A Simple and Accurate Algorithm for Speed Measurement in Electric Drives Using Incremental Encoder

Francesco Brugnano, Carlo Concari, Emir Imamovic, Filippo Savi, Andrea Toscani, Roberto Zanichelli
Dipartimento di Ingegneria dell'Informazione
University of Parma
Parco Area delle Scienze, 181/A, I-43124 Parma, Italy
francesco.brugnano@studenti.unipr.it, carlo.concari@unipr.it, emir.imamovic@unipr.it, filippo.savi2@studenti.unipr.it, andrea.toscani@unipr.it, roberto.zanichelli@studenti.unipr.it

***Abstract*—Incremental encoders and their use in electric drives are nowadays considered standard and not given much thought. Nevertheless, speed measurement through the signals coming from an incremental encoder can be subject to errors in some operating conditions if special care is not taken. Techniques exist in literature to enhance the precision of speed measurement, but they often require special hardware (i.e. FPGA) or a non-negligible portion of computing power. This paper proposes a novel technique for precise speed measurement, with minimal error, that can be implemented on a low-cost microcontroller with standard quadrature decoder peripheral.**

Keywords—Electric drives, encoder, speed measurement.

I. Introduction

Incremental encoders are widely used in commercial motor drives because they allow for very precise position and speed measurement without excessively increasing system cost.

The usage of incremental encoders is usually taken for granted, nevertheless, incorrect use of an incremental encoder can result in unexpectedly large errors.

Speed errors can result from inherent encoder imperfections or from incorrect use of their output signals. Encoder imperfections are usually due to manufacturing tolerances, and can't be eliminated. One approach towards reducing their effects is the time-stamping of encoder pulses, but it's also a costly procedure that has to be repeated for every motor-encoder assembly, and therefore it's unlikely to be undertaken for low-cost industrial motor drives [1].

Speed is usually calculated from encoder signals as the ratio of angular path ($\Delta\vartheta$) over time (Δt), usually keeping one of the two quantities constant (period measurement, also called fixed space, FS, or frequency measurement, also called fixed time, FT). Determining speed is a task with many degrees of freedom and can result in significant errors if not carried out properly. Many different techniques exist to enhance the precision of speed measurement [2]. The simplest approach is the use of some sort of (usually low-pass) filtering on the speed, which can increase the resolution of the measurement [2], [3]. The drawback in this case is a reduction of the dynamic response due to the resultant feedback delay.

Signal processing techniques such as adaptive windowing can give very good results, but often require abundant computing power or dedicated hardware such as implementation on FPGAs [4]-[6], which come at a premium in terms of cost. Another solution is the use of very high frequency counters [7], that require also a large number of counter bits in order to be able to measure an adequately low minimum speed; such facilities aren't usually found in low-cost microcontrollers.

Considering cost, techniques implemented on standard microcontrollers are preferable. One of the simplest is adaptive FS [8], that allows a maximum guaranteed speed error, although not the minimum theoretical error. Hybrid FS/FT with compensation of the time displacement between encoder pulses and timer pulses achieves the minimum theoretical error [9], [10], but involves multiple timers and requires very precise sequencing of measurement tasks, which can interfere with the execution of other real-time tasks on the same microcontroller.

This paper presents a period based technique with adaptive timer time-base, which ensures very good precision within an extended speed range together with a very simple microcontroller implementation.

II. Frequency and Period Measurement Comparison

A. Frequency measurement

A simple method to estimate the rotor speed is the measure of the frequency of the encoder pulses, this frequency results proportional to the angular speed.

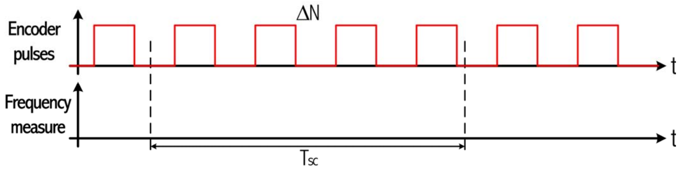

Fig. 1. Frequency measure.

The measure of the frequency is obtained counting the encoder pulses ΔN during a fixed gate time T_{SC} (Fig. 1), the angular speed ω is given by:

$$\omega = \frac{2\pi \cdot \Delta N}{N_p \cdot T_{sc}} \tag{1}$$

where N_p is the number of the pulses per revolution (ppr) of the encoder shaft.

