Digital Sensorless Speed Direct Current Motor Control By the Aid of Static Speed Estimator

Constantin Pavlitov[1], Yassen Gorbounov[2], Radoslav Rusinov[2], Anton Dimitrov

[1] C. Pavlitov is with the Automation of Electrical Drives Department, Technical University of Sofia, Bulgaria.
[2] Y.Gorbounov, R. Rusinov are with the Department of Automation of Mining Productiom, University of Minig and Geology "St. Ivan Rilski", Sofia, Bulgaria

Abstract— this paper deals with speed control of DCM without angular speed measuring device. This device sometimes appears to be rather expensive and unaffordable in everyday common speed controllers. However since speed feedback has to exist - measuring of anchor current and mathematical speed estimator device is suggested [1] instead speedometer. The synthesis of sensorless speed controller is performed in three major steps. The first one is referred to identification of the motor; the second one reflects Matlab Simulink model of the whole system – motor and regulator and the third step reveals microcontroller implementation. The implementation has been made by the aid of MSP430 16-bit MCU and most of the functions have been performed by programmable means. This article will be quite helpful for those ones who are eager to make their transition from Matlab simulation steps towards digital controller system implementations.

Index Terms— sensorless control, dc motor control, speed estimator, speed control.

I INTRODUCTION

This article does not pretend for newly scientific principles of control. Its purpose is to demonstrate a teaching example for the transition between the mathematical models of electrical drive into its practical implementation. That is why the control programe has been discussed in details and model pecuarilities have been given as crutial point guidelines. The example is dedicated to sensorless speed control of direct current motor. We hope that extensive authors' experience of the area will be of benefit to the young digital electrical drive designers in their process of learning.

II MATHEMATICAL MODEL

The mathematical model consists of several consecutive models: Motor Identification; Regulator Device Model; Speed Estimator and Angular Speed Filter Device; Mathematical Model of the Whole System.

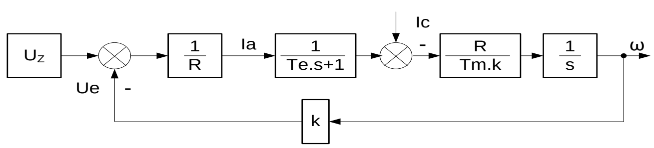

Figure 1 Direct Current Motor Matlab Simulink Model

A Motor Identification

Matlab Simulink model [2], [3] is presented in Fig.1.

In the figure: U_z is the power bridge supply voltage; $U=\gamma U_z$ is the anchor voltage which maximal value is 12[V] and duty cycle is γ; I_a is the anchor current with a maximal value of 1.1[A]; I_c is the static anchor current that is proportional to the value of the static torque; R is the active anchor resistance (it is measured within a stuck anchor: $R=U_z/I_a=11[\Omega]$); T_E is the electrical time constant of the motor (it is measured by an L-meter many times and the average value is taken into account: $T_E=2.94\bullet10^{-4}[s]$; ω is the angular speed [rad/s]; k is the motor constant ($k=(U_z-R.I_a)/\omega=0.02$); T_m is the mechanical time constant which is

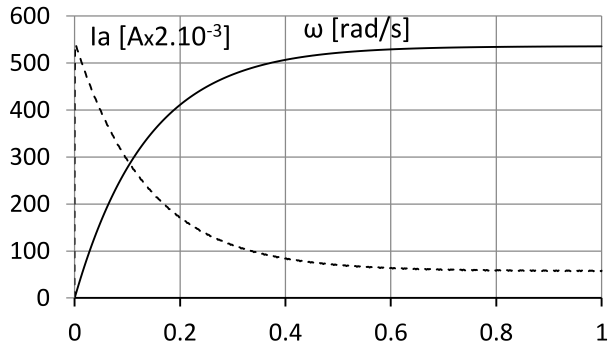

Figure 2 Starting Motor transition process. U_Z=12V, I_C=116mA; W_{MAX}=534rad/s.

measured with the time for self stopping of the motor and here it is 138ms; *J* is the moment of inertia – (1):

