

# Synchronous Constant Elapsed Time Speed Estimation using Incremental Encoders

Alecksey Anuchin, *Senior Member, IEEE*, Anton Dianov *Senior Member, IEEE*,  
and Fernando Briz, *Senior Member, IEEE*

**Abstract**—This paper proposes a high-performance speed measurement method for variable speed electric drives using incremental encoders. It is based on the well-known constant elapse time (CET) method. However, the principle of speed estimation is changed from event driven to ready on demand. This helps to remove the jitter effect, caused by the asynchronous nature of the CET method, and make the speed estimation time constant regardless of the current speed. The proposed method can be implemented on modern motion control microcontrollers. Experimental results using a vector controlled Permanent Magnet Synchronous Motor (PMSM) are provided to confirm the performance of the proposed algorithm and the improvements with respect to conventional methods. Synchronous CET method of speed estimation allows to double the accuracy compared to the original CET method for the same behavior of the speed loop, or to increase the speed response time twice for the same accuracy.

**Index Terms**—angular velocity, incremental encoder, speed estimation, quadrature encoder, variable speed drives

## I. INTRODUCTION

HIGH-performance electric drives require the measurement of the rotor position both for torque control and motion control. Modest accuracy in the position feedback is enough for torque control. On the contrary, motion control normally implies very precise estimation of speed as well as position, if a position control loop is implemented. Therefore, high-performance drives must be equipped with precision speed and position sensors. So, the most popular choice in this case is an angular encoder with a speed calculation algorithm based on the differentiating of the measured position.

The drive itself consists of several elements (see Fig. 1). An inverter can be represented by a zero-order hold with a time constant equal to the duration of the pulse-width modulation (PWM) cycle. Motor has its own mechanical time constant, which is proportional to the inertia of the drive. Time constants of all other elements strongly depend on the particular implementation of a control system. As shown in Fig. 1, the delay in the current measurement is usually one half of the PWM cycle, if the current signals are processed by analog-to-digital converter (ADC) in the beginning and middle of the

PWM cycle. The current loop performance is limited by these delays in inverter and ADC, and for the most common tuning, its transfer function can be represented by a first order lag element with the time constant of the two PWM cycles.

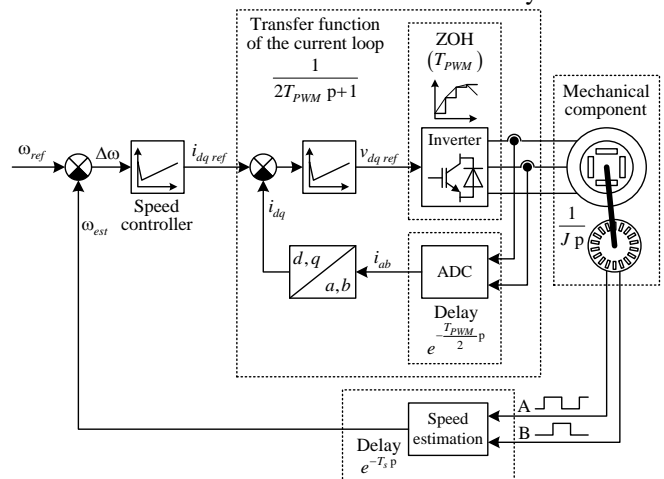


Fig. 1. Typical configuration of an electric drive with speed loop.

The same delay in the speed estimation, together with the passband of the nested current loop, limits the performance of the speed control. Thus, the delay in the speed feedback or speed estimation time should be as small as possible.

The problem of precise speed estimation was studied independently in [1] and [2], and the M/T-method and constant elapsed time (CET) method were proposed. Both methods provide the same accuracy and have common principles of operation. They were implemented by means of microprocessor systems. The achieved results solved the problem of speed measurement, but when microcontrollers broke through to the market in 1990's, those researches were forgotten for many years and mentioned only in a few papers like [3], where ASIC implementation of the same methods was suggested. The key difference in [3] is that the amount of rotation doubles with the increase in speed, whereas in CET, it changes for a minimal possible increment multiple of a single encoder pulse. During that time, many papers [4, 5] investigated period- and frequency-based methods and switching between them in order to achieve maximum accuracy using microcontroller; in [6], it

The paper was submitted for review on 25<sup>st</sup> of February 2018.

This research is supported by the Russian Science Foundation grant (Project № 16-19-10618).

A. Anuchin is with the Moscow Power Engineering Institute, Moscow, Russian Federation (e-mail: anuchin.alecksey@gmail.com).

A. Dianov is with the Samsung Electronics, Suwon, South Korea (e-mail: anton.dianov@samsung.com).

F. Briz is with the Department of Electrical, Computer, and Systems Engineering, University of Oviedo, Gijón, Spain (e-mail: fernando@isa.uniovi.es).

is suggested to implement the CET method by means of FPGA. Later in [7], the FPGA method was improved by implementing a new time measurement system with 400 ps resolution, and in [8], the CET method was implemented using modern microcontroller.

In [9], it was suggested to implement an average filter to process the results of the frequency-based method. This solution is comparable with the CET or M/T methods in its accuracy but brings huge CPU burden on high speeds when the input frequency from the encoder increases. This method can be implemented using FPGAs, but not a microcontroller.

Summing up the above, the improvement of the speed loop response time is possible if the sampling time  $T_s$  is reduced. In [2], the sampling time is suggested to be half of the electrical time constant of the current loop; in this case the jitter effect of the asynchronous speed estimation becomes neglectable. But the smaller sampling time decreases the accuracy of the speed estimation because its error is inverse of the number of the digital timer ticks spent on the speed estimation. Better accuracy can be achieved by means of FPGA with ps resolution of the digital timer clock, but this solution is more complex and expensive in comparison to the implementation of the whole control system using a single microcontroller.

Resolution of speed measurements using CET or M/T methods is proportional to the sampling time. The main drawback is that the speed estimations are driven by encoder pulses. This results in unpredictable delays from  $T_s/2$  to  $3T_s/2$ , affecting to speed controller performance. In this paper a novel method of speed measurement is proposed, in which speed estimates are synchronized with the speed controller. The proposed method implements a synchronous CET, bounding the measurement delay to be approximately  $T_s/2$ . This enables an increase of either the speed loop bandwidth or the accuracy of the speed estimation. An increase of the accuracy of the speed estimation would result in less noise in the torque reference, and in the end in a reduction in the THD of the currents. If the sampling time for the speed control loop is decreased, but with the same accuracy in the estimated speed and same gains in the speed controller, the dynamics response of the speed control should be improved in general.

