Model based design controller for the stepper motor

Conference Paper · June 2008

DOI: 10.1109/AQTR.2008.4588816 · Source: IEEE Xplore

Calin G. Rusu
Universitatea Tehnica Cluj-Napoca

Iulian Birou
Universitatea Tehnica Cluj-Napoca

Enikő Szőke Benk
Universitatea Tehnica Cluj-Napoca

Model Based Design Controller for a Stepper Motor

Calin RUSU[1], Iulian BIROU[1], Enikö SZÖKE[1]
[1]Technical University of Cluj, Dept. of Electrical Drives and Robots, Cluj, Romania

Abstract - **The motivation of this work is to make a library for an embedded system useful in implementing of the field-oriented PM stepper motor control. The tool is based on DSP processor board, a hardware component oriented for embedded applications and Matlab/Simulink environment. Matlab is a standard in rapid prototyping of the control applications, but it doesn't have an adequate hardware support. The target controller is connected via serial port to the host PC computer. An integrated development toolbox having a distributed nature and real-time requirements was developed. It is useful for Rapid Control Prototyping and for Hardware-in-the-Loop Simulations. The full functionality of Matlab/Simulink can be used for parameter's visualization without interrupting or impeding the real time control process of the embedded microcontroller. Finally, the field-oriented for a two-phase PM stepper motor application is presented in order to demonstrate the effectiveness for this real time embedded controller.**

I. Introduction

The Matlab facility offered by its graphical user interface Simulink, becomes a standard software for simulation and control system design. Also, from the Simulink block diagram the C code for the target controller can be automatically generated. However, the usual ways to implement the embedded code control, on the target hardware, for the most cases are made "by hand". This process needs more time and could be a considerable source of errors. For the most industrial applications present solution is less expensive than others.

Because the Matlab/Simulink runs on the host PC in parallel with the embedded control task which runs on the target, it is possible to exchange parameter and to visualized the measurement of data. The real time control is not affected by the communications between controller and PC. Next sections will present the system architecture, the real time library and the new Simulink blocks. The field-oriented control applied to a two-phase PM stepper motor application is presented as an embedded code control generation of this real time library. Simulation and experimental results are finally presented.

II. Development System

The developed system consists of two separate subsystems as is shown in Fig. 1. The MSK target system architecture is based on a TMS320F243, a most popular 16-bit fixed point DSP, used in motor control applications. It can be programmed in the high-level programming language C.

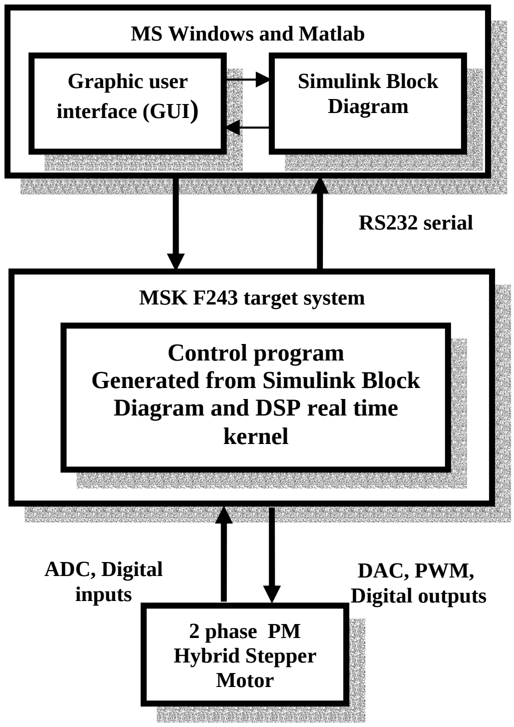

Figure 1. System structure

The master (server) is the host computer (PC) on which Matlab is running under Windows, and the slave (client) is the target MSK F243 controller, on which the program is running. Both systems are connected via RS232 serial port, so as they can exchange the program and/or data one each others.

The communication works according with the client-server principle. This is 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. The PC sends the given data to the MSK F243 and requests the measurement data which are delivers by the target system. The MSK itself is not able to initiate a communication with the host PC. The communication itself has a lower priority than of the control task which runs on the MSK controller, in order to don't loose the real time control. In the worst case, the communications will be slightly delayed. In this way an undistorted graphical output or data logging can be achieved under Matlab.

