Embedded target toolbox for DSP control applications of BLDC motor

Calin Rusu[*], Alexandru Bara[**] and Benk Enikö[*]
[*] Department of Electrical Drives and Robots,
Technical University of Cluj-Napoca, Faculty of Electrical Engineering,
15 C. Daicoviciu Street, 4000020 Cluj-Napoca, Romania, E-Mail: calin.rusu@edr.utcluj.ro

[**] Department of Electrical Drives and Automation,
University of Oradea, Faculty of Electrical Engineering and Information Technology,
1 Universitatii Street, 410087 Oradea, Bihor, Romania,, E-Mail: albara@uoradea.ro

Abstract - ***This paper presents a Matlab embedded target toolbox used for the DC brushless (BLDC) motor applications. The synthesis, code generation, and the implementation of the control program, and also the control task itself are carried out on the host PC, under the Simulink. The target system is a brushless DC motor control kit – MSK243, connected on serial port COM to the host computer. With this real time library, it is possible to develop a Rapid Control Prototyping and Hardware-in-the-Loop Simulations. During the running control task on the target, the full functionality of Matlab/Simulink can be used for parameter's visualization without interrupting or impeding the control process on the MSK240 board.***

Keywords: ***embedded toolbox, BLDC motor, DSP control***

I. INTRODUCTION

The Matlab facility offered by its graphical user interface (GUI) Simulink, becomes a standard software tool for controller design and simulations in the control of closed-loop systems. Graphical editing followed by the processing of data simulation it is also possible. The Real-Time Workshop generates C code automatically from a Simulink block diagram. However, the usual ways to implement the resulting control program, from such a Simulink scheme, on target hardware, in most cases, are made "by hand". This process needs more time and could be a considerable source of errors. The solution offered by dSPACE is a good one for the complex systems. For the most universities in their educational purposes and even for the simple industrial applications there is no need for such a high complex hardware and software system, these are anyway too expensive. Present real time library – (MCK243 toolbox) is a convenient solution between the full capability of the motion control kit MCK243 as a closed loop control system on a side, and the output functionality of Matlab / Simulink as control unit on the other side. The control program, for the target MCK243 system, is generated from a Simulink block diagram "by simple click a button" on the host PC and after than, it could be downloaded and executed on the MCK243 system that is connected via COM port to the PC. By means of Matlab and Simulink, which run on the host PC in parallel, it is possible to do user inputs, parameter changes and visualization of measurement data. The real time control should nevertheless not be affected by the necessary communications between MCK243 and PC. The MCK243 system uses a TMS320C243, one of the most popular 16-bit fixed point DSP, used in motor control applications. It can, among other things, be programmed in the high-level programming language C. Next sections will presents the real time library – MCK243 rtlib, and the new Simulink blocks of this toolbox are explained. By means of this toolbox, graphical programming and controlling of the TMS320C243 DSP via Matlab / Simulink are possible As a model example, the speed control application of a brushless DC motor is presented. Simulation of the behavior control system and experimental results are finally presented and compared.