## II. INCREMENTAL ENCODERS AND INTERFACE WITH MICROCONTROLLERS

### A. Interface Between the Incremental Encoder and Microcontroller Peripherals

The incremental encoder produces a sequence of two signals shifted by  $90^\circ$ , where the sign of the shift depends on the direction of the rotation. In a simple case, these signals can be produced by a system of light-emitting diode, a metal or glass disc with slits, and two photodiodes. The light passes through the slits onto the photodiodes, which are phase shifted at  $90^\circ$  corresponding to a single slit. The signals from these two photodiodes—"A" and "B"—produce a sequence of pulses as the encoder rotates. The main parameter of the incremental encoder is the number of pulses per revolution (ppr) given by each channel— $K$ . The direction of rotation is obtained from the

phase shift between these two signals.

These two signals come to the inputs of the microcontroller, which are internally connected with a special peripheral device called Quadrature Encoder Pulse Module (QEP), Encoder Counter or similar, depending on the manufacturer of the chip. The signals "A" and "B" are connected to the QEP decoder logic, which detects the rising and falling edges of the signals. Following each transition of any of these signals, the QEP logic provides the signal "QDIR" that indicates the direction of rotation and the clock signal for position counter increment/decrement "QCLK", Fig. 2. These two signals increment or decrement counter in the  $QPOSCNT$  register, which stores current rotor position as shown in Fig. 2.

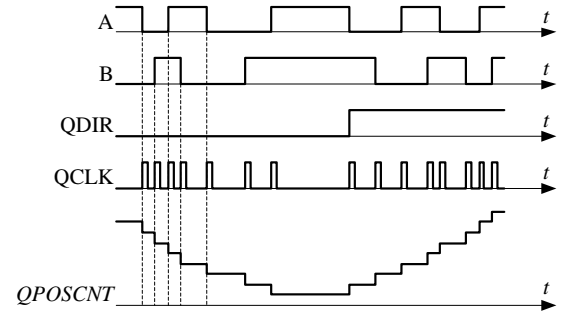


Fig. 2. From top to bottom: A-B encoder signals; direction of rotation detected from the encoder signals; QCLK signal; and position counter ( $QPOSCNT$ ), during a speed reversal.

The position counter counts every edge of the signals "A" and "B"; therefore, each QCLK pulse corresponds to  $360/N$  mechanical degrees. So, the resolution of the encoder is as follows:

$$N = 4K. \quad (1)$$

### B. Incremental Encoder Non-idealities

Encoders suffer from two main non-ideal behaviors that can have a non-negligible impact on the accuracy of the estimated speed. The first one occurs when a signal line comparator inside the encoder has an incorrect reference value, and the duty cycles of the channels are not equal to 50%. This is shown in Fig. 3a. The second occurs when the photodiodes of the "A" and "B" channels are not properly placed, so the phase shift between both the channels is not equal to  $90^\circ$ , as shown in Fig. 3b, and this error can be as high as  $10^\circ$  in some encoders [10]. Finally, differences among consecutive teeth should be expected, the error is usually  $<1^\circ$ .

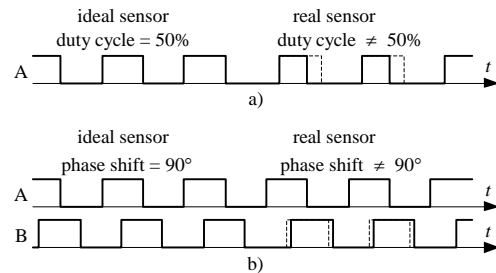


Fig. 3. Incremental encoder non-idealities.

### C. Time Measurement

Microcontrollers have timers that can be used to measure time. Typically, the minimal measurable quantity of time is equal to one CPU cycle. For example, if a microcontroller has a CPU frequency of 60 MHz, then the resolution of time measurement— $h$ —is its inverse value, which is equal to 16.67 ns.

## III. OVERVIEW OF THE SPEED ESTIMATION METHODS

### A. Frequency-based Speed Estimation

The frequency-based method [11] counts pulses from the sensor during a given sampling time  $T_s$ . The speed  $n$  in rpm can be estimated by the following:

$$n_f = \frac{60}{T_s} \cdot \frac{\Delta c}{N}, \quad (2)$$

where  $\Delta c$  is the amount of rotation represented by the number of QCLK pulses obtained from the position encoder during sampling time  $T_s$ . Though simple, this method has poor accuracy defined as:

$$\Delta n_f = \frac{60}{T_s} \cdot \frac{1}{N} = \frac{60 f_s}{N}. \quad (3)$$

For instance, for an encoder with  $N = 4000$  and sampling frequency of 2 kHz for the speed control, the error is as follows:

$$\Delta n_f = \frac{60 \cdot 2000}{4000} = 30 \text{ rpm}. \quad (4)$$

### B. Period-based Speed Estimation

The period-based method measures the time needed to rotate a certain angle. The angle must be an integer number  $\Delta c$  of QCLK pulses—amount of rotation. It is noted that the values of  $\Delta c$  being integer multiples of four is advisable [5], as four corresponds to a single period of the encoder signals A or B, so most of the inaccuracies of the incremental encoders do not impact the result. It is also noted that though the intermediate edges are ignored by the speed estimator, they still could be used for position estimation.

For the amount of rotation  $\Delta c = 4$ , the speed in rpm  $n$  can be obtained as follows:

$$n_p = \frac{60}{T_s} \cdot \frac{4}{N}, \quad (5)$$

where  $T_s$  is the time elapsed to rotate the specified angle. As the speed increases,  $T_s$  decreases, as shown in Fig. 4. This value depends on the current speed and amount of rotation and can be expressed from (5) as follows:

$$T_s = \frac{60 \cdot 4}{n \cdot N}. \quad (6)$$

Obviously, the higher the CPU clock frequency  $f_{CPU}$  used to measure time and the smaller the minimal quantity of time  $h$ , the higher the accuracy of  $\Delta t$  measurement. The absolute error for the estimated speed is given as follows:

$$\Delta n_p = \frac{h}{T_s} n = \frac{h \cdot n^2 \cdot N}{60 \cdot 4}. \quad (7)$$

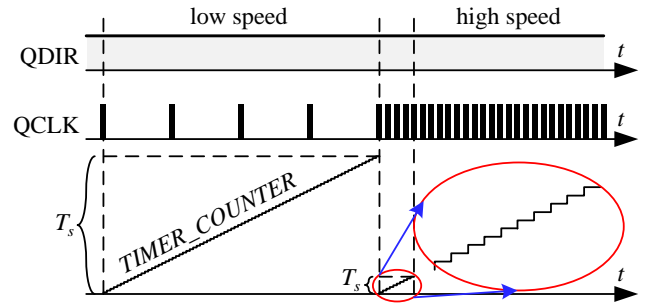


Fig. 4. Period-based method diagram. From top to bottom: direction of rotation; QCLK signal; and period timer counter.