III. Embedded Target Library

Matlab allows engineers to develop a control application algorithm in the high level graphical language of data-flow and state-flow diagrams. The Simulink toolbox contains many libraries of whose elements can be embedded in Simulink block diagrams. Simulink also provides user-defined blocks, in the form of s-function blocks, which can be modified to perform user-defined tasks. Furthermore, every Simulink block allows for a set of “callback functions,” which are executed upon the specific events, when running a Simulink block diagram.

Present library is a bridge between the full capability of the MSK F243 embedded DSP controller, as a closed loop control system on a side, and the functionality of Matlab / Simulink environment, as control unit on the other side. The library blocks are generated as the C MEX S-functions and provide a hardware abstraction layer allowing the access to peripherals of the DSP embedded controller (ADC, PWM, Timer, QEP, SPI, SCI DI, DO). Each of these blocks defines its simulation behavior and provides a user interface for the parameter settling. The blocks can be directly used by Simulink so as the graphical programming and controlling for the target system via Matlab/Simulink becomes possible. The developer does not need to study all details relating to control registers of DSP peripherals. He/she only specifies the fundamental parameters (e.g. the resolution of ADC, the input pin, the conversion time, the mode of operation) and selects high level methods and events to access the peripheral.

The code generated by Real Time Workshop (RTW) for each of these blocks is defined by a Target Language Compiler (TLC) script in a *tlc* file. During the code generation, a code is generated for each block in the model according with the corresponding *tlc* file. These programs are combined according to the data flow in the model and finally, a make file is generated from a predefined template.

As a result, the code is build-up and the executable application can be downloaded to the target embedded system. The real time control application is ready for execution on the MSK F243 microcontroller board under the full control of PC host.

To open the MSKlib F243 library, at the Matlab prompt you must type *MSKF243*. After that, a *MSK F243lib* window will appear like in Fig 2. The Simulink functional blocks of the library has two icons, one is for the PC host (server) and the other is for MSK F243 target system (client). The library blocks provide a high-level access to the hardware units of the MSK F243 board as is shown in Fig. 3a and 3b, respectively.

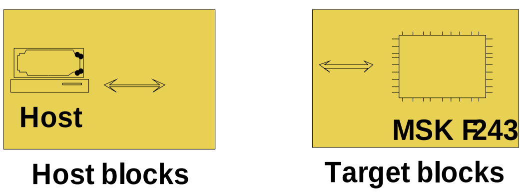

Figure 2. MSK F243 library

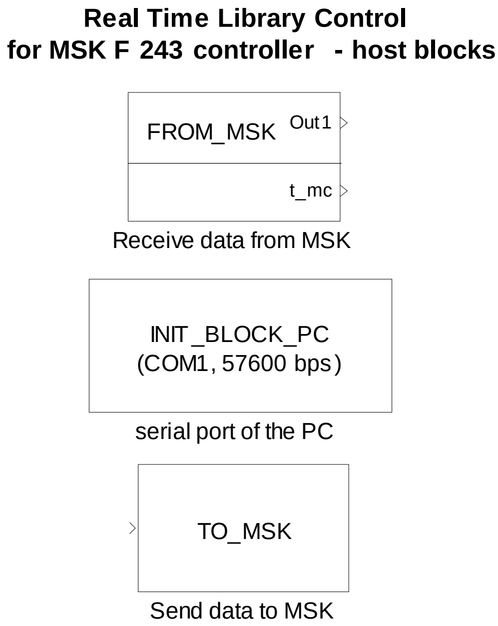

Figure 3a. Embedded DSK24X toolbox (host blocks)