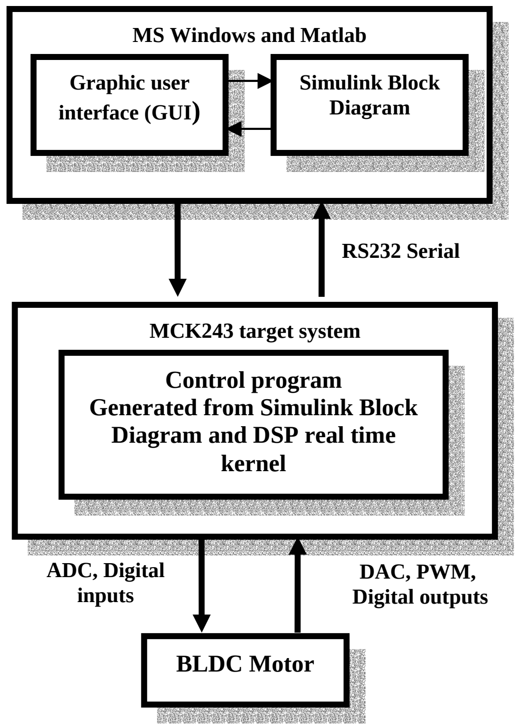

Figure 1. System architecture and structure

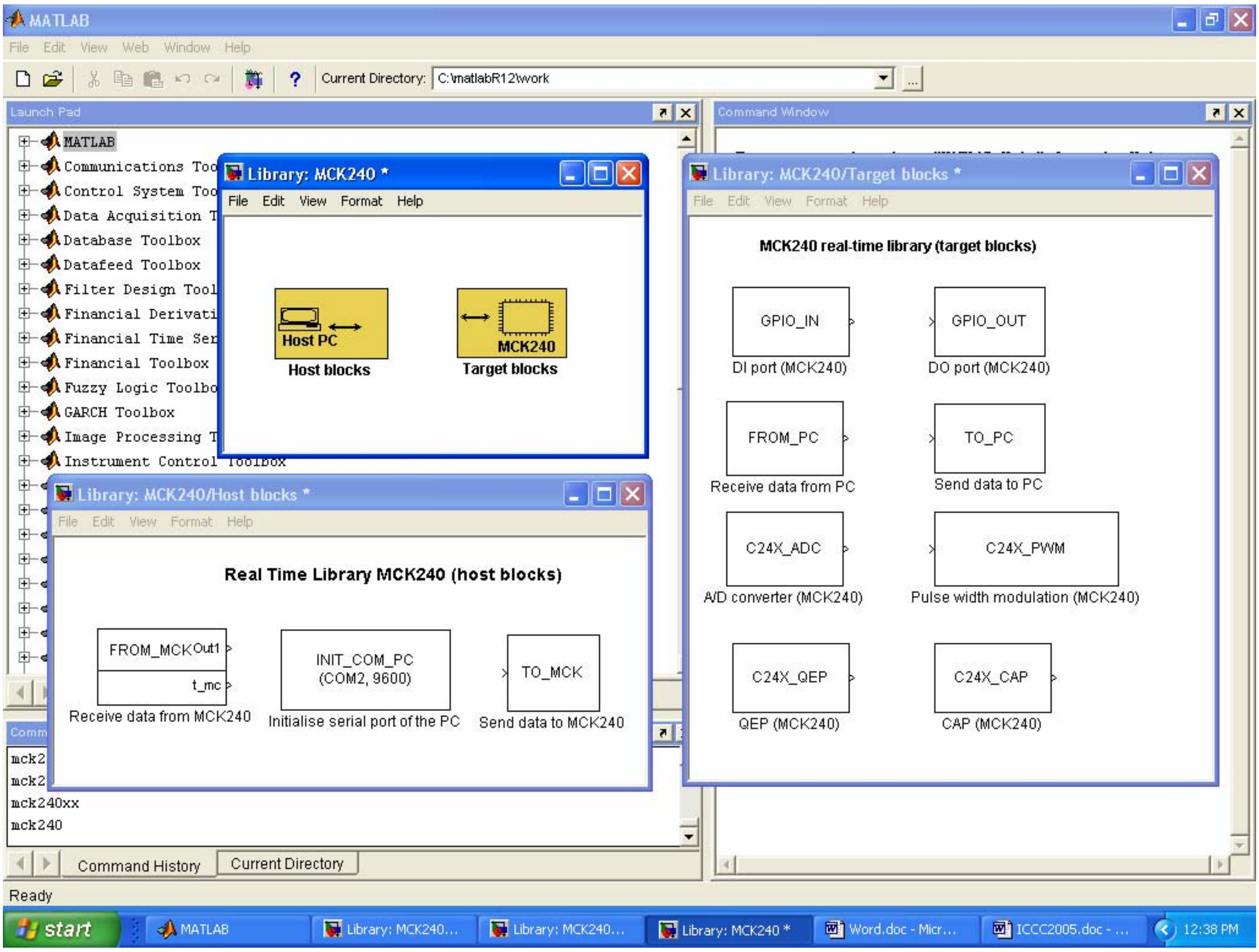

Figure 2. Main blocks for MCK243 Real Time Library (host and target blocks)