As an example, for a spindle drive rotating at 30 000 rpm, the 1000 ppr encoder output sampling frequency reaches 2 MHz. If the position increment is set to four counts, then the elapsed time is 2  $\mu$ s. If the CPU operates at 60 MHz, the absolute error is:

$$\Delta n_p = \frac{1}{60\,000\,000} \cdot 30\,000^2 \cdot 4000}{60 \cdot 4} = 250 \text{ rpm}. \quad (8)$$

### C. Combined Method and its Accuracy

To reduce the speed estimation error of the period-based method at high speeds, many authors suggest switching to the frequency-based method and counting the number of encoder pulses during specified period of time [5, 4]. The optimal switching point that crosses the error curves can be derived from (3) and (7) as follows:

$$\frac{60}{T_s \cdot n \cdot N} = \frac{h \cdot n \cdot N}{60 \cdot 4}, \quad (9)$$

which after the expression of speed gives an optimal switching speed:

$$n_{opt} = \frac{2 \cdot 60}{N \sqrt{h \cdot T_s}}. \quad (10)$$

### D. Constant Elapsed Time Method

The CET method [1] or M/T method [2] uses a similar principle as period-based method. It operates with some initial amount of rotation for the single speed estimation. As the shaft of the encoder rotates that specified angle, the elapsed time is compared to the referenced value. That value corresponds to the desired sampling frequency of the speed estimator. If the elapsed time becomes smaller than the referenced value, the amount of rotation is incremented by 4 QCLK encoder pulses. Thus, with the increase in the speed, the amount of rotation increases to keep the elapsed time approximately constant.

Contrary, when the speed decreases, the elapsed time increases. Now it is compared to the reference, and when it becomes bigger than the referenced value, the amount of rotation is decreased by 4 QCLK encoder pulses. The minimal amount of rotation is limited by 4 pulses in order to avoid the impact of encoder inaccuracies [5], as shown in Fig. 3. The elapsed time with respect to speed for the CET method is

displayed in Fig. 5. The minimal measured speed is detected with the help of a watch-dog software timer implementation that detects the stop and speed reversal of the drive.

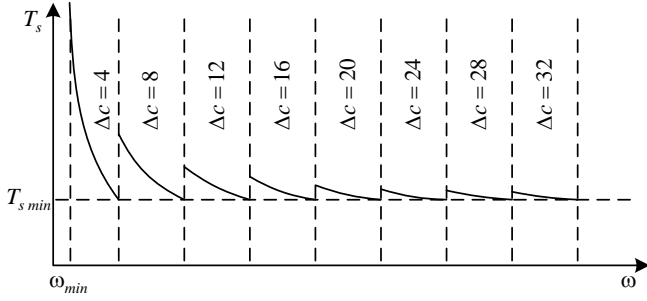


Fig. 5. Elapsed time with respect to speed for CET method.

The current speed can be estimated using (2). As the elapsed time is kept near the specified value, and the amount of rotation is represented with the exact number of pulses, the absolute error for that method does not exceed the following:

$$\Delta n_{max} = \frac{h}{T_{s\ min}} \cdot n. \quad (11)$$

#### E. CET method with scalable amount of rotation

For the simplification of computations, the CET method was modified to use the amount of rotation represented in QCLK pulse from the set  $\Delta c = \{4, 8, 16, 32, 64, \dots\}$  [3] and [6]. Every time the speed doubles, the amount of rotation doubles too in order to keep the elapse time in the specified range. The accuracy of this method is comparable with CET, but the sampling time varies as shown in Fig. 6; thus, the delay and jitter effect have higher impact on the speed loop performance.

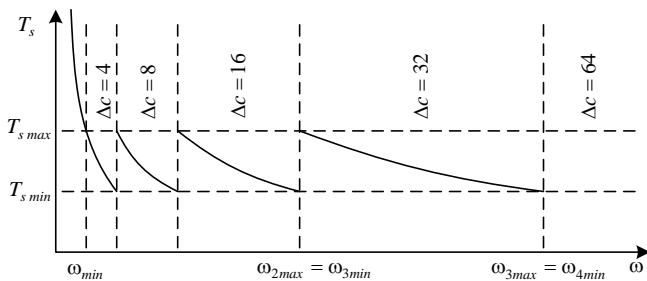


Fig. 6. Elapsed time with respect to speed for CET method with scalable amount of rotation.

#### F. Period-based Method with Averaging

This method computes the average value of all estimations  $S$  of the period-based method occurred during sampling time as follows:

$$n_{av} = \frac{1}{S} \sum_{i=1}^S \left( \frac{60}{T_s[i]} \cdot \frac{4}{N} \right) = \frac{4 \cdot 60}{S \cdot N \cdot \sum_{i=1}^S T_s[i]}. \quad (12)$$

The accuracy of the method is the same as CET because time estimation is oversampled  $S$  times, and the total maximum error in time measurement is equal to the minimal quantity of time  $h$ , whereas the total sampling time is the same as for the CET

method.

#### IV. SYNCHRONOUS CONSTANT ELAPSED TIME METHOD

The main disadvantage of the most accurate method—the CET method—is that it operates asynchronously with speed control loop execution because it is event driven by the arrival of encoder pulses. The estimation happens when a particular encoder pulse occurs. This event can be predicted and scheduled by selecting a certain amount of rotation, but in case of speed deviation due to the change of load or torque of the drive, it may occur earlier or too late. Thus, it is necessary to modify the speed estimation algorithm from event driven to ready on demand.

Incremental encoders provide microcontrollers two quadrature signals which are decoded by the QEP logic of internal peripheral device into position. Usually, this module is able to measure the interval between the input pulses, but it is not able to measure the instants of time when the certain pulse occurs. For measurement of the instants of time, the encoder signals are to be connected to the time capture units of the same microcontroller as shown in Fig. 7.