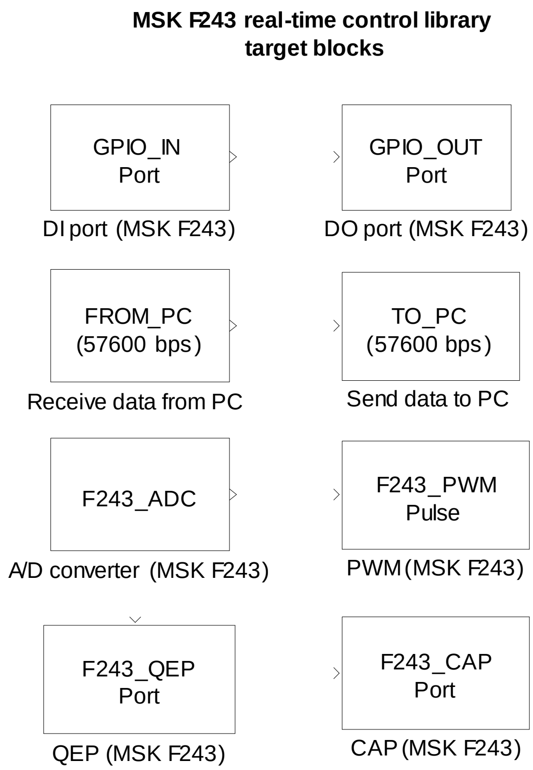

Figure 3b. Embedded MSK F243 toolbox (target blocks)

The C code can be automatically generated from the Simulink model by using of Real-time Workshop (RTW). A block diagram based on the embedded target blocks can be converted directly into a MSK F243 executable task. During an initial stage, Real-Time Workshop translates the block diagram into a series of standard (ANSI) C code files. These files are then cross-compiled to the TMS320F243 executable.

IV. PM HYBRID STEPPER MOTOR CONTROL

The PM hybrid stepper motor was designed as an *ac* two-phase synchronous motor for high torque and low speed application. By neglecting the space harmonics, the control loops for torque and position can be designed in the similar manner as for the brushless motor, if the well-known *d-q* transformation is applied. The *d*-axis is chosen on the flux axis while the *q*-axis is π/2 ahead in the rotor direction. According to this definition, the *d*-axis corresponds to the maximum flux linkage. The *d-q* frame is rotating synchronously with the magnetic field. As results, the motor becomes a high-dynamic AC servo drive and produce high torque at a given winding current (because of their many poles).

This is a desirable solution for low-speed direct drive applications where more powerful actuators in small size are required. In such cases the force-speed conversion by gear drive is not necessary.

The controller for the PM hybrid stepper motor consists of a two current controllers. The *q*-current is directly proportional with the torque of the motor. From the computational point of view the control algorithms request a high-speed controller system. MSK F243 DSP controller is a good choice for this application.

The dynamic properties for the hybrid stepper motor may be described by a set of non-linear differential equations, linking the stator currents and the voltages with the mechanical quantities - torque, speed and angular position, [2], [3], [4].

The analysis and control of such a motor appears to be complicated because of the coupling of all control inputs. The problem can be overcome by applying the field-oriented control of the quantities, which reduce the control of a PM hybrid stepper motor to that of a separated *dc* motor. Applying *dq* transformation the stepper motor model becomes:

$$\begin{aligned}
\frac{di_{sd\theta}}{dt} &= \frac{1}{L_{sd}}\left(v_{sd\theta} - R\cdot i_{sd\theta} + \omega_m \cdot L_{sq}\cdot i_{sq\theta}\right)\\
\frac{di_{sq\theta}}{dt} &= \frac{1}{L_{sq}}\left(v_{sq\theta} - R\cdot i_{sq\theta} - \omega_m \cdot L_{sd}\cdot i_{sd\theta} + \omega_m\cdot\Psi_M\right)\\
\frac{d\omega_m}{dt} &= \frac{1}{J_m}\left(K_t\cdot i_{sq\theta} + (L_{sd}-L_{sq})\cdot i_{sd\theta}\cdot i_{sq\theta} + B_m\cdot\omega_m\right)\\
\frac{d\theta_m}{dt} &= \omega_m
\end{aligned} \qquad (1)$$

where L_{sd}, L_{sq} are the direct and quadrature components of inductance; $i_{sd\theta}$, $i_{sq\theta}$ are the direct and quadrature currents and $v_{sd\theta}$, $v_{sq\theta}$ are the direct and quadrature voltages.

The resulting system (1) is still nonlinear. Fig. 4 shows the variable reluctance influences which are represented through the cross coupling terms $z_r\cdot\omega_m\cdot L_{sd}\cdot i_{sd\theta}$ and $z_r\cdot\omega_m\cdot L_{sq}\cdot i_{sq\theta}$ respectively.

