechanical torque.

IQ[Amp] is the component of the motor phase current that creates effective torque. The current controller attempts to make IQ equal to the current command. ID is the component of the motor phase current that does *not* create torque. The current controller tries to null ID.



When the motor is off (MO=0), IQ and ID are not calculated and return zero.

<b>Attributes:</b>	<b>Type:</b>	Status report, Real
	<b>Source:</b>	Program, RS-232, CANopen
	<b>Restrictions:</b>	None
	<b>Unit modes:</b>	All

**See also:**

[AN\[N\]](#), [MC](#), [PL\[N\]](#), [CL\[N\]](#)

**Reference chapter in the *SimplIQ Software Manual*:**

Chapter 10, "The Current Controller"