The maximum angular speed estimation error $\Delta\omega$ due to the angular quantization error is:

$$\Delta\omega = \frac{2\pi \cdot (\Delta N + 1)}{N_p \cdot T_{sc}} - \frac{2\pi \cdot \Delta N}{N_p \cdot T_{sc}} = \frac{2\pi}{N_p \cdot T_{sc}} \tag{2}$$

so, the percentage error can be computed as follow:

$$e_\omega = \frac{\Delta\omega}{\omega} = \frac{2\pi}{\omega \cdot N_p \cdot T_{sc}} [\%] \tag{3}$$

as it can be seen from (3) the measuring error is inversely proportional to the angular speed ω. At relatively low speed the percentage error becomes unacceptable.

B. *Period measurement*

This technique measures the period of the encoder pulses, using a time-base signal with period T_{hf}. It is necessary to count the time-base signal pulses using a digital counter (Fig. 2).

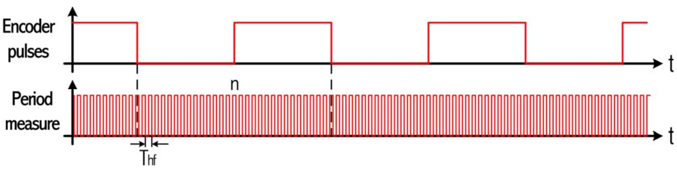

Fig. 2. Period measure.

The angular speed ω is given by:

$$\omega = \frac{2\pi}{N_p \cdot n \cdot T_{hf}} \tag{4}$$

where n is the number of the pulses of the time-base signal. The maximum angular speed estimation error $\Delta\omega$ due to the quantization error is:

$$\Delta\omega = \frac{2\pi}{N_p \cdot n \cdot T_{hf}} - \frac{2\pi}{N_p \cdot (n+1) \cdot T_{hf}} = \frac{2\pi \cdot [(n+1) - n]}{n \cdot (n+1) \cdot N_p \cdot T_{hf}} = \\ = \frac{2\pi}{n \cdot (n+1) \cdot N_p \cdot T_{hf}} = \frac{\omega}{n+1} \tag{5}$$

and the percentage error can be computed as:

$$e_\omega = \frac{\omega}{\omega \cdot (n+1)} \cong \frac{\omega}{\omega \cdot n} = \frac{\omega \cdot N_p \cdot T_{hf}}{2\pi} [\%] \tag{6}$$

as it can be seen from (6) the measuring error is proportional to the angular speed ω, so if the speed decreases, the percentage error also decreases, but if the speed drops under a certain speed ω_{MIN} the counter that counts the time-base pulses overflows. This limit speed is given by:

$$\omega_{MIN} = \frac{2\pi}{N_p \cdot T_{hf} \cdot 2^{nbit}} \tag{7}$$

where *nbit* is the number of bits of the counter.

The higher the time-base frequency ($f_{hf} = 1/T_{hf}$), the lower the percentage error, but also the higher the minimum speed ω_{MIN}. It is clear that there is a trade-off between the measurement error and the minimum estimable speed.

To obtain a certain ω_{MIN} value, the time-base frequency f_{hf} must respect the following relation:

$$f_{hf} \le \frac{N_p \cdot \omega_{MIN} \cdot 2^{nbit}}{2\pi} \tag{8}$$

C. *Frequency and period measurement comparison*

The two methods have been compared in terms of angular speed percentage error. As shown in (3) and (6), using frequency measurement this error is inversely proportional to the estimated speed, while using period measurement it is directly proportional to the estimated speed. There is a critical speed ω_c that leads to an equal estimation error (Fig. 3):

$$\frac{2\pi}{\omega \cdot N_p \cdot T_{sc}} = \frac{\omega \cdot N_p \cdot T_{hf}}{2\pi} \rightarrow \omega_c = \frac{2\pi}{N_p}\sqrt{\frac{1}{T_{sc} \cdot T_{tf}}} \tag{9}$$

Most electrical drives for industrial applications feature a nominal speed up to 6000 rpm and encoder with up to 1024 pulses per revolution (ppr). Usually, the control algorithm needs a refresh rate for the speed estimation of at least 1 kHz, equivalent to $T_{sc} = 1\,\text{ms}$. A low-cost motor control-oriented microcontroller provides timers with a resolution of 16 bit, that are able to count a time-base frequency at least up to 84 MHz, so f_{hf} can be as high as 84 MHz. Using (9) it results $\omega_c = 1778$ rad/s (about 16,982 rpm).