The reference voltages will be chosen $v^*_{sd\theta}$, $v^*_{sd\theta}$ while the actual de-coupling applied voltages will be $v_{sd\theta}$, $v_{sq\theta}$. The cross coupling effects can be canceled by feedforward compensation as shown in Fig. 5.

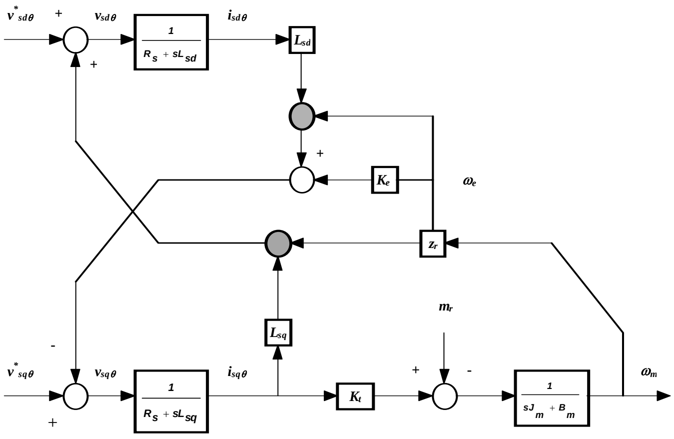

Figure 4. Block diagram of the *d-q* model

Taking a current-controlled voltage source inverter, the reference voltage commands are decided by the *d-* and *q-axis* compensators and by the decoupling feedforward compensation:

$$\begin{aligned}
v^*_{sd\theta} &= v_{sd\theta} - \omega_m \cdot L_{sq}\cdot i_{sq\theta}\\
v^*_{sq\theta} &= v_{sq\theta} + \omega_m \cdot L_{sd}\cdot i_{sd\theta}
\end{aligned}. \qquad (2)$$

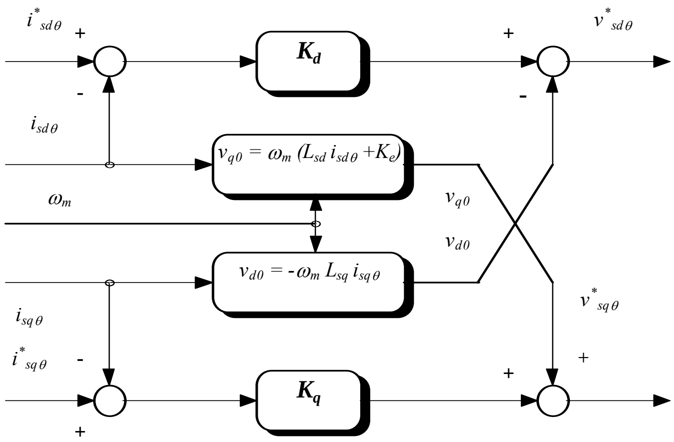

Figure 5. Decoupling current controllers

Now the *d-* and *q*-axis current control loops are decoupled.

Based on previous theoretical considerations the control scheme for the field-oriented PM hybrid stepper motor drive is shown in Fig. 6. In this control method the magnitude of the current flowing into the windings are controlled using the loop with current feedback. The current in the motor phase windings are directly measured with the current sense resistors connected in series with the phases. The pulse width modulation signals with varying duty cycle are used for currents regulation. The currents are compared with the desired value of currents (references), forming the error signals. Than the currents error are compensated via the PID regulators and the appropriate control action are taken.

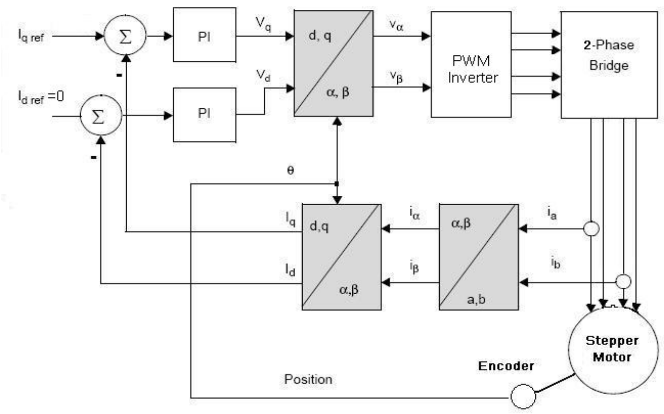

Figure 6. Control scheme for PM hybrid stepper motor drive

The application is developed as a Simulink model.and it consists of the controlled subsystem (stepper motor) and the controller subsystem presented in Fig. 7.