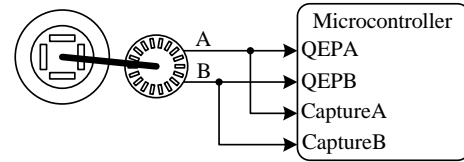


Fig. 7. Connection of incremental encoder to microcontroller for implementation of synchronous CET method.

The speed estimation should be performed immediately before the execution of the speed controller, to introduce the smallest possible delay in the speed loop. The actual position  $c$  from encoder pulses is available as well as the direction of rotation. The capture units provide the instants of time of the last rising and falling edges of the signals “A” and “B”. The current state of the signals can be obtained from a digital input/output module. Thus, at the moment, when the speed should be estimated, the following data are available:

- actual position from QCLK encoder pulses  $c[k]$ ;
- direction of rotation;
- two sequential instants of time when the last rising  $t_{AR}[k]$ ,  $t_{BR}[k]$ , and falling  $t_{AF}[k]$ ,  $t_{BF}[k]$  edges occurred;
- current state of the encoder signals.

Consider the first case from Fig. 8a. The amount of rotation between two consecutive estimations is a multiple of 4. The motor rotates in the positive direction, and the current states of the encoder signals “A” and “B” are 1 and 0, respectively. Thus, the most recent elapsed time can be evaluated as a difference between the instants of time when the current and last rising edges of the A signal occurred. The equations for position increment and elapsed time in this particular case are given below:

$$\left. \begin{aligned} \Delta c &= c[k] - c[k-1], \\ T_s &= t_{AR}[k] - t_{AR}[k-1]. \end{aligned} \right\} \quad (13)$$

The second case from Fig. 8b represents a situation when the last and current states of the encoder are different. The amount of rotation is not a multiple of 4, and the encoder inaccuracies may impact the measured speed; therefore, the position increment as well as elapsed time should be corrected before obtaining the speed using (2).

The nearest amount of rotation that is a multiple of 4 is one encoder pulse bigger than the difference between the current and the previous position counter values. That amount of rotation corresponds to the time difference between the current and previous rising edges of A signal as shown in Fig. 8b. Therefore, the amount of rotation and elapsed time in this particular case can be expressed as follows:

$$\left. \begin{aligned} \Delta c &= c[k] - c[k-1] + 1, \\ T_s &= t_{AR}[k] - t_{AR}[k-1], \end{aligned} \right\} \quad (14)$$

whereas the speed equation (2) remains the same.

The same rules for each combination of the current and previous encoder states were derived and presented in Table I. Two different approaches can be used at low speeds when the amount of rotation is less than 4: using the amounts of rotation smaller than 4 or increasing elapsed time.

The first approach, that uses amounts of rotation smaller than 4, is sensitive to encoder non-idealities. But this is the only way to keep the sampling time for the speed feedback constant. Actually, low speeds naturally decrease in accuracy due to the higher impact of speed deviation while a single estimation occurs. Therefore, it is better to have some actual information with some error, instead of the accurate result in average, while the drive has already stopped or even changed the direction of rotation. The rules for this approach for the amount of rotation of 3, 2, and 1 encoder pulses are included in Table I. An example for the amount of rotation of 2 QCLK encoder pulses is presented in Fig. 9. The current speed estimation was performed after the rising edge of signal “A” in two QCLK pulses from last speed estimation. Therefore, the sampling time should be evaluated as a difference between the instant of time of the last rising edge and the last falling edge of signal “A”:

$$\left. \begin{aligned} \Delta c &= c[k] - c[k-1], \\ T_s &= t_{AR}[k] - t_{AF}[k-1]. \end{aligned} \right\} \quad (15)$$

The second approach will result in error if fast changes of speed occur. In case of implementation of this algorithm, the speed estimator operates only with the “>3” rule from Table I regardless of the recent amount of rotation:

$$\left. \begin{aligned} \Delta c &= c[k] - c[k-1] + 2, \\ T_s &= t_{AR}[k] - t_{AR}[k-1]. \end{aligned} \right\} \quad (16)$$

For zero angular increment, the zero speed should be detected. The zero-speed detection do not require any timeout counter as for conventional CET or M/T methods.

TABLE I  
EQUATIONS FOR EVALUATION OF AMOUNT OF ROTATION AND ELAPSED TIME  
(FOR THE POSITIVE DIRECTION)

$c[k] - c[k-1]$	States		Amount of rotation	Sample time $T_s$	
	New A	Old B			
>3	1 0	1 0	$c[k] - c[k-1]$	$t_{AR}[k] - t_{AR}[k-1]$	
	1 0	1 1	$c[k] - c[k-1] + 1$	$t_{AR}[k] - t_{AR}[k-1]$	
	1 0	0 1	$c[k] - c[k-1] + 2$	$t_{AR}[k] - t_{AR}[k-1]$	
	1 0	0 0	$c[k] - c[k-1] + 3$	$t_{AR}[k] - t_{AR}[k-1]$	
	1 1	1 0	$c[k] - c[k-1] + 3$	$t_{BR}[k] - t_{BR}[k-1]$	
	1 1	1 1	$c[k] - c[k-1]$	$t_{BR}[k] - t_{BR}[k-1]$	
	1 1	0 1	$c[k] - c[k-1] + 1$	$t_{BR}[k] - t_{BR}[k-1]$	
	1 1	0 0	$c[k] - c[k-1] + 2$	$t_{BR}[k] - t_{BR}[k-1]$	
	0 1	1 0	$c[k] - c[k-1] + 2$	$t_{AF}[k] - t_{AF}[k-1]$	
	0 1	1 1	$c[k] - c[k-1] + 3$	$t_{AF}[k] - t_{AF}[k-1]$	
	0 1	0 1	$c[k] - c[k-1]$	$t_{AF}[k] - t_{AF}[k-1]$	
	0 1	0 0	$c[k] - c[k-1] + 1$	$t_{AF}[k] - t_{AF}[k-1]$	
	0 0	1 0	$c[k] - c[k-1] + 1$	$t_{BF}[k] - t_{BF}[k-1]$	
	0 0	1 1	$c[k] - c[k-1] + 2$	$t_{BF}[k] - t_{BF}[k-1]$	
	0 0	0 1	$c[k] - c[k-1] + 3$	$t_{BF}[k] - t_{BF}[k-1]$	
	=3	0 0	0 0	$c[k] - c[k-1]$	$t_{BF}[k] - t_{BF}[k-1]$
1 0		1 1	3	$t_{AR}[k] - t_{BR}[k-1]$	
1 1		0 1	3	$t_{BR}[k] - t_{AF}[k-1]$	
0 1		0 0	3	$t_{AF}[k] - t_{BF}[k-1]$	
0 0		1 0	3	$t_{BF}[k] - t_{AR}[k-1]$	
=2		1 0	0 1	2	$t_{AR}[k] - t_{AF}[k-1]$
		1 1	0 0	2	$t_{BR}[k] - t_{BF}[k-1]$
		0 1	1 0	2	$t_{AF}[k] - t_{AR}[k-1]$
		0 0	1 1	2	$t_{BF}[k] - t_{BR}[k-1]$
=1		1 0	0 0	1	$t_{AR}[k] - t_{BF}[k-1]$
		1 1	1 0	1	$t_{BR}[k] - t_{AR}[k-1]$
		0 1	1 1	1	$t_{AF}[k] - t_{BR}[k-1]$
		0 0	0 1	1	$t_{BF}[k] - t_{AF}[k-1]$
=0		x x	x x	0	$\infty$