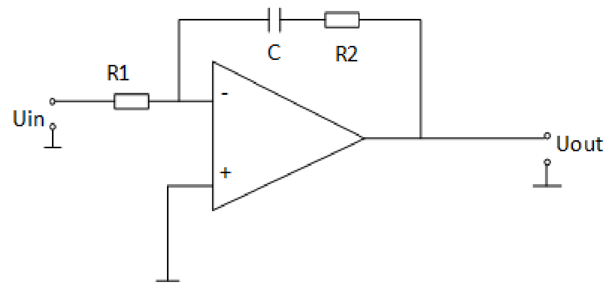

Figure 3 Analogue variant of the PI regulator

$$J = \frac{k^2.T_m}{R} = \frac{0.02^2 \cdot 0.1375}{11} = 5.103510^{-6}\left[kg.m^2\right] \quad (1)$$

The starting transition process of the motor is given in Fig.2.

B Regulator device

The regulator device in classical electrical drives [2], [4], [5] of the kind is given in Fig.3.

In fact it is analogue PI regulator with two time constants T1=R1.C and T2=R2.C. U_{in} and U_{out} are correspondingly the input and output of this device. The transient function is easy to be obtained by this circuit diagram. The corresponding equation is (2):

$$U_{OUT}(s) = \frac{R_2 + \frac{1}{C.s}}{R_1}.U_{IN}(s) = \frac{R_2.C.s+1}{R_1.C.s}.U_{IN}(s) =$$
$$= \frac{T_2.s+1}{T_1.s}.U_{IN}(s) \quad (2)$$

The digital recursive equation [6] is (3).

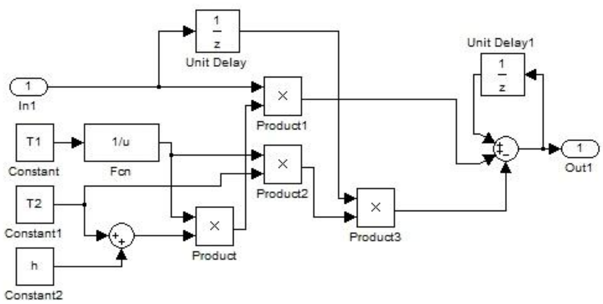

Figure 4 Matlab implementation of the PI regulator

$$U_{OUT}(k) = U_{OUT}(k-1) +$$
$$+ \frac{T_2}{T_1}\left[U_{IN}(k) - U_{IN}(k-1)\right] + \frac{h.U_{IN}(k)}{T_1} \quad (3)$$

The Matlab implementation of this regulator is demonstrated in Fig.4. The input *h* represents the sample time of the system.

C The Speed Estimator and Angular Speed Filter Device

Since the device is sensorless [1] the angular speed doesn’t have to be measured. But for the sake of speed feedback, angular speed has to be estimated. This estimation has been done by (4). It gives static relationship between the angular speed and anchor current and voltage.

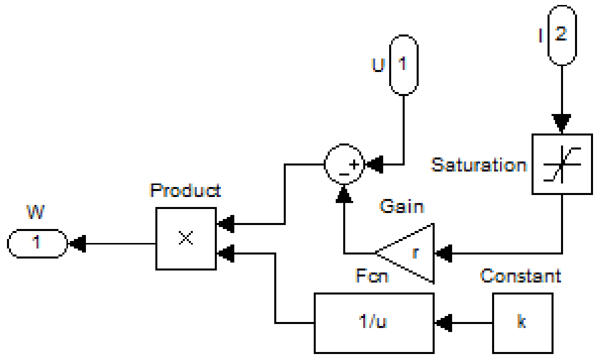

Figure 5 The Matlab Simulink model of the speed estimator device

$$\omega = \frac{U - I_a.R}{k} \quad (4)$$

It will be seen further in the final model that the lack of dynamical relation is not so badly reflected to the quality of the whole system. The Matlab Simulink model has been presented in Fig.5.