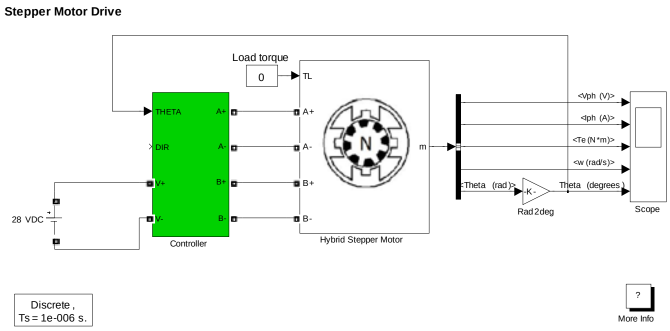

Figure 7. Simulink block diagram

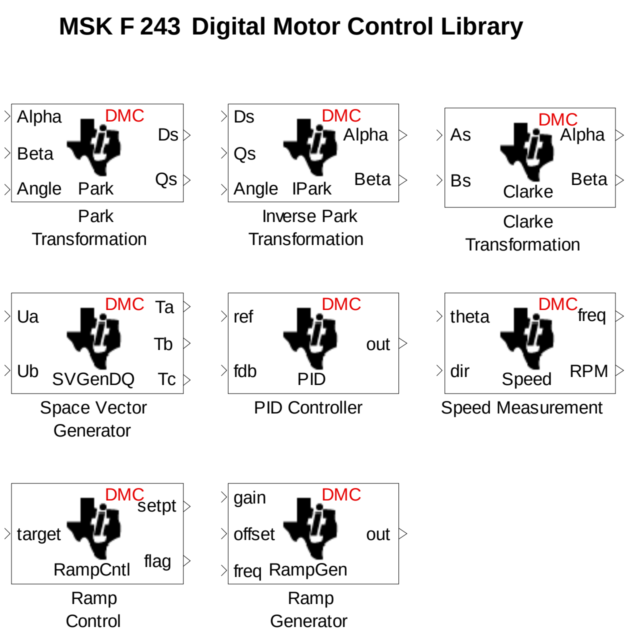

Figure 8. Digital Motion Control Library

In order to make the implementation for the controller we used the Digital Motion Control Library. Fig. 8 shows the blocks of this library.

V. Experimental Results

The embedded controller structure is presented in Fig. 9 and a picture of the embedded system connected to the PWM inverter unit is presented in Fig. 10.

