FAST AC ELECTRIC DRIVE DEVELOPMENT PROCESS USING SIMULINK CODE GENERATION POSSIBILITIES

Jakub Vonkomer, Tatiana Radičová, Milan Žalman, Miroslav Suchánek

Institute of Control and Industrial Informatics,

FEI STU Ilkovičova 3, 812 19 Bratislava, Slovak Republic

jakub.vonkomer@stuba.sk

Abstract

Vector control is essential part of any modern AC drive. Therefore development of high performance vector control of induction machine is presented. Vector control algorithms described in this paper have been intensively tested in the Simulink environment with SimPowerSystems toolbox and later on a real 1.1 kW induction machine using dSpace controller board as well. Schemas and screen captures of the models are included. Design of the controllers and flux observer is described. Finally the comparison of results between simulation and real motor is presented.

Index terms – AC drive, vector control, speed control, dSpace, current controllers, flux controller, Simulink, SimPowerSystems

Introduction

The development process of electric drives is usually very time-intensive itself and it also takes a lot of time to transform the model to the final code. Manual code writing of the model introduces a lot of bugs and errors. Some of the bugs even require many hours to be fixed. But nowadays there is no need to do it this way, because of MATLAB code generation possibilities. In our case we used the Real Time Workshop for generating code for dSpace. The generated C code is then compiled and downloaded to the dSpace processor and can be immediately executed. However there are more supported targets for Simulink code generation (TI C2000, Freescale MPC5xx …). This process of design and development is called “model based design”.

Vector Control

Vector control allows high performance control of electromechanical variables like torque, speed, position in both static and transient states. Main task for vector control is direct control of magnetic flux and torque. There are plenty of methods of vector control but we focus on rotor-flux-oriented vector control. The vector control basis can be derived from

$$M_m = k_m \left| \hat{\Psi}_r \times \hat{i}_s \right| \quad \text{Where } k_m = \frac{3}{2} p' \frac{L_m}{L_r} \tag{1}$$