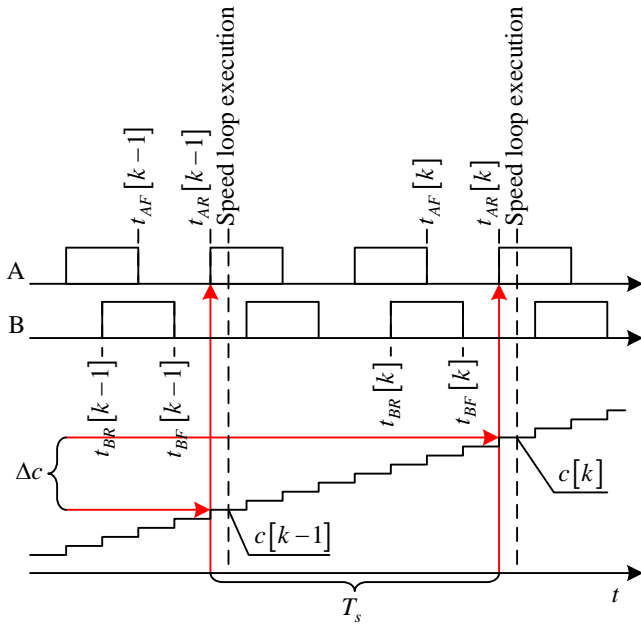
## V. EXPERIMENTAL RESULTS

### A. Experimental Setup

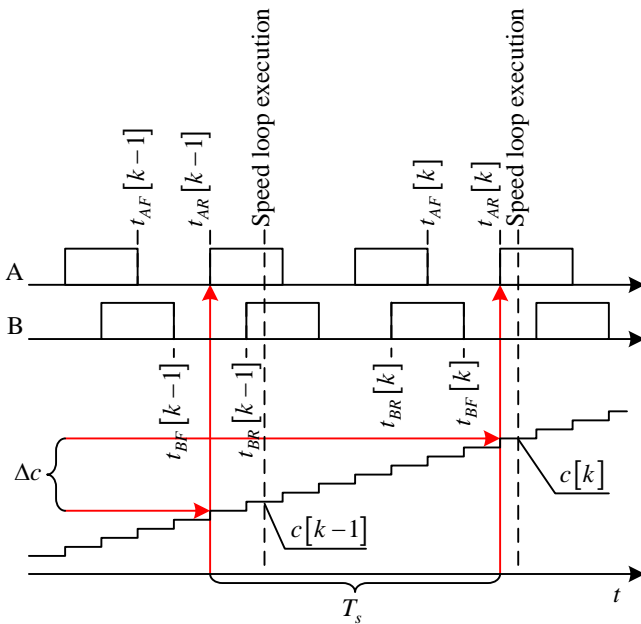
The experimental setup includes a Texas Instruments inverter board DRV8302-HC-C2-KIT fed from 28 V DC power supply. The controller PCB, based on Piccolo microcontroller unit TMS320F28035, operates at 60 MHz core frequency. This microcontroller has a single event capture unit; therefore, the speed estimation algorithm was adapted to process only signal “A” of the encoder. The inverter feeds a permanent magnet synchronous motor, which has 2 pole pairs, 6 A of rated RMS current, and 5000 rpm of maximum speed. The motor is coupled with incremental encoder HEDM-5500#B06 with 1000 ppr. The experimental setup is shown in Fig. 10.

The control system is written in C using Code Composer Studio IDE and implements field-oriented control with the current loop execution frequency of 10 kHz. The control system has current controllers of a proportional-integral type for D- and Q-axes. The speed loop contains proportional controller with the gain of 0.08 A/rpm. The current reference is limited by 10 amperes. The execution frequency of the speed loop was set to 2 kHz. This helps highlight the difference between the original CET and proposed synchronous CET methods. The experimental data were obtained by means of digital software oscilloscope, which is based on “DLOG” module from Control Suite library by Texas Instruments.

The implementation of the synchronous CET method using Assembly language needs approximately 3  $\mu$ s for 60 MHz microcontroller and 13  $\mu$ s for C language implementation.



a)



b)

Fig. 8. Explanation on evaluation of amount of rotation and elapsed time.

**B. Motor Start up**

Experiments were conducted with different tunings of the CET method and synchronous CET presented in Table II in order to show the differences in the response time. The accuracy is computed according to the microcontroller frequency with the assumption that the incremental encoder has ideal teeth shapes. In practice the difference in the consecutive teeth shapes results in error in the amount of rotation. This problem could be solved performing a calibration during commissioning of the electric drive.