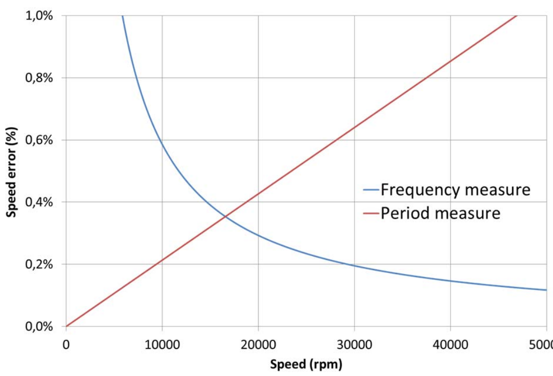

Fig. 3. Frequency measure and period measure percentage error comparison.

For most electrical drives, period measurement performs better than frequency measurement over the whole speed range. Period measurement also gives the maximum estimated speed refresh rate, especially at high speed range.

A possible drawback is that using high value of time-base frequency leads to a high value of ω_{MIN}. In the considered conditions, with *nbit* = 16, it results:

$$\omega_{MIN} = \frac{2\pi}{1024 \cdot 11.905 \cdot 10^{-9} \cdot 2^{16}} \cong 7.865 \text{ rad/s} \qquad (10)$$

III. The Proposed Algorithm

To overcome the trade-off between the resolution of the measure and the minimum estimable speed, a new algorithm is proposed in this paper.

The basic idea is to change the time-base frequency according to the value of the speed. As the speed decreases below a certain limit, the time-base frequency decreases.

Obviously, the threshold is set with a hysteresis that avoids rebounds between two adjacent values of frequency, as shown in Fig. 4.

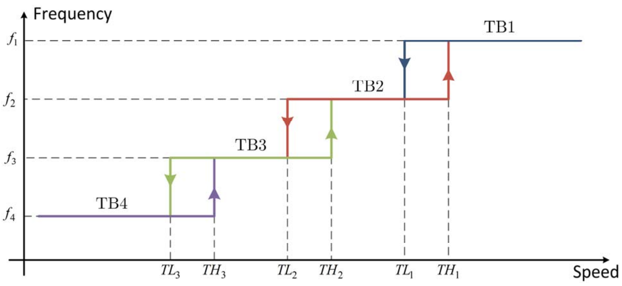

Fig. 4. Time-base frequency usage as function of speed (TB1, TB2, TB3 and TB4 are different time-bases).

The different time-base values are obtained using a hardware prescaler that divides, usually by a power of two, the main clock frequency of the timer of the microcontroller. Tab. I summarizes a possible solution.

TABLE I. Time-Base Frequency Usage

Time-base frequency [Hz]		*Minimun speed [rad/s]*	*Percentage error [%]*
TB1	84,000,000	7.865	$1.94 \cdot 10^{-6} \cdot \omega$
TB2	10,500,000	0.983	$15.5 \cdot 10^{-6} \cdot \omega$
TB3	1,312,500	0.123	$124.3 \cdot 10^{-6} \cdot \omega$
TB4	164,062.5	0.015	$993.4 \cdot 10^{-4} \cdot \omega$

The algorithm is implemented using the hardware features of the timer peripheral that allow a very low time-consuming implementation. All the necessary instructions are located inside the interrupt service routine (ISR) of the timer. This interrupt is generated whenever an edge of a signal of the encoder occurs. In case of high motor speed combined with a high resolution encoder, the frequency of the interrupt may be reduced using hardware features of the timer peripheral, which allow to generate interrupts on the occurrence of a programmable number of encoder pulses. The first operation of the ISR is to store the counter value (it will be used in the following to compute the speed), then the counter is reset, so that it is ready for a new measurement. If during the previous execution it resulted necessary to change the time-base, this task is done, and finally the speed is computed using the previously stored value of the counter. Then, the future time-base frequency is decided using the new estimated speed (it will be fixed during the next execution of the ISR), in compliance with a precise algorithm that considers a series of thresholds. This sequence of operations is depicted in Fig. 5.