Scalar form of previous equation

$$M_m = k_m \Psi_r i_s \sin \varphi \tag{2}$$

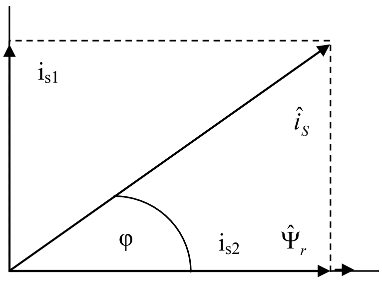

Figure 1 - Vector diagram of stator current and rotor magnetic flux

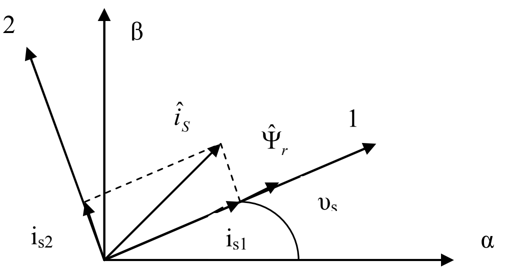

Figure 2 - Vector diagram of the IM state variables

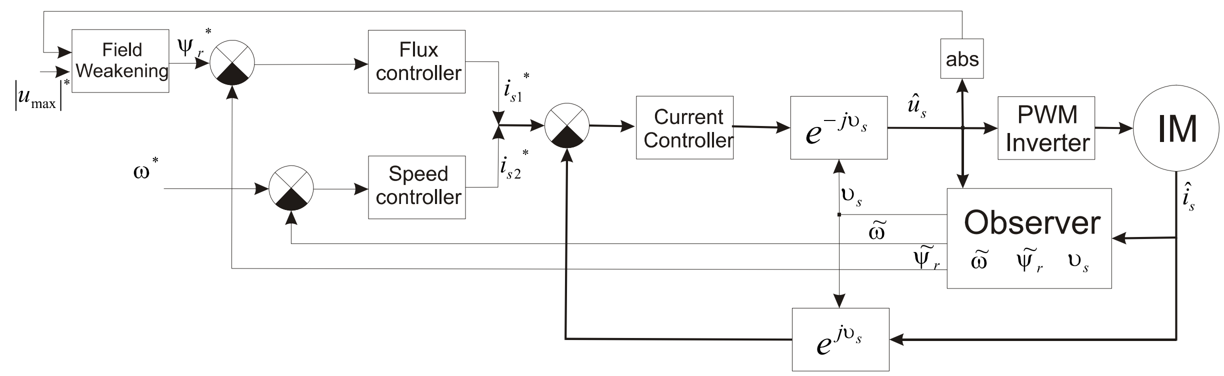

Figure 3 - Simplified flow chart of vector control

All the controllers and observers are derived from the mathematical model of the induction machine:

$$\hat{u}_s = R_s\hat{i}_s + \frac{d\hat{\psi}_s}{dt} + j\omega_k\hat{\psi}_s \tag{3}$$

$$0 = R_r\hat{i}_r + \frac{d\hat{\psi}_r}{dt} + j\omega_{sl}\hat{\psi}_r \tag{4}$$

$$M_m = \frac{3}{2}p'\frac{L_m}{L_r}\Im\{\,\hat{i}_s \cdot \psi_r^*\} \tag{5}$$

$$\omega = p'\omega_m \tag{6}$$

$$M_m - M_l = J\frac{d\omega}{dt} \tag{7}$$

Table 1 - Description of symbols and acronyms used in this paper:

ω	- angular speed of motor	T_1	- current model time constant ($T_1=\sigma L_s/R_1$)
ω_{sl}	- slip speed	T_r	- rotor time constant ($T_r=L_r/R_r$)
p’	- number of pole pairs	J	- moment of inertia
σL_s	- stator leakage inductance	M_m	- motor torque
L_s	- stator winding inductance	M_l	- load torque
L_r	- rotor winding inductance	$\hat{i}_s$	- stator current vector
L_m	- mutual inductance	$\hat{u}_s$	- stator voltage vector
R_s	- stator resistance	$\hat{\psi}_r$	- rotor flux vector
R_r	- rotor resistance	$\hat{\psi}_s$	- stator flux vector
R_1	- combined resistance ($R_1=R_s+L_m^2/L_r^{2*}R_r$)		

The structure type of all controllers is PI and they are designed by using pole-placement method. All systems are described by first order dynamic equations.

Current controllers

Design of current controllers is derived from the following transfer function:

$$G_S(s) = \frac{i_{S1}(s)}{u_{S1}(s)} = \frac{i_{S2}(s)}{u_{S2}(s)} = \frac{1/R_1}{T_1 s + 1} \tag{8}$$

Final current controller parameters are achieved using the following equations:
(ω_o is desired dynamics, b desired damping)

$$K_p = \left(2b\omega_o - \frac{1}{T_1}\right) R_1 \cdot T_1 \qquad T_i = \frac{K_p}{R_1 T_1 \omega_o^2} \tag{9}$$

Flux controller

Design of flux controller is derived from transfer function:

$$G_S(s) = \frac{\Psi_r(s)}{i_{S1}(s)} = \frac{L_m}{T_r s + 1} \tag{10}$$

Final current controller parameters

$$K_p = \left(2b\omega_o - \frac{1}{T_r}\right) \frac{T_r}{L_m} \qquad T_i = \frac{K_p L_m}{T_r \omega_o^2} \tag{11}$$

Speed controller

Speed controller is designed by knowing the mechanical parameters

Transfer function:

$$G_S(s) = \frac{\omega(s)}{i_{S1}(s)} = \frac{3}{2} \cdot p' \cdot \frac{L_m}{L_r} \psi_r \cdot \frac{1}{Js + B} \tag{12}$$

Final equations:

$$K_p = \frac{(2b\omega_o J - B)}{\frac{3}{2} \cdot p' \cdot \frac{L_m}{L_r} \psi_r} \qquad T_i = \frac{K_p \cdot \frac{3}{2} \cdot p' \cdot \frac{L_m}{L_r} \psi_r}{J \cdot \omega_o^2} \tag{13}$$

Flux weakening controller

Flux weakening is necessary for achieving speeds higher than rated speed of the motor. The design of flux weakening controller is too complicated for this paper. But in general it is voltage controller which is activated when the voltage is close or equal to saturation.