II. SYSTEM ARCHITECTURE

The developed system consists of two separate subsystems as is shown in Fig. 1. The main is the host PC computer on which Matlab is running under Windows, and the last one is the target system - MCK243 hardware, which will runs the control program. Both systems are connected via RS232 port (COM1 or COM2), so as they can exchange the data one each others.The communication is interrupted driven and buffered on both sides. It works according with the client-server principle. The PC sends the given data to the MCK243 and requests the measurement data which are delivers by the target system. The MCK240 itself is not able to initiate a communication with the PC. The communication itself has a lower priority than of the control task which runs on the MCK243, in order to avoid the blocking of control. In the worst case, the communications will be slightly delayed. This situation could be accepted because each measured value is sent associated with the absolute time of the measurement to the host PC. In this way an undistorted graphical output or data logging can be achieved under Matlab. Communications are done by “messages”, of which structure consist of a “header” as identification and a “body”. The length of the body depends on the message itself.

To open the library blockset, at the Matlab prompt, you must type *MCK240rtlib* which contains libraries of blocks designed for MCK243 target board. The libraries are in two groups, one is the host blocks and the others is the target blocks.

III. BRUSHLESS DC MOTOR MODEL

Brushless DC motor considered in this paper is a surface mounted, non-salient pole, permanent magnet (PM) synchronous machine with trapezoidal flux distribution in the air gap. This kind motor is very attractive in servo and/or variable speed application since it can produce a torque characteristic similar to that of a permanent magnet DC motor while avoiding the problems of failure of brushes and mechanical commutation.

Basic voltage current relations can be written as:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_a & L_{ba} & L_{ca} \\ L_{ba} & L_b & L_{cb} \\ L_{ca} & L_{cb} & L_c \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (1)$$

where

R	Winding resistance per phase
L_a	Self inductance per phase *a*
L_{ba}	Mutual inductance between phases *b* and *a*
e_a	Per phase *a* EMF
V_a	Per phase *a* voltage

and

$$L_a = L_b = L_c = L$$
$$L_{ab} = L_{bc} = L_{ca} = M \ .$$

Due to symmetry in the isolated neutral configuration of stator windings of the motor we have that:

$$i_a + i_b + i_c = 0 \quad (2)$$

$$Mi_b + Mi_c = -Mi_a \ . \quad (3)$$

Using the above equations, the BLDC model can be reduced to the following state space form:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} -\frac{R}{L_s} & 0 & 0 \\ 0 & -\frac{R}{L_s} & 0 \\ 0 & 0 & -\frac{R}{L_s} \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \quad (4)$$

$$+ \begin{bmatrix} -\frac{1}{L_s} & 0 & 0 \\ 0 & -\frac{1}{L_s} & 0 \\ 0 & 0 & -\frac{1}{L_s} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - \frac{1}{L_s} \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}$$

where

$$L_s = L - M \ . \quad (5)$$

The torque and mechanical equations are

$$T_e = \left(\frac{P}{2}\right) \frac{e_a \cdot i_a + e_b \cdot i_b + e_c \cdot i_c}{\omega_r} \quad (6)$$

$$J\left(\frac{2}{P}\right) \cdot \frac{d\omega_r}{dt} + B\left(\frac{2}{P}\right)\omega_r = T_e - T_L \quad (7)$$

and

$$\omega_r = \omega_m \left(\frac{P}{2}\right) , \quad (8)$$

respectively.