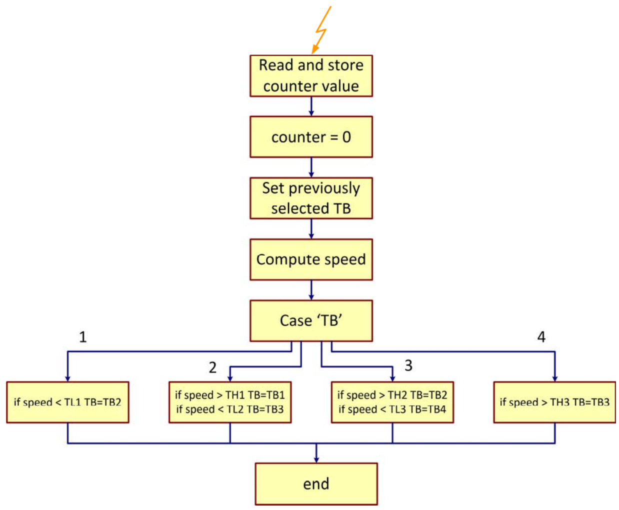

Fig. 5. Sequence of operations executed inside the ISR of the timer, for the speed estimation task.

Using this algorithm a low percentage error for the speed estimation and a low minimum speed can both be achieved, as shown in Fig. 6.

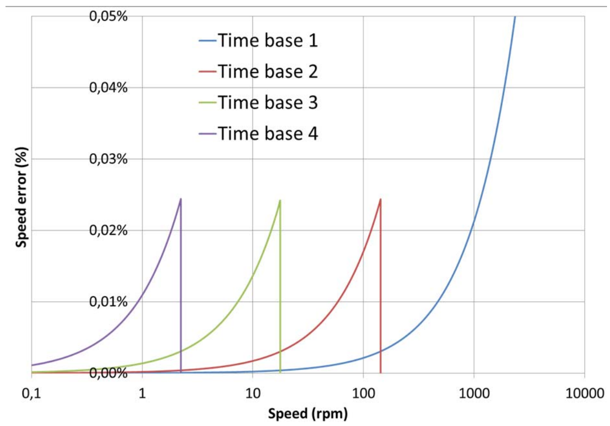

Fig. 6. Speed estimation percentage error.

As can be seen, the speed estimation percentage error is less than 0.03% over a wide speed range including low speed values.

IV. Simulations

Simulations in MATLAB/Simulink environment were used in order to evaluate the proposed algorithm. This simulation

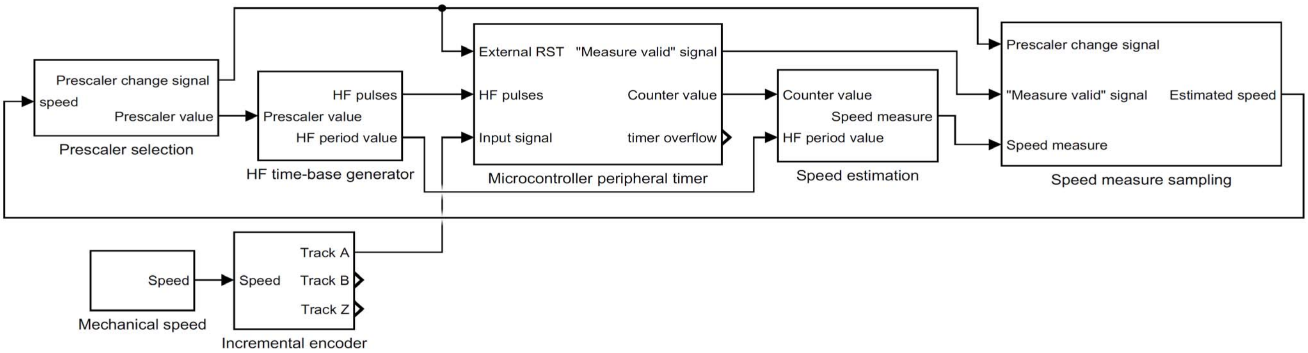

Fig. 7. Block scheme of the simulation.

relies on a general microcontroller timer peripheral, that is able to count high frequency pulses. A generic timer for motor control can measure internally-generated pulses (time-base signal) that occur between edges of input signals. It is also possible to reduce the frequency of time-base pulses using an internal prescaler. All these functions are totally supported by the peripheral hardware resources so this is a low time-consuming task.