The estimator device is built in the feedback of the electrical drive. The speed value is a function of the anchor voltage and the anchor current. Every fluctuation of the arguments has made influence on the angular speed output - especially the current which is rapidly changing. High harmonics presented in the current will be propagated freely towards the angular speed output. That is why an aperiodic digital filter [7] is suggested at the output of the estimator. It has to cut high harmonics and noises coming from the current Hall measuring device. The transient function of this device is pointed out by (5):

$$U_{OUT}(s) = \frac{1}{T.s+1}.U_{IN}(s) \quad (5)$$

The recurrent equation which represents the digital form [7] of the filter is (6):

$$U_{OUT}(k)=\frac{T}{T+h}.U_{OUT}(k-1)+\frac{h}{T+h}.U_{IN}(k) \qquad (6)$$

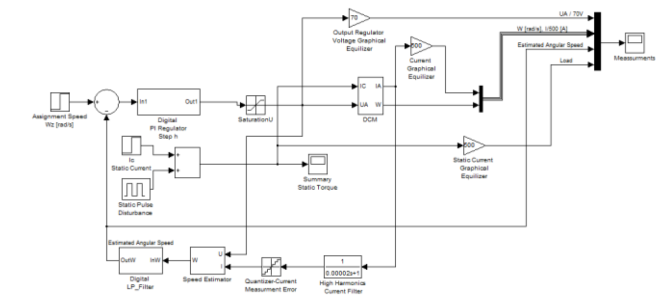

Figure 6 Matlab Simulink whole system model

This final equation will be used for the practical implementation.

D The Mathematical Model of the Whole System

The Matlab Simulink whole system model is pictured in Fig.6.

The identified DCM motor is included as subsystem DCM. The speed PI digital regulator is built in corresponding subsystem. The sample time of the regulator is equal to *h*. The speed estimator is also built into a subsystem. Due to its algebraic expression representation the speed value is filtered by Low Pass Filter in order fast current fluctuations to reduce their influence over angular speed value. The electrical drive load is performed by two elements [8], [9] – permanent and dynamic. These two components represent

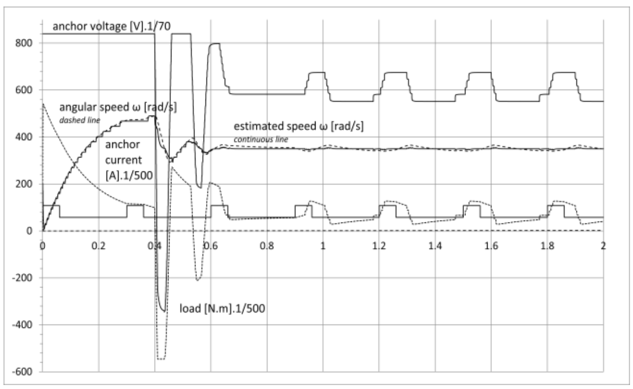

Figure 7 Speed regulation process simulation: (a)-anchor voltage x1/70[V], (b)-anchor current x1/500[A], (c)-regulated angular speed [rad/s], (d)-estimated speed [rad/s], (e)-load x1/500[N.m]

the friction of bearings [8], [9] and magnet stator influence.

Current quantizer determines the error of the Hall measuring [10] device.

The result of simulations has been done under following simulation circumstances: regulator time constants T2 = 0.05[s], T1 = 0.005[s]; sample time step h=1[mS]; $U_{Z\text{-}max}$=12[V]; I_{amax}=1.1[A]; ω_{amax}=350[rad/s]; load current I_{cmax}=200[mA].

It is well seen by the curves in Fig.7 that in spite of disturbances in the load the curve of regulated speed remains almost smooth and constant. The values of h, T1 and T2 have been obtained as a result of many simulations. These results are overlapped with the practical implementation.

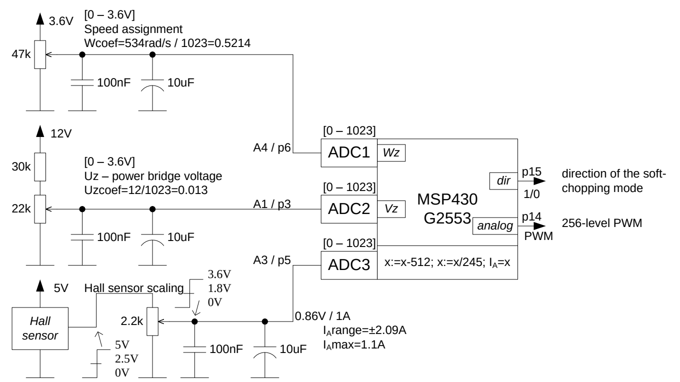

Figure 8 Hardware of the sensorless DC motor speed control

III Practical Implementation

The practical implementation of the electrical drive system consists of two major parts – hardware and software.