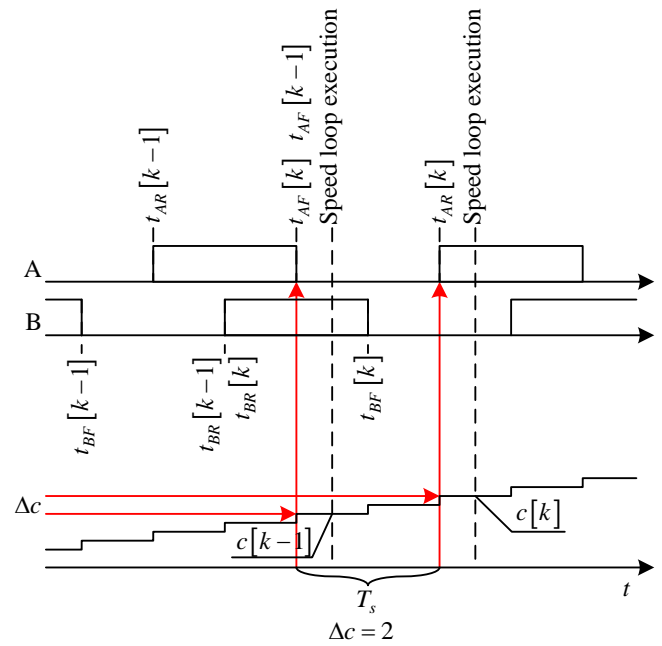


Fig. 9. Explanation for the amount of rotation of 2 QCLK encoder pulses.



Fig. 10. Motor kit with the PMSM and 1000 ppr incremental encoder.

TABLE II  
PROPERTIES OF THE SPEED MEASUREMENT METHODS DURING EXPERIMENTS

#	Method	Maximum error at settled speed, rpm	Speed estimation delay, $\mu$ s
1	CET	0.07	from 250 to 750
2	CET	0.13	from 125 to 375
3	Synchronous CET	0.07	$\approx$ 250

The transient of the motor start up with the conventional CET method is shown in Fig. 11. The speed reference changes from zero to 2000 rpm. As a result, the speed controller saturates and applies 10 A reference to the Q-axis current loop. The motor

starts to rotate with approximately constant acceleration. At the instance of a time of 4 ms, the first speed estimation occurs. The execution of the speed controller occurs every 500  $\mu\text{s}$ , which causes the Q-current reference update. With increase in the speed, the elapsed time remains approximately on the same level of 30000 CPU cycles or 500  $\mu\text{s}$ . To keep it there, the speed estimation algorithm changes the amount of rotation dynamically, but the speed estimation is not synchronized with the speed controller execution; it can occur at any time in between. Thus, the speed controller uses the result with variable delay in the feedback. As the drive reaches the referenced speed, the oscillations in the speed and Q-axis current transients can be seen. They are caused by a too high gain of the speed controller for this variable delay in the feedback.

During the next experiment presented in Fig. 12, the speed elapse time was set to twice the smaller level of 15000 CPU cycles or 250  $\mu\text{s}$ . That reduces the accuracy by half in the speed estimation in comparison to the previous experiment. The speed transient in this case has acceptable oscillation that fades very quickly. The decrease in the delay in the feedback made the speed loop more stable.

The proposed synchronous CET method (see Fig. 13) has the same behavior at the beginning of the transient. It also keeps the elapsed time near 500  $\mu\text{s}$ , but now the estimation is strictly synchronized with the speed loop execution. The estimation is synchronized, and the speed value is updated simultaneously with the Q-current reference. Due to the elimination of the variable delay in the feedback, the drive reaches the referenced speed with the same level of oscillations as for the conventional CET method with the poorer accuracy.

The conducted experiments shown that it is possible to obtain twice higher proportional gain of the speed controller than for conventional CET method with the same accuracy. The q-axis current reference is proportional to the difference between referenced and actual speeds multiplied by proportional gain of the speed controller  $K_{SC}$ :

$$i_{q\text{ref}} = K_{SC} (n_{\text{ref}} - n), \quad (17)$$

and assuming that the torque and current commands are equal to their actual values in a steady state, the torque equation can be written as:

$$T = K_T i_q = K_T K_{SC} (n_{\text{ref}} - n), \quad (18)$$

where  $K_T$  is the torque coefficient. After simplification it gives the stiffness of the speed-torque curve

$$\beta = \frac{\Delta T}{\Delta n} = K_T K_{SC}, \quad (19)$$

which can be twice higher for synchronous CET method.

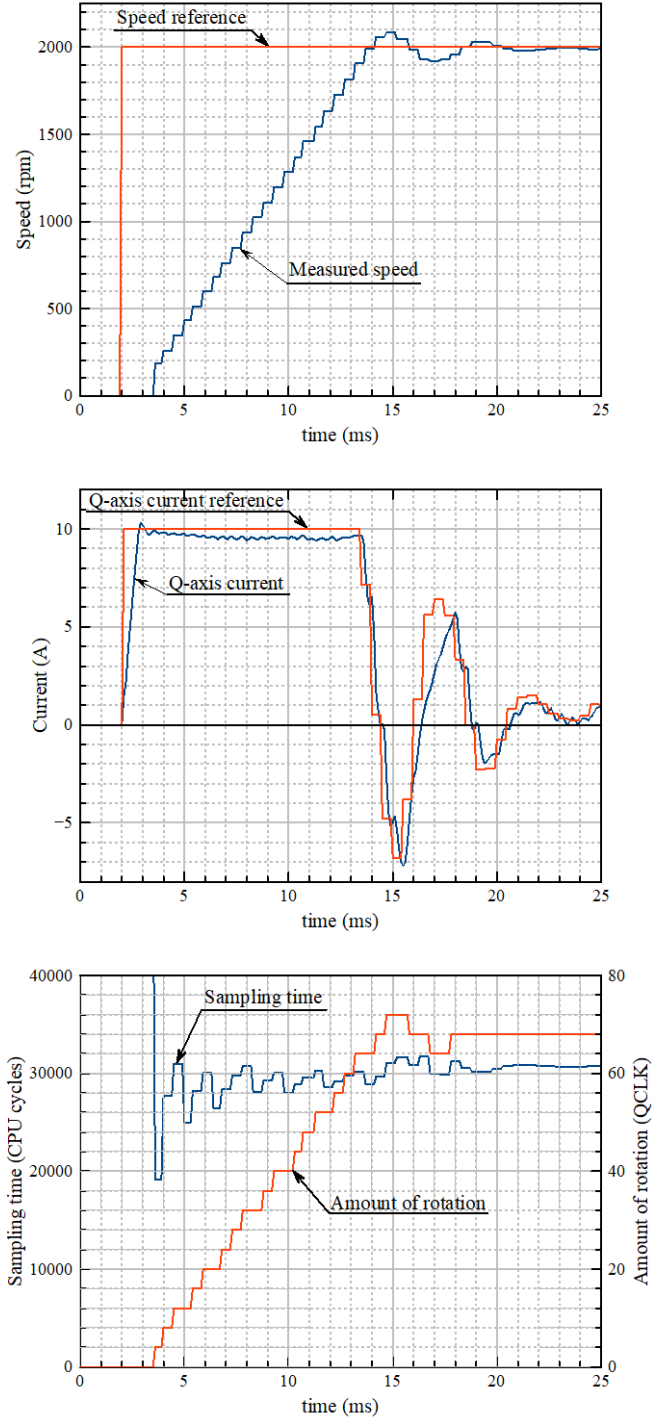


Fig. 11. Motor start up with CET speed estimation method and 500  $\mu\text{s}$  sampling time.