As previously discussed, the algorithm has the purpose of varying the frequency of the time-base pulses used for the period measurement. This helps reducing the minimum measurable speed and minimizing the speed error.

The simulation is composed using blocks that model the various hardware and firmware parts that are used to implement this algorithm (Fig. 7).

The first block establishes the prescaler value used to generate the time-base signal (HF pulses) according to the estimated speed. Then a block replicates a general microcontroller timer peripheral behavior that counts the number of pulses that occur between two encoder edges (input signal). The "speed estimation" block, using relation (4), computes the speed. This speed is sampled by the following block that latches the value and triggers the calculation of the speed when needed by the control logic to guarantee the consistency of output data. All the parameters of the simulation that can influence the measure, as encoder pulse per revolution value, time-base frequency and the possible prescaler values, are accessible and tunable through a MATLAB script.

In order to prove the benefits of the proposed algorithm in terms of precision of the estimated speed, two different simulation scenarios were considered: the first with a constant prescaler value for the high frequency timer and the second one with the introduction of a variable prescaler as proposed in this paper. In both cases a particular speed reference composed by traits with constant speed alternated with traits at constant acceleration starting from zero was used (Fig. 8).

The speed estimation is not valid until the first period between two consecutive edges is measured and captured from the timer peripheral. In this case, also the simulation takes into account of the fact that the speed increases with an acceleration that cannot be infinite, so the speed is not constant, and that introduces another error in the speed estimation.

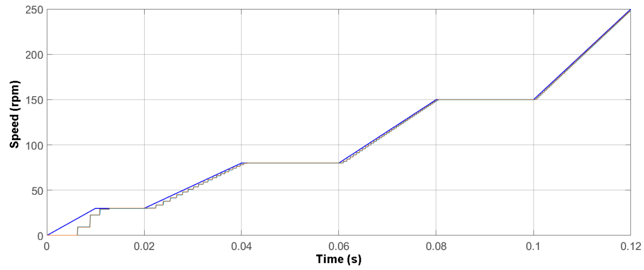

Fig. 8. Speed reference and estimated speed.

During the constant speed traits, the only error is due to the precision of the measure so it is possible to make valid comparisons between the algorithms. As it can be seen in Fig. 9 with the prescaler variation algorithm, in the constant traits the speed measure has an error that is negligible, and in the acceleration traits it suffers only from the error due to speed variations during the measurement period.

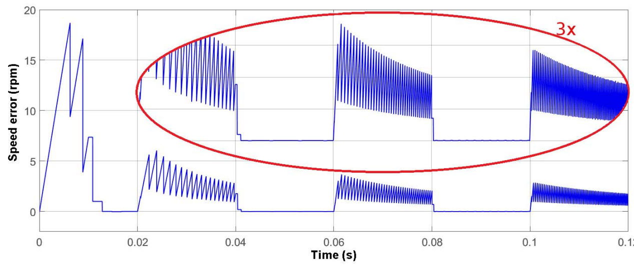

Fig. 9. Speed estimation error in case of variable prescaler value.

Considering instead the fixed prescaler, it is possible to see in Fig. 10, the effect of the loss of resolution in the speed estimation that increases with the speed and also has a higher effect on acceleration traits.

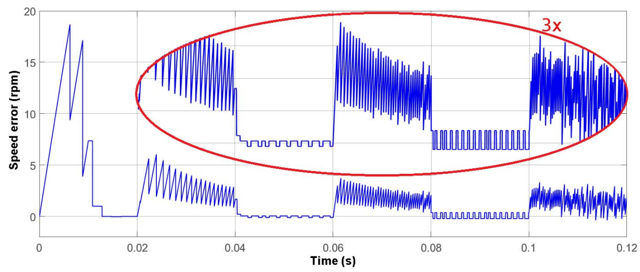

Fig. 10. Speed estimation error in case of fixed prescaler value.

Fig. 11 shows a performance comparison between two period measure strategies: the traditional fixed time-base frequency (also called fixed prescaler value), and the proposed algorithm that change the time-base frequency using a prescaler (also called variable prescaler value) with the aim of obtaining the best performance for every speed value over the whole required speed range.

In the proposed study-case the speed range is from 4 to 4000 rpm and the encoder features 1024 ppr. With a fixed time-base frequency, using (8), f_{hf} results 4.47 MHz, so the prescaler value must be fixed at least at 32, to avoid the overflow of the timer counter register. Right column of fig. 11 shows the speed estimation for four different speed values in the previous conditions. Obviously in e), f), g) and h) charts the prescaler is equal to 32.