A Hardware of the System

The hardware unites the several scaled inputs/outputs and power electronic circuit in Fig.8.

Inputs are organized in three 10bit ADC converters and outputs consist of two 8bit DACs. In fact, DACs have been performed as PWM converters. The frequency of the PWMs is constant and the pulse width is determined by 8bits binary digit word.

Every input pin is scaled with a coefficient which reflects the physical input parameter entered inside of the processor like ω_z[rad/s]=$W_{coef}.W_{pot}$; U_z[V]=$U_{zcoef}.U_{zpot}$ etc.

The output pin value for PWM is taken by Gamma (duty cycle), transferring it from float to byte and after that filling OutGamma into the PWM output:

```
OutGamma = 255 * Gamma;
analogWrite(analogOutPin, OutGamma); // as PWM
```

The power circuit is presented in Fig.9 (based on L293B).

The DC motor can be switched in both ways, hard chopping, controlling q1 and q2 (both signals are inverted to each

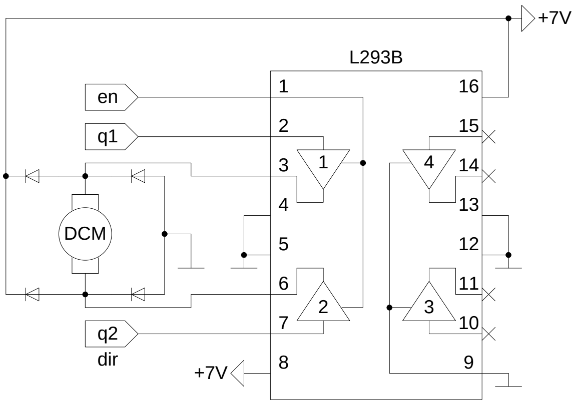

Figure 9 Power electronic circuit. The transistors at the output of the bridge are bipolar

other) and soft chopping, controlling q1 and dir. The signal dir inverts only during the change of motion direction. *En* signal enables the power circuit. It should be mentioned here that during the change of movement direction (dir) the code supplied to PWM should be complemented in order the absolute speed value to be kept intact.