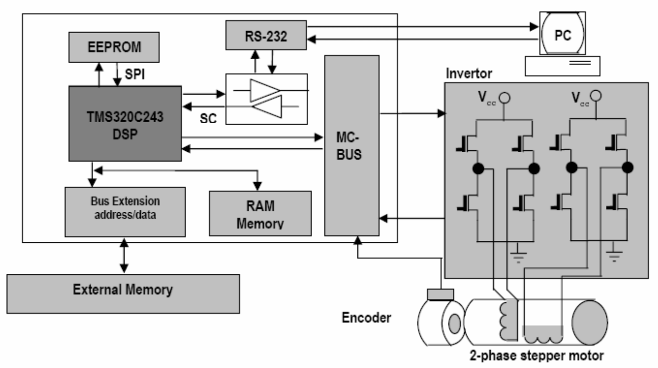

Figure 9. Block diagram of the embedded controller and PWM unit

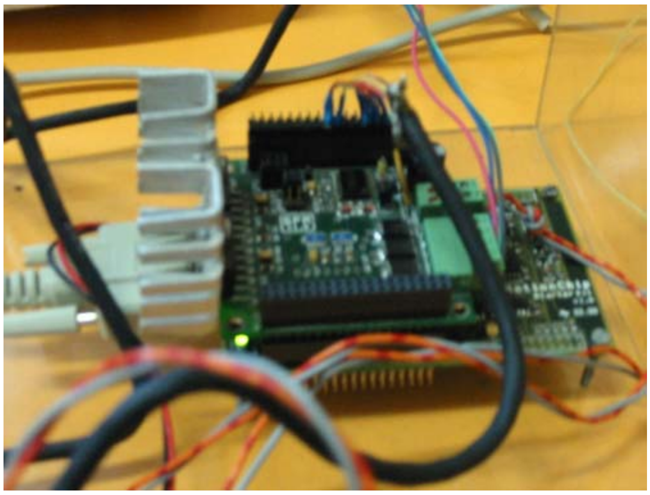

Figure 10. Embedded controller and the PWM inverter

The parameters of the PM hybrid stepper motor which was used for experimental results are as follows: 200 steps/rotation z_r=50 rotor teeth, i_N=2.0A, m_N=0.3Nm, R=3Ω, L=0.02mH, B_m = 3x10^{-3} Nm/s/rad, J_m= 1.25x10^{-4} kgm^2, ψ_{PM} =0.0044Wb.

The acquisition of phase currents (i_a and i_b), made by the F243_ADC target block of the library is shown in Fig. 11.

Four PWM channels (PWM1 through PWM3) are used to control the voltage source inverter. The F243_PWM library function is used to set-up the parameters for PWM unit. The PWM pulses for one phase looks like in Fig. 12.

The rotor the position is detected by F243_QEP block of the MKS system. The pulses are provided by the motor encoder and these are captured by the QEP unit.

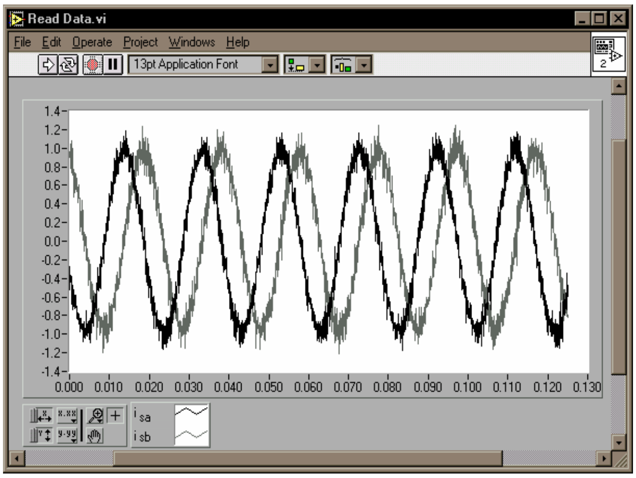

Figure 11. Currents in the motor phases - i_a and i_b

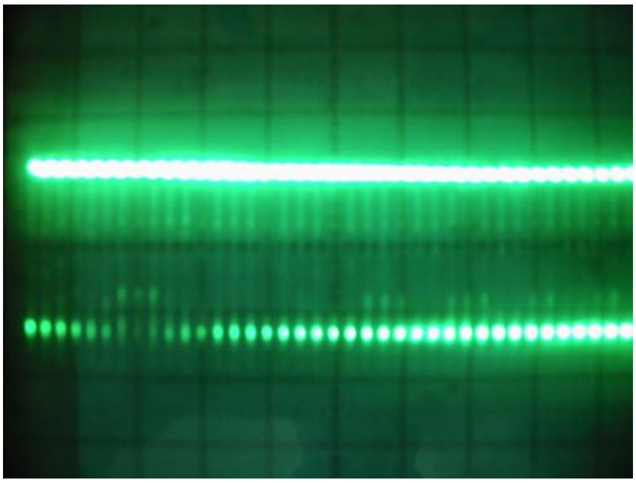

Figure 12. PWM pulses for one motor phase

Based on the communications between embedded system and PC, in the Fig. 13 the acquisitions for the currents and position were recorded and visualized by the master system into a graphical windows.