Flux observer

Flux observer is the heart of vector control, because of the control algorithm is oriented in rotor magnetic flux reference frame. That is one of the reasons vector control is often called “Field Oriented Control”.

Following equations and figure 4 describes presented flux observer. ω is mechanical angular speed, υ is mechanical rotor position (time integral of ω).

$$s\hat{\psi}_r = -\frac{1}{T_r}\hat{\psi}_r + j\omega\hat{\psi}_r + \frac{L_m}{T_r} i_s \tag{14}$$

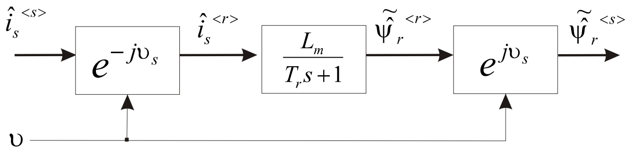

Figure 4 – flux observer

Simulation model

The aim of our project was to develop speed vector control and sensorless vector control for the Induction Machine. Speed sensorless vector control means feedback speed control without any mechanical speed sensor. Speed is observed from motor voltages and motor currents only.

Our primary platform for developing and testing purposes was Simulink with SimPowerSystems toolbox. We used model of induction machine, three phase source and many other power electronics blocks for close-to-reality simulation of the drive.

The basic block is "*VECTOR CONTROL*" which contains all the required controllers, observers, state information, error flags, etc. This block is later copied to the dSpace model. Figure 5 shows the main window of the simulation SimPowerSystems model.

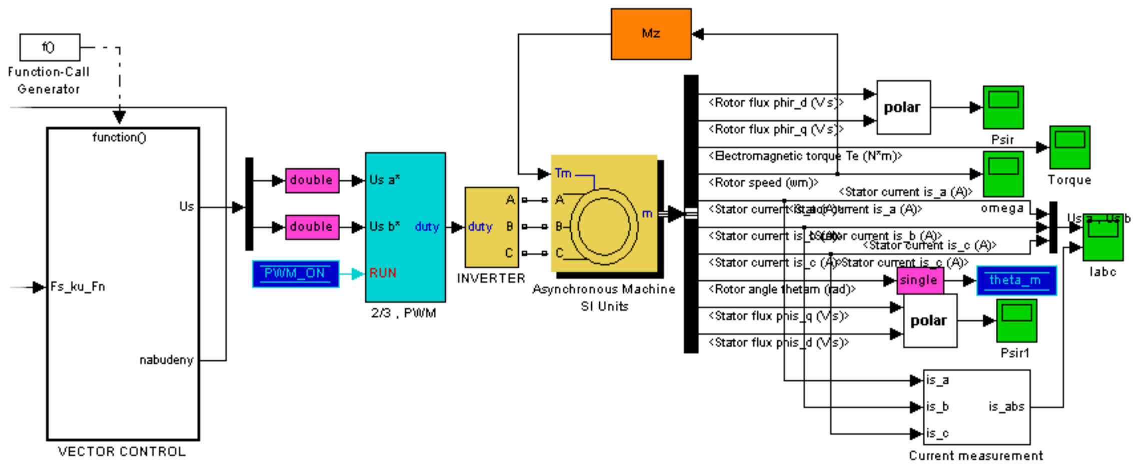

Figure 5 - Simulation model

Figures 6, 7 and 8 show the inner connections of selected blocks. The model is purely discrete with sample time of 200 µs. All the blocks are part of Embedded MATLAB subset, so there is not any problem with code generation. Sample time for simulating the electronics has been chosen as 10 µs. In figure 8 the flux current controller is presented.

We prefer to use constant blocks for parameters instead of gain blocks. Because constant blocks can be more easily tuneable, they might be configured as inputs in the ControlDesk applications and its value can be changed on the fly. The same applies for code generation for the other platforms. Variables inside the constant block may be configured as exported variables as part of global structure or as global variable as well. These variables may be changed by the custom code. So it is possible to change the behaviour of the model on the fly. The entire model is based on IEEE 754 single precision floating point numbers so this model is faster in execution and can be very easily ported to any target with floating point numbers support.