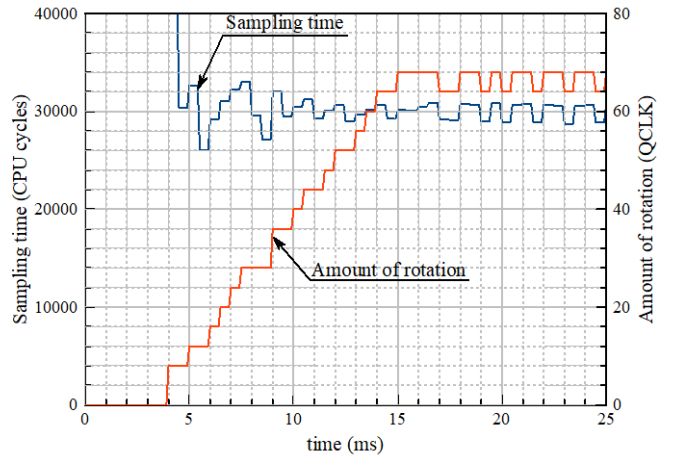
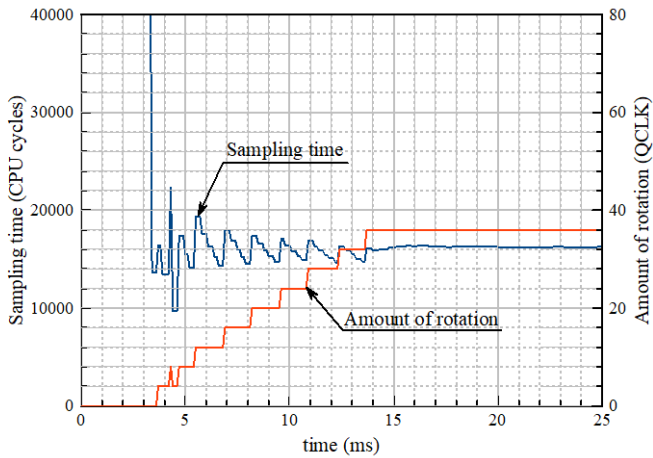
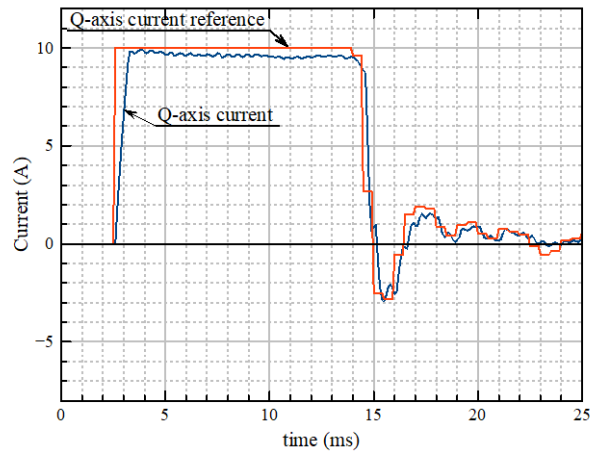
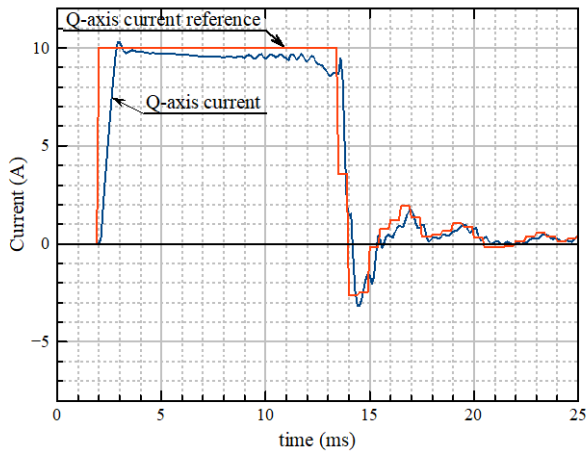
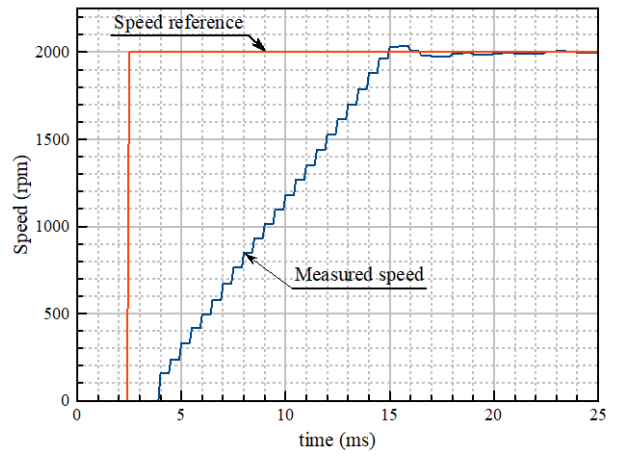
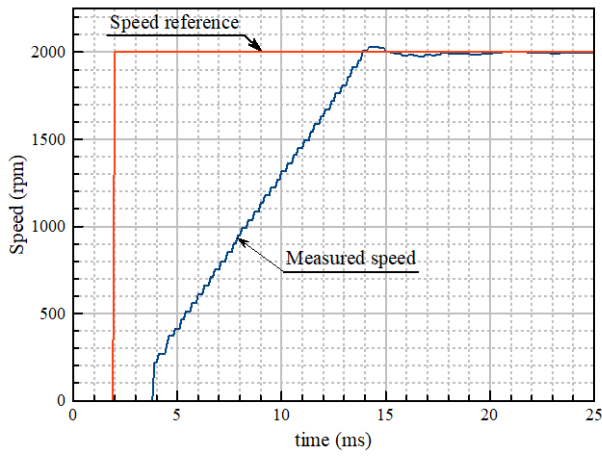


Fig. 12. Motor start up with CET speed estimation method and 250  $\mu$ s sampling time.