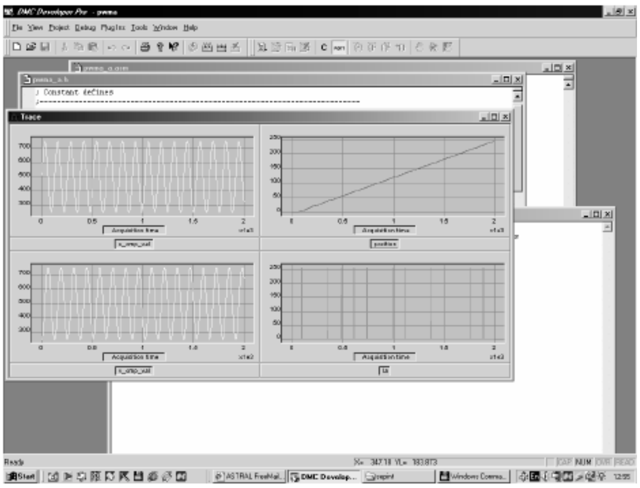

Figure 13. Acquisition for currents and position are recorded and visualized on a graphical user interface

VI. CONCLUSIONS

An embedded toolbox for a target MSK F243 controller was developed under Matlab/Simulink. The tool is a bridge between the hardware component oriented for embedded applications and Matlab/Simulink environment.

The field-oriented control for a 2 phase stepper motor was implemented as application. The pulse width modulation signals with varying duty cycle are used for currents regulation. The currents are compared with the desired value of currents (references), forming the error signals. Than the currents error are compensated via the PID regulators and the appropriate control action are taken.

The control scheme and code generation for the target the system is carried out on the host PC, under the SIMULINK and Real Time Workshop, respectively. The target system is a MSK F243 system, connected on serial port COM to the host computer. The Matlab environment is also used for parameter's visualization without disturb the real time control of the target system. The result shows that the embedded toolbox can be used in Rapid Prototyping and/or Hardware-in-the-Loop Simulations for the target MSK F243 controller.

REFERENCES

[1] Rusu C., - "An adaptive position controller of the hybrid stepper motor drive with load torque observer", PCIM'98, Nurnberg, Germany, vol. IM Control, pp.181-189

[2] Imecs M., Kelemen A., Rusu C., Marschalko R.. -"*Vector control of PM hybrid stepping motor with tracking torque*". The 7th International Conference IMACS-TC1'93, 7-8-9 July 1993, Montreal, Canada.

[3] Kelemen A., Imecs M., Marschalko R., Rusu C. -"*Torque and Speed Control Systems for PM Hybrid Stepping Motor*". ICEMA, Adelaida, Australia, vol. 3, pp. 506-512, 13-15 Sept 1993.

[4]. Kellermann H. *et al.* - *Field Oriented Position Control of a Hybrid Stepper Motor.* European Power Electronics EPE 95, Sevilla, Spain., vol.3,pp.908-913.

[5] Kiyonobu Mizutani, Shigero Hayashi, Nobouky Matsui - *Modeling and Control of Hybrid Stepping Motor*. IEEE/IAS Annual Meeting, 1993, pp.289-294.

[6] Nobouky Matsui, Makoto Nakamura, Takashi Kosaka - *Instantaneous Torque Analysis of Hybrid Stepping Motor*. IEEE/IAS Annual Meeting, 1995, pp.701-706.

[7] P.P. Acarnley - *Stepping Motors: A Guide to Modern Theory and Practice*. New-York: Peregrinus, 1992

[8] Törngren M., Henriksson D., Redell O., El-Khoury J., Simon D., Hanzalek Z. and Årzén K.. Co-design of Control Systems and their real-time implementation - A Tool Survey. Technical report. Department of Machine Design. Royal Institute of Technology – KTH, Stockholm Sweden. August 2006.

[9] Real-Time Workshop Embedded Coder User's Guide. The MathWorks, Inc., www.mathworks.com, 2005.

[10] Real-Time Workshop User's Guide. The MathWorks, Inc., www.mathworks.com, 2005.

[11] Real-Time Workshop Embedded Coder Developing Embedded Targets. The MathWorks, Inc., www.mathworks.com, 2005.

[12] Simulink – Writing S-functions. The MathWorks, Inc., www.mathworks.com, 2005.

[13] Real-Time Workshop Target Language Compiler. The MathWorks, Inc., www.mathworks.com, 2005.

[14] Microsoft Component Object Model. Microsoft Corporation, msdn.microsoft.com, 2005.

[15] MATLAB – External Interfaces. The MathWorks, Inc., www.mathworks.com, 2005.