B Software of the System

The sensorless DCM control program looks like this:

```
const int analogInPinWz = A4; //the potentiometer Wz is
                                   //attached to this pin
// Speed Assignment - Pin 6
const int analogInPinHall = A3; //Hall detector. Current
proportional readings. Pin 7.
const int analogInPinUz = A1; //Power Supply Voltage from
ADC. Reflects bridge power supply. Pin 3.
const float Wcoef = 0.5215; // =(524rad/s)/1024
const float Hall=245;          // = 512/2.09A - maximal
measured current 5.678A
// Sensitivity 0.317 V/A - determined by hall detector
const float UzCoef = 0.0130; // = 12V/1024 -12V max
bridge power supply
const int analogOutPin = 14; // Analog output pin
int dir =LOW; //soft chopping switch variable-points out the
direction of movement
const int digitalOutPin = 15;//soft chopping switch Pin
float Gamma = 0;              //duty cycle of the regulator
int OutGamma=0;               //duty cycle PWM presentation
int HallPot=0; //value read by Hall sensor
int Wpot = 0;  // value read by the Wz potentiometer
int UzPot = 0; // value supplied by the bridge potentiometer
float Wz=0;    // assigned speed normalized value [rad/s]
float I=0; // normalized current variable [A]
float Uz=0;// normalized power supply voltage [V]
float W0=0; // Estimator output
float A = 0; //Filter coefficient T/T+h
float B = 0; //Filter coefficient h/T+h
float Wold = 0;    // Filter output
float Wnew = 0;  // Filter input
float W=0;         //Speed variable
float U = 0;       //Anchor voltage variable
float Unew = 0;    //Regulator output variable in moment k
float Uold = 0;    //Regulator output variable in moment k-1
const float k = 0.020;       //Motor constant
const float R = 11.3;        // Anchor resistance
float dW = 0;                // Speed change in a step
const float T=0.001;         //LPF time constant
const float T1=0.05;         // bigger regulator time constant
const float T2=0.005;        // smaller regulator time constant
float ki =0;       // integral regulator coefficient
float kp = 0       // proportional regulator coefficient
float h=0.001;     // sample  time
void setup () {
  ki=T2/T1;        //regulator coefficients calculation
  kp = h/T1;
  pinMode(digitalOutPin,OUTPUT);
  Serial.begin(9600);
  A=T/(T+h);
  B=h/(T+h);
}
void loop() {
  Wpot = analogRead(analogInPinWz); // read speed as-
signment
  Wz=Wcoef*Wpot+50; // assignment normalized
  HallPot = analogRead(analogInPinHall); //read current
feedback
  HallPot = HallPot - 512;
  I=HallPot/Hall; //current normalized
  UzPot = analogRead(analogInPinUz); //read bridge power
supply
  Uz= UzCoef*UzPot; //supply is normalized
  W0=(U-I*R)/k; //estimate speed
  Wnew = A*Wold+B*W0; //W0 filtered
  W=Wnew;
  dW=Wz-W;
  Wold = Wnew;
  Uold=Unew; //refresh the regulator
  Unew=Uold+ki*dW+kp*Wz; //calculate new value
  if (Unew > Uz) Unew = Uz;
  if (Unew < -Uz) Unew = -Uz;
  Gamma = Unew/Uz; //Calculate duty cycle
  U = Gamma*Uz;
  if (Gamma >=0 ) dir =LOW; else dir = HIGH;
  digitalWrite(digitalOutPin,dir);
  if (Gamma < 0 ) Gamma=1+Gamma;
  OutGamma = 255 * Gamma; //Transfer gamma from float
to integer
  analogWrite(analogOutPin,OutGamma); //out gamma as
PWM
  delay (0.5); //extend sample time if needed
}
```

IV Experimental Results

Speed results are estimated by oscilloscope (see figures 10 to 15) which measures sinusoidal signal generated by