Fig. 13. Motor start up with Synchronous CET speed estimation method and 500  $\mu$ s sampling time.

## VI. CONCLUSIONS

This paper proposes a new speed estimation method that is synchronized with speed controller execution. The method is based on the conventional CET method but is improved and adapted for implementation on modern microcontrollers. The main advantage of the proposed synchronous CET method is the absence of jitter in the speed feedback that allows to use more aggressive tuning of the speed controller and achieve



higher stiffness of the closed-loop speed-torque curve of the drive.

The performance of the proposed method has been confirmed experimentally on a Texas Instruments microcontroller-based drive. Its implementation is easy, as it exploits the microcontroller peripherals without any additional hardware requirements. It should be mentioned that the implementation does not require any interrupt routines and is done only by the polling of certain registers of peripheral devices before the execution of the speed controller. The estimation algorithm needs approximately 800 CPU cycles for implementation using C language and 200 CPU cycles for Assembly.

#### REFERENCES

- [1] T. Ohmae, T. Matsuda, K. Kamiyama, M. Tachikawa, "A Microprocessor-Controlled High-Accuracy Wide-Range Speed Regulator for Motor Drives," *IEEE Trans. on Industrial Electronics*, vol. IE-29, issue 3, pp. 207–211, Aug. 1982, 10.1109/TIE.1982.356665.
- [2] R. Bonert, "Digital Tachometer with Fast Dynamic Response Implemented by a Microprocessor," *IEEE Trans. on Industry Applications*, vol. IA-19, issue 6, pp. 1052–1056, Nov. 1983, 10.1109/TIA.1983.4504334.
- [3] E. Galvan, A. Torralba, L. G. Franquelo, "ASIC implementation of a digital tachometer with high precision in a wide speed range," *IEEE Trans. on Industrial Electronics*, vol. 43, issue 6, pp. 655–660, Dec. 1996, 10.1109/41.544553.
- [4] A. C. Negrea, M. Imecs, I. I. Incze, A. Pop, C. Szabo, "Error compensation methods in speed identification using incremental encoder," in *Proc. EPE*, Isai, Romania, 2012, pp. 441–445, 10.1109/ICEPE.2012.646385.
- [5] R. Petrella, M. Tursini, L. Peretti, M. Zigliotto, "Speed measurement algorithms for low-resolution incremental encoder equipped drives: a comparative analysis," in *Proc. ACEMP'07*, Bodrum, Turkey, 2007, pp. 780–787, 10.1109/ACEMP.2007.4510607.
- [6] J. N. Lygouras, K. A. Lalakos, P. G. Ysalides, "High-performance position detection and velocity adaptive measurement for closed-loop position control," *IEEE Trans. on Instrumentation and Measurement*, vol. 47, issue 4, pp. 978–985, Aug. 1998, 10.1109/19.744653.
- [7] J. N. Lygouras, T. P. Pachidis, K. N. Tarchanidis, V. S. Kodogiannis, "IEEE Trans. on Instrumentation and Measurement", vol. 57, issue 9, pp. 2035–2043, Mar. 2008, 10.1109/TIM.2008.919039.
- [8] F. Brugnano, C. Concarì, E. Imamovic, F. Savi, A. Toscani, R. Zanichelli, "A simple and accurate algorithm for speed measurement in electric drives using incremental encoder," in *Proc. IECON2017*, Beijing, China, 2017, pp. 8551–8556, 10.1109/IECON.2017.8217502.
- [9] A. F. Ilmiawan, D. Wijanarko, A. H. Arofah, H. Hindersyah, A. Purwadi, "An easy speed measurement for incremental rotary encoder using multi stage moving average method," in



**Fernando Briz** (A'96–M'99–SM'06) received the M.S. and Ph.D. degrees from the University of Oviedo, Gijon, Spain, in 1990 and 1996, respectively.

He is currently a Full Professor with the Department of Electrical, Computer and Systems Engineering, University of Oviedo. His research interests include electronic power converters and ac drives, power systems, machine monitoring and diagnostics and digital signal processing.

Dr. Briz received an IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS Award and nine IEEE Industry

*Proc. ICEECS*, Kuta, Indonesia, 2015, pp. 363–368, 10.1109/ICEECS.2014.7045279.

- [10] HEDS-9000/9100, Two Channel Optical Incremental Encoder Modules, Data Sheet, Avago Technologies.
- [11] F. Briz, J. A. Cancelas, A. Diez, "Speed measurement using rotary encoders for high performance AC drives," in *Proc. IECON94*, Bologna, Italy, 1994, pp. 538–542, 10.1109/IECON.1994.397844.



**Alecksey Anuchin** (M'13–SM'19) received his B.Sc., M.Sc., Ph.D., and Dr. Eng. Sc. degrees from Moscow Power Engineering Institute in 1999, 2001, 2004, and 2018, respectively.

He has more than 20 years of experience covering control systems of electric drives, hybrid powertrains, and real-time communications. He is the author of three textbooks on the design of real-time software for the microcontroller of the C28 family and Cortex-M4F, and control system of electric drives (in Russian). He has published more than 100 conference and journal papers. He delivers lectures on "Control Systems of Electric Drives", "Real-time Software Design", "Electric Drives," and "Science Research Writing" in Moscow Power Engineering Institute. He is in a head position at the Electric Drives Department for the last 8 years.



**Anton Dianov** (M'06–SM'18) received his B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Moscow Power Engineering Institute in 2000, 2002, and 2005, respectively.

Since 2005 he has been a Senior Engineer with Samsung Electronics, where he develops motor drives for home appliances. He is the author of more than 30 journal and conference papers on electrical drives and motor control. He is the author of several patents on control algorithms for electrical drives and power electronics. He is a member of editorial board of several journals on power electronics and electrical drives including International Journal of Power Electronics (IJPELEC) and Journal of Power Electronics (JPE). His research interests are sensorless electrical drives, electrical drives and advanced control algorithms.

Applications Society Conference, and IEEE Energy Conversion Congress and Exposition prize paper awards. He is currently the Chair of the Industrial Drives Committee of the IAS-IPCS.