IV. BRUSHLESS DC MOTOR CONTROL

A common way to control BLDC motor is through the use of a voltage source current-controlled PWM inverter. The inverter must supply a quasi-square current waveform, of which magnitude is proportional to the motor shaft torque [2]. Then, by controlling the phase currents, torque and speed can be adjusted. Fig. 3 shows the BLDC motor fed by the PWM inverter. It receives PWM signals from the drive circuit behind, based on the converted current feedback of Hall sensors and position encoders.
The Hall sensors are placed on the stator of BLDC motor that give 120^o phase shifted square waves in phase with respective phase voltages. These waves represent the rotor position. The decoder circuit converts these waves into the six-step command signals. Converted signals with appropriate polarities are than used to drive the gates of the MOSFETs (S_a+, S_a-,…,S_c+, S_c-) of the respective phases.

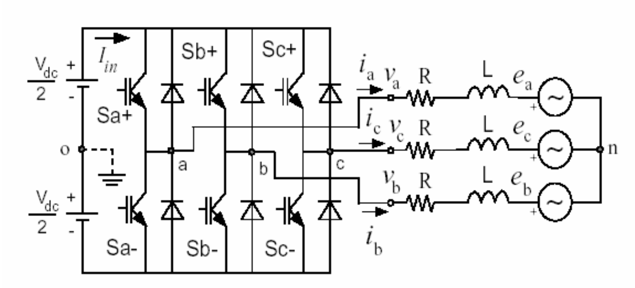

Figure 3. BLDC current controlled PWM inverter

The actual phase currents of the rotor, track the command currents by hysteresis-band current control. At any instant time, only two of the phase currents are enabled, one with positive polarity and the other with negative polarity. When these currents (equal in magnitude) tend to exceed the hysteresis band, both the MOSFETs are turned off at the same instant to initiate current feedback through the free wheeling diodes. The currents i_a, i_b and i_c needed to produce a steady torque without torque pulsations and the back EMF's e_a, e_b, e_c are shown in Fig. 4.

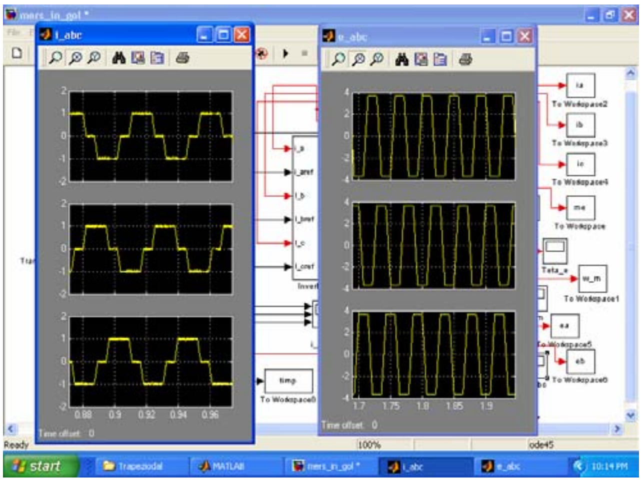
Figure 4. Currents and back EMF of the BLDC motor

The developed torque is directly proportional to the phase current. It means that the torque can be maintained constant by a stable supply current. In order to realize this state, one method is to keep the ratio of voltage to frequency a constant. This can be achieved by using a speed and current closed-loop control. Fig. 5 shows the overall block diagram of the developed model for BLDC drive system. The simulation is performed using the Matlab/Simulink software. The control structure has an inner current closed-loop and an outer-speed loop to govern the current. To control the rotor speed - ω_m, we have developed a PI controller, The controller regulates the rotor speed by varying the frequency of the pulse based on signal feedback from the Hall-ICs.