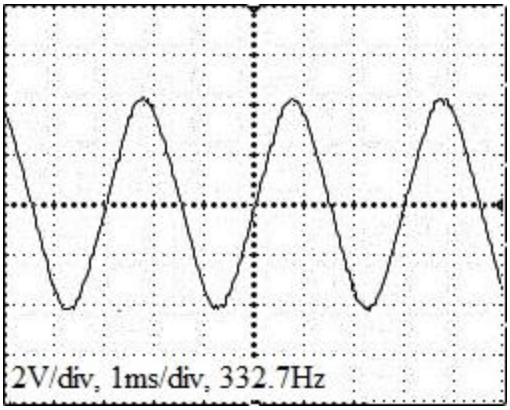

Figure 10 Highest motor speed: f/4 gives rot./s

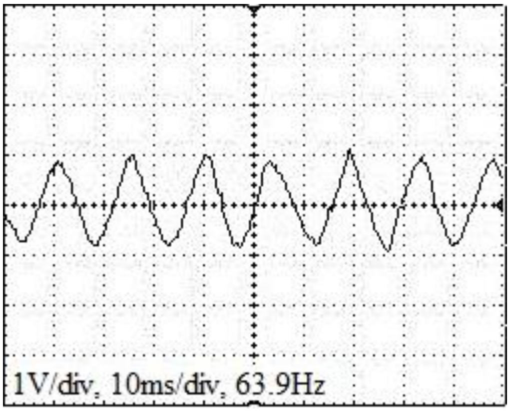

Figure 11 Lowest motor speed: f/4 gives rot./s

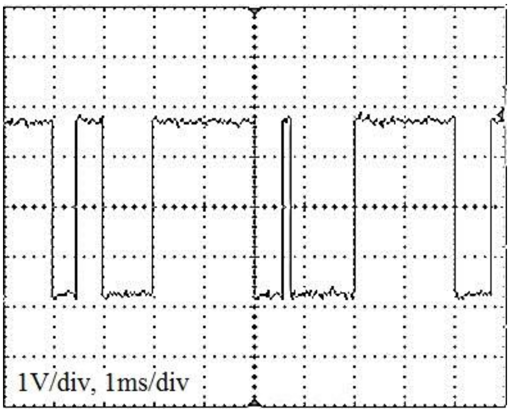

Figure 12 Processor PWM: it is rapidly changing

measuring winding embedded inside of the motor PIT 14/5. The loading torque is 0.004[N.m].

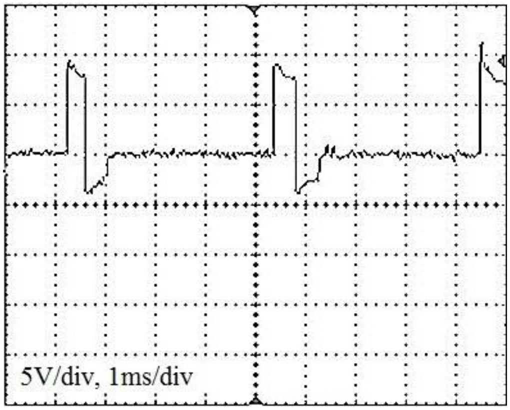

Figure 13 Power side soft chopping PWM. The upper spike corresponds to the upper bridge transistor saturation; the down spike reflects the down bridge transistor saturation.

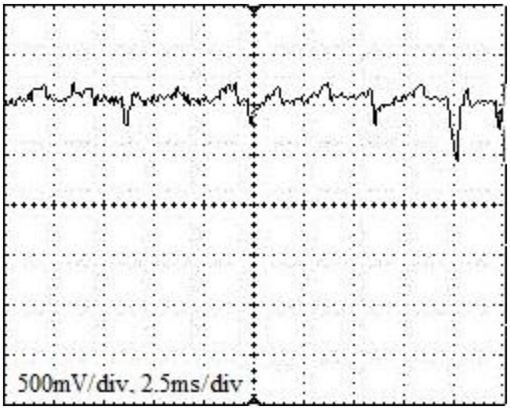

Figure 14 High motor speed anchor current. The variations of the anchor current are smaller in high speed.

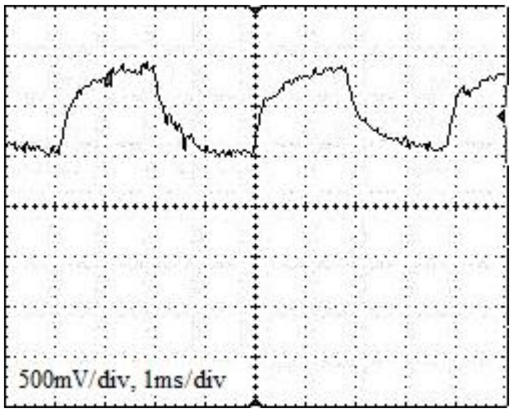

Figure 15 Low motor speed anchor current. Variations of the anchor current are bigger than in high speeds.

V CONCLUSION

The implemented DC motor speed control system gives opportunity for the following conclusions:

Due to the rather high preciseness and adequateness of the mathematical model, simulation system gives high predictability of the results. It hasn't been serious problem to predict the best sample time and time constants of the regulator.

It has been seen that one of the most significant factor for speed smoothness is accurate measuring of the anchor current. This accuracy is strongly dependant by the Hall detector range ability change. Since most of the low current industrial Hall detectors range is between ±5A it is hard to reach high accuracy in mA's range. This leads to comparatively low ratio factor between the high and low speeds. In this case a reduction of the current measuring range (by rewinding the sensor) has been reached: ±2.09A in case of 1.1A max current and accuracy of ±20mA. The relation between highest and lowest speed has reached factor of 7 in case of maximal load.