Vice versa using the variable prescaler strategy, as it can be seen from the left column of fig. 11, the prescaler value changes from 1 in charts a) and b) to 4 in chart c) and 32 in chart d). So doing the overall performances achieved in terms of measure accuracy are better.

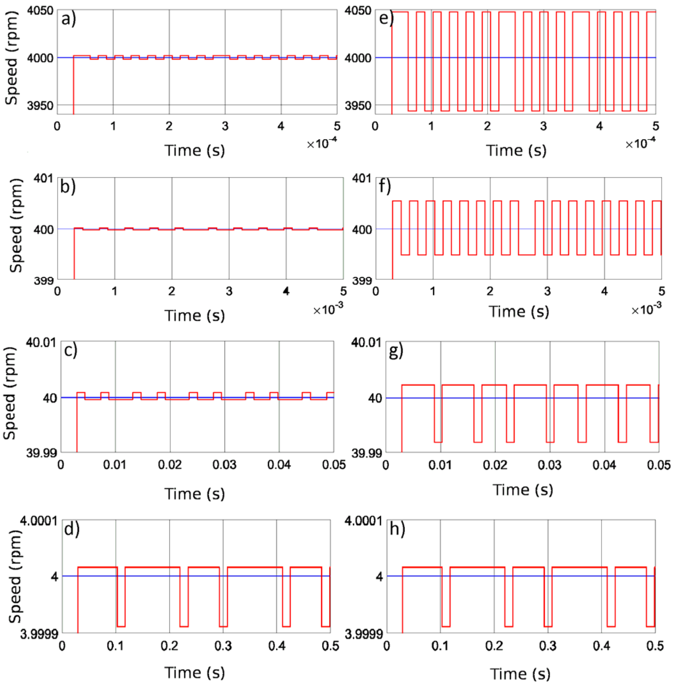

Fig. 11. Comparison between the variable prescaler (left column) and the fixed prescaler (right column) period measurement for different speed values. For a) and b) charts the prescaler is equal to 1 (frequency 84MHz). For c) chart the prescaler is equal to 4 (frequency 21MHz). For d), e), f), g) and h) the prescaler is equal to 32 (frequency 2.625 MHz).

Using the proposed technique a very good performance in term of speed estimation accuracy can be achieved over the whole speed range.

V. Experimental Results

The algorithm described in paragraph III was proved on a custom hardware platform. The processor unit used for the experimental tests was an ARM Cortex M4F core located inside an NXP Kinetis MKV46F256 MCU, with a timer peripheral that reflects the features of the generic one used for the simulation. Table II summarizes the hardware setup characteristics. Fig. 12 shows the comparison between the frequency and the period measurement method during an acceleration ramp from 0 to about 90 rpm. It is immediate to see that for the observed speed range and with the conditions listed in table II the period measurement gives a better speed estimation both in terms of resolution and update frequency.

TABLE II. Hardware Setup Data

Quantity	***Value***
Timer clock frequency (f_{hf})	84,000,000 Hz
Timer period (T_{hs})	11.905 ns
Prescaler value range	0 ÷128; only 2^n, n ∈ (0, 7) values are possible
Counter register resolution (*nbit*)	16 bits
Gate time (T_s)	1 ms
Encoder pulses per revolution value (N_p)	2048

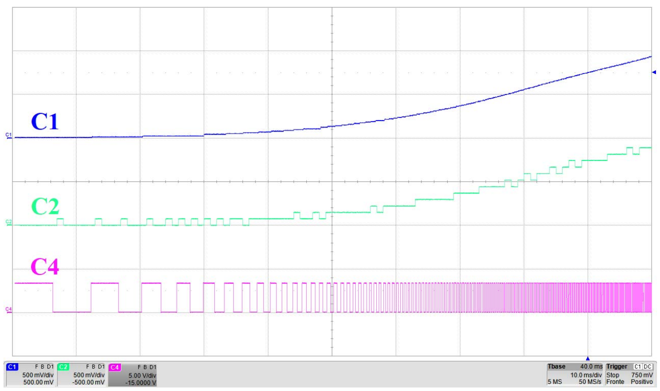

Fig. 12. Comparison between period measurement (C1, blue trace) and frequency measurement (C2 green trace). Y-scale: 50 rpm/div; X-scale: 10 ms/div. All measurements obtained using encoder signals (C4, violet trace).