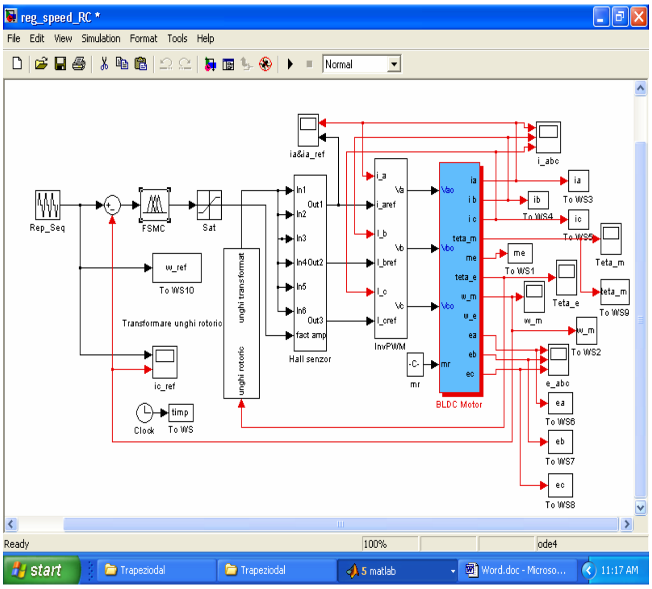

Figure 5. Block diagram of speed control system

Fig. 6 presents the simulation results for the current windings (i_a, i_b, i_c), back EMF's (e_a, e_b, e_c) and the rotor speed (ω_m), respectively.

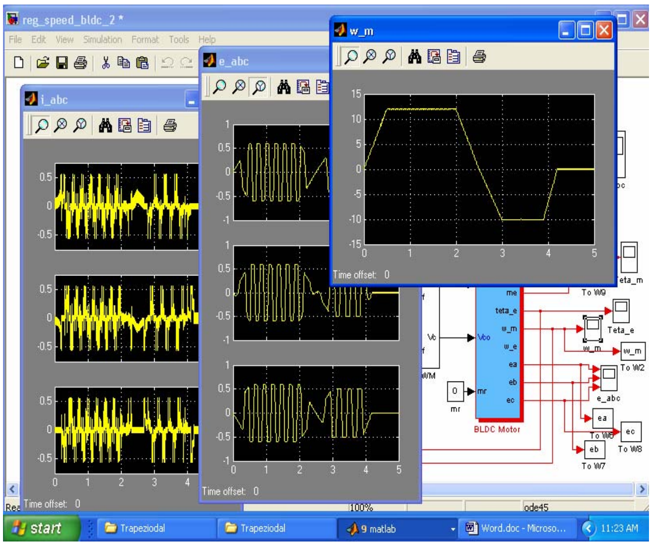

Figure 6. Current in BLDC phases, back EMF voltages and speed

The speed feedback loop compares the rotor speed ω_m with the desired rotor speed ω_{ref} to produce the command torque component current. If the speed is faster than desired speed it will reduce the frequency of the pulse to be injected by the inverter. Vice versa, if speed is slower frequency will be increased.

VI. DSP CONTROLLER

The target system is a MCK243 is composed by a Pittman BLDC 3400 and a DSP TMS320F243 based controller. Fig. 7 presents a picture of the system drive. The motion system includes basic motor control with closed loops for current and speed control.

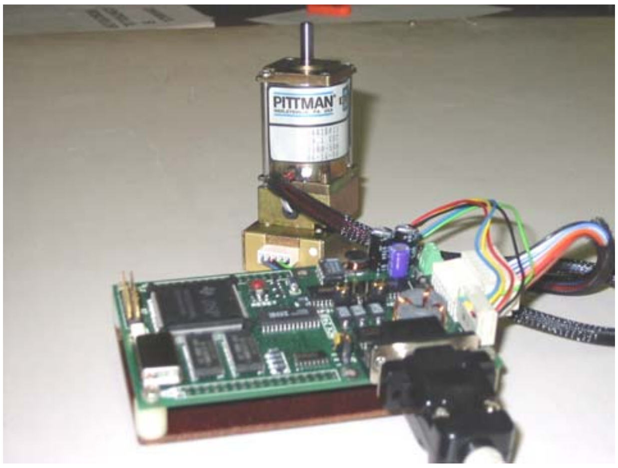

Figure 7. MCK243 target controller with BLDC motor

The MCK system integrates the power electronics peripherals – 12 PWM channels, three 16 bit multi-mode general purpose timers, 16 channel 10 bit ADC with simultaneous conversion capability, four capture pins, encoder interface capability, SCI, SPI, Watch Dog etc. Six PWM channels (PWM1 through PWM6) control the three-phase voltage source inverter. The target library block can be seen in Fig. 2 left side.