Since the measured current is very rich to high harmonics there is need to filter the current and eliminate them. But this filter should be low pass with very small time constant in order not to lose significant information by the anchor current. In spite of this filtering, fluctuations of the current influence significantly the angular speed after the estimator. This pointed out that there is need of low pass angular speed filter. It's time constant have to be much higher than first (current's) one but the bigger the time constant of LP the bigger instability of the system becomes, due to the phase deviation of angular speed. So the best time constant appears to be around 1-2ms. Obviously this filter is recommended to be digital. It has been predicted quite sharply its time constant and its significance to the given design. All this has been done by the Matlab model.

The optimal value of sample time has been determined by several Matlab model experiments searching for the best parameters of the regulator performance.Usually this value coincides with the value determined by the maximal computational power of the applying DSP.

It has been predicted that satisfactory sample time is equal to 1mS. It means the processor should be fast enough to reach these requirements. MSP430 is in the point of its limits but succeeded. The task will be facilitated if the controlled motor is of a greater power which means it would have a bigger time constant. Finally it can be made a conclusion that the computational power of this microcontroller is fully enough for the sensorless control of the DC motors with power more than 5-10 Watts.

REFERENCES

[1] C. Pavlitov, H. Chen, I. Colak, T. Tashev, Y. Gorbounov: Sensorless Control of SRM by the Aid of Artificial Neural Network Adaptive Reference Model. In: EPE 2011, Birmingham, United Kingdom, 2011.

[2] C. Pavlitov, L. Spirov, Y. Gorbounov, N. Stefanov: Parallel processing in electrical drives control. In: Computer Science'2006, Istanbul, Turkey, 2006.

[3] Y. Gorbounov, C. Pavlitov, R. Rusinov, A. Alexandrov, K. Hadjov, D. Dontchev: CPLD design of Parallel Processing Algorithms for Electromechanical System Control. In: Computer Science 2008, Kavala, Greece, 2008 (Best Paper AWARD).

[4] A. Lewis: A Mathematical Approach to Classical Control, 2012.

[5] J. Polderman, J. Willems: Introduction to the Mathematical Theory of Systems and Control, 2013.

[6] I. Landau, G. Zito: Digital Control Systems: Design, Identification and Implementation, Communications and Control Engineering, ASIN B001QFZ9FY, 2006.

[7] R. Allred: Digital Filters for Everyone: Second Edition, ISBN-10 1481084739, 2013.

[8] M. Chilikin: Obshii kurs elektroprivoda (in Russian), Energia, Moscow, 1971.

[9] P. Childs: Mechanical Design Engineering Handbook, ISBN-10 0080977596, 2013.

[10] K. Sozanski: Digital Signal Processing in Power Electronics Control Circuits, ISBN-10 1447152662, 2013.

Constantin Pavlitov. The author is an associate professor in the Department of Electrical Drives Automation at Faculty of Automation in the Technical University of Sofia. He delivers lectures in "Logical Control of Electromechanical Systems", "Computer Monitoring and Control of Electromechanical Systems" and "Microprocessor Control of Electrical Drives". He gives also lectures in the English language department of Engineering in the area of Computer Architecture included in the subject "Computing II". His scientific interests are in the field of fuzzy and neuro-fuzzy control of electrical drives and their identification with the aid of artificial neural networks.

Yassen Gorbounov. The author received a PhD degree from the Faculty of Automation at Technical University of Sofia, Department of Electrical Drives Automation in 2013. He is an assistant professor in the University of Mining and Geology "St. Ivan Rilski", Sofia and delivers lectures in "Measurement of nonelectric quantities" and "Microrpocessor systems". He is a member of the IEEE, Region 8, the Federation of the Scientific and Engineering Unions in Bulgaria (FNTS) and John Atanasoff Union of Automation and Informatics (UAI). He is an author or co-author of over 20 journal and conference papers and is a co-author of 1 book. His research interests include automatic control of electrical drives, switched reluctance motor and generator control, application of neural networks and fuzzy logic for motor control, parallel processing algorithms with programmable logic devices.

Radoslav Rusinov. The author is an assistant professor in the University of Mining and Geology "St. Ivan Rilski", Sofia. He received his MSc in 1999 from Technical University of Sofia. His research interest include control systems, electrical drives, fuzzy logic, neural networks.