Fig. 13 shows the performances of frequency measurement for two different gate time values. As it can be seen higher gate time values lead to an intrinsic delay when the speed is not constant. On the other hand as stated in (3) higher gate-time values guarantees a better accuracy of the estimated value.

Figs. 14 and 15 show the speed measurement error when using the period measurement method for different speeds without changing the time-base frequency.

The two figures clearly indicate that the measurement error increases with speed, accordingly with (6). This limitation can be overcome using the proposed algorithm, which tunes the prescaler value and consequently the time-base frequency in order to have the smaller error over the whole speed range.

Fig. 16 shows how the algorithm acts during an acceleration ramp: at zero speed the prescaler is set to the higher value (minimum frequency) and as the speed increases it is reduced in three steps to the lower value (maximum frequency).

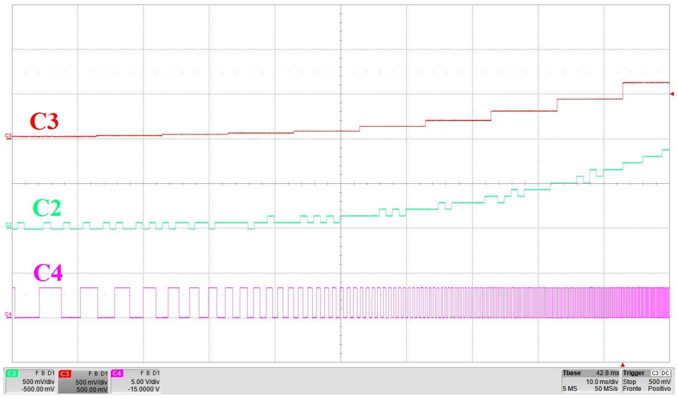

Fig. 13. Frequency measurement. Comparison between two different gate time values (C3, red trace: 10ms; C2, green trace: 1ms). Y-scale: 50 rpm/div; X-scale: 10 ms/div. Trace C4, violet: one encoder channel.

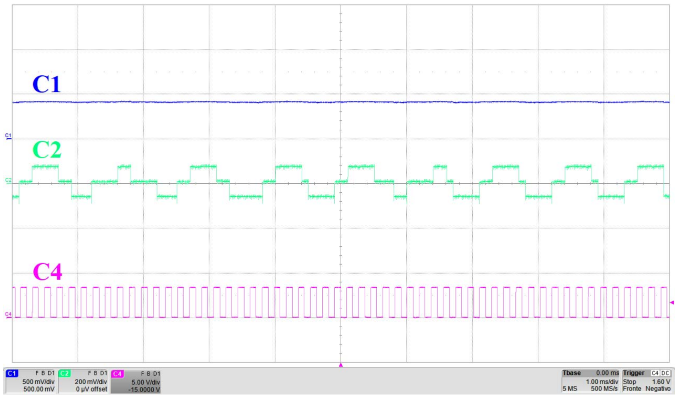

Fig. 14. Speed estimation error (C2, green trace, 2rpm/div) in case of fixed prescaler value equal to 128 at constant speed of 80 rpm (C1, blue trace, 100rpm/div). X-scale: 1 ms/div. Trace C4, violet: one encoder channel.

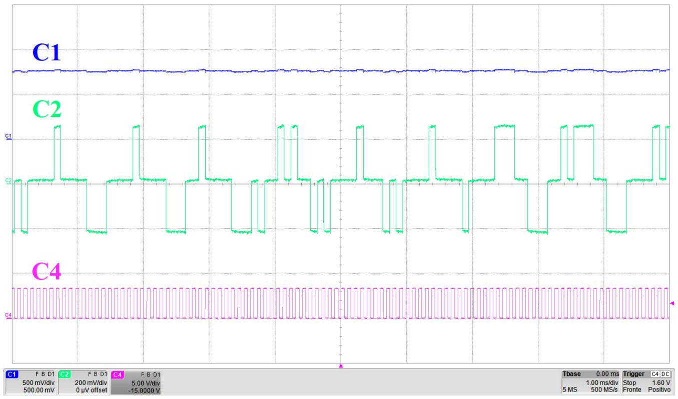

Fig. 15. Speed estimation error (C2, green trace, 2rpm/div) in case of fixed prescaler value equal to 128 at constant speed of 150 rpm (C1, blue trace, 100 rpm/div). X-scale: 1 ms/div. Trace C4, violet: one encoder channel.