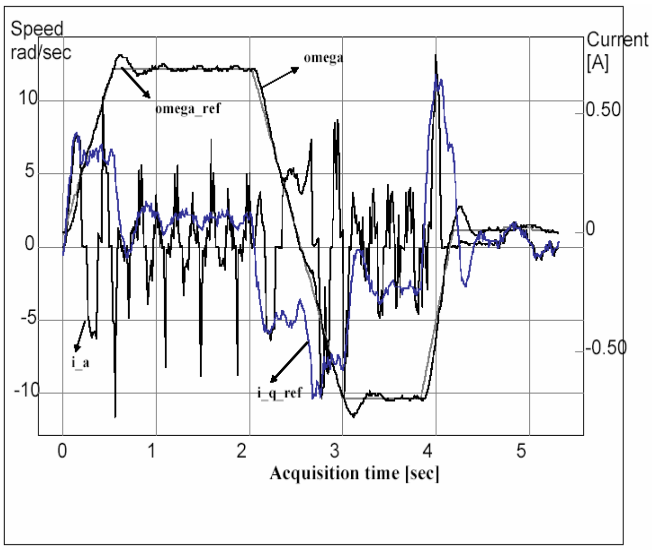

Figure 8. Speed ω_m and current phase i_a

The control program, for the target MCK243 system, is generated from the Simulink block diagram presented in Fig.5, on the host PC. After than, it is downloaded and executed on the MCK243 system. Matlab and Simulink, runs on the host PC in parallel with the MCK243 control task, being possible to exchange parameters and data one each other. In this way, Fig. 8, presents a graphical output for the BLDC rotor speed - ω_m and current winding - i_a, as measured data by the target system.

VII. CONCLUSIONS

In this paper a real-time Matlab toolbox used to control a DC brushless motor is presented. The control scheme, code generation and the implementation of the control program for the target system are carried out on the host PC, under the user interface of MATLAB. The target system is a motor control kit – MCK243, connected on serial port COM to the computer. The result of simulation and experiment shows that the real time library is useful in Rapid Control Prototyping and/or Hardware-in-the-Loop Simulations for the target MCK243 system.

VIII. REFERENCES

[1] MCK243 User Manual, Technosoft SA, CH-6977 Ruvigliana, Switzerland.

[2] MSK243 / MCK243 User Manual, Technosoft SA, CH-6977 Ruvigliana, Switzerland.

[3] DMCD-Pro User Manual, Technosoft SA, CH-6977 Ruvigliana, Switzerland.

[4] Embedded Target for the TI TMS320C2000™ DSP Platform, For Use with Real-Time Workshop, The MathWorks Web site http://www.mathworks.com.

[5] Jong-Bae Lee, Tae-Bin Im, Ha-Kyong Sung, Young-Ouk Kim "A Low Cost Speed Control System of Brushless DC Motor Using Fuzzy Logic", IEEE Trans. Ind. Application, vol.2, 1999, pp.433-436.

[6] Rapid control prototyping approach to fuzzy speed control of brushless dc motor, R. Nejat TUNCAY Zeynep ERENAY Murat YILMAZ Özgür USTUN, *Istanbul Technical University, Faculty of Electrical & Electronics Engineering, Department of Electrical Engineering,34469, Maslak, Istanbul, Turkey*

[7] Rusu, C. Birou, I., *Model Based design Controller for the Stepper Motor*, Proceedings of 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2008, May 22-25, 2008, Cluj-Napoca, Romania, Tome II, pp.175-179, IEEE Catalog Number: CFP08AQT-PRT, ISBN 978-1-4244-2576-1, Library of Congress 2008904446.