VI. Conclusion

This paper presented a simple but effective technique for accurate speed measurement through incremental encoder over an extended speed range. The proposed technique was compared with other techniques and validated through simulations and experiments. The simplicity of the proposed technique makes it suitable for implementation on industry-oriented motor drives based on low-cost microcontrollers.

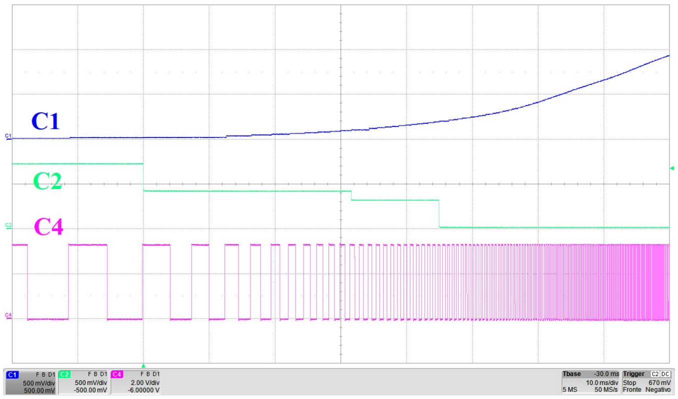

Fig. 16. Speed estimation using the proposed algorithm (C1, blue trace, 50 rpm/div) with variable prescaler value (C2, green trace 5 unit/div). Trace C4, violet: one encoder channel. Prescaler values represented as powers of two.

References

[1] R. Merry, R. van de Molengraft and M. Steinbuch, “Error modeling and improved position estimation for optical incremental encoders by means of time stamping,” 2007 American Control Conference, New York, NY, 2007, pp. 3570-3575.

[2] A. C. Negrea, M. Imecs, I. I. Incze, A. Pop and C. Szabo, “Error compensation methods in speed identification using incremental encoder,” 2012 International Conference and Exposition on Electrical and Power Engineering, Iasi, 2012, pp. 441-445.

[3] A. F. Ilmiawan, D. Wijanarko, A. H. Arofat, H. Hindersyah and A. Purwadi, “An easy speed measurement for incremental rotary encoder using multi stage moving average method,” 2014 International Conference on Electrical Engineering and Computer Science (ICEECS), Kuta, 2014, pp. 363-368.

[4] E. Kilic, O. Baser, M. Dolen and E. I. Konukseven, “An enhanced adaptive windowing technique for velocity and acceleration estimation using incremental position encoders,” ICSES 2010 International Conference on Signals and Electronic Circuits, Gliwice, 2010, pp. 61-64.

[5] D. S. Debnath, M. Pal, K. Banerjee, B. Dam and K. Majumdar, “An FPGA-based incremental encoder signal conditioner with reduced error in rotational rate estimation over a wide range of rotational speeds,” 2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC), Kolkata, 2016, pp. 120-124.

[6] H. Huang and W. Chou, “Hysteresis Switch Adaptive Velocity Evaluation and High-Resolution Position Subdivision Detection Based on FPGA,” in IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 12, pp. 3387-3395, Dec. 2015.

[7] A. Waheed and L. Cai, “Alternative design for optical incremental encoder measurement systems,” 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, 2016, pp. 634-639.

[8] A. Anuchin, A. Dianov, D. Shpak, V. Astakhova and K. Fedorova, “Speed estimation algorithm with specified bandwidth for incremental position encoder,” 2016 17th International Conference on Mechatronics - Mechatronika (ME), Prague, 2016, pp. 1-6.

[9] R. Petrella and M. Tursini, “An Embedded System for Position and Speed Measurement Adopting Incremental Encoders,” IEEE Trans. on Industry Applications, vol. 44, no. 5, pp. 1436-1444, Sept.-Oct. 2008.

[10] C. A. Negrea, I. I. Incze, M. Imecs, A. V. Pop and C. Szabo, “An improved speed identification method using incremental encoder in electric drives,” Proceedings of 2012 IEEE International Conf. on Automation, Quality and Testing, Robotics, Cluj-Napoca, pp. 536-540.