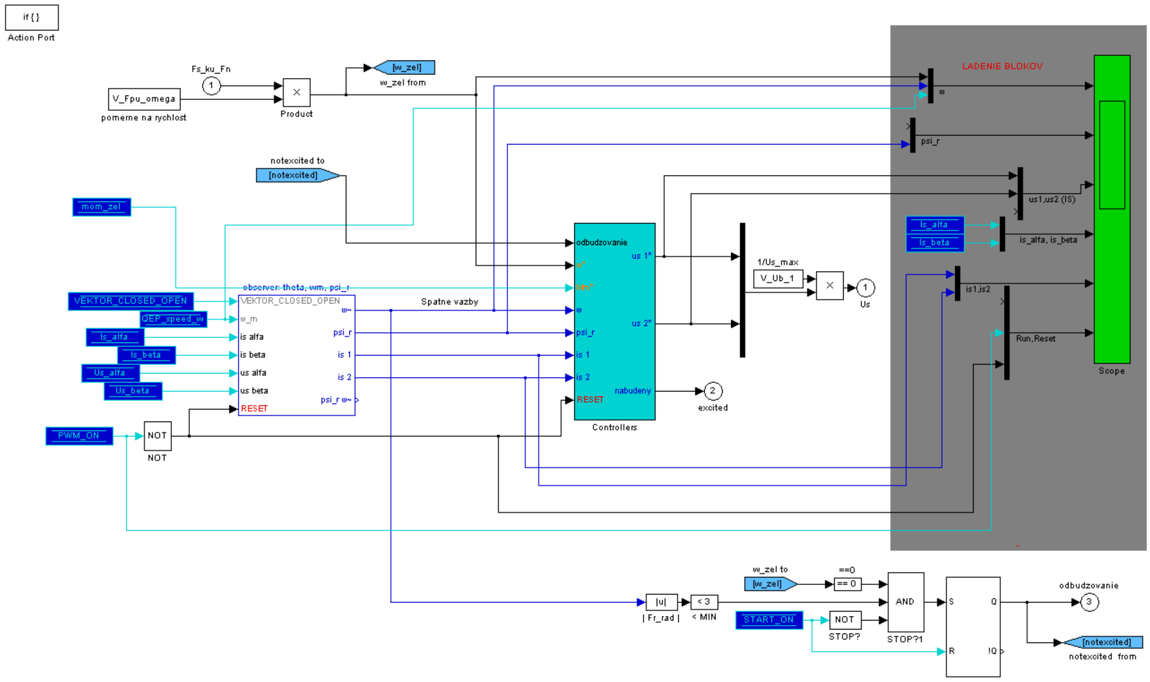

Figure 6 - Inner connection of the VECTOR CONTROL block

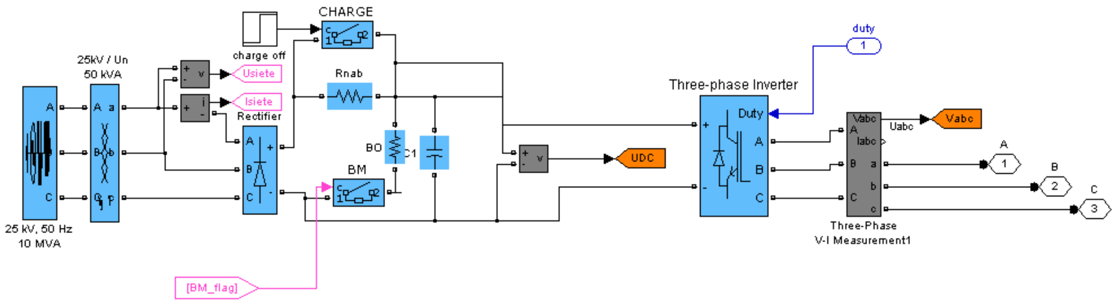

Figure 7 - Inner connection of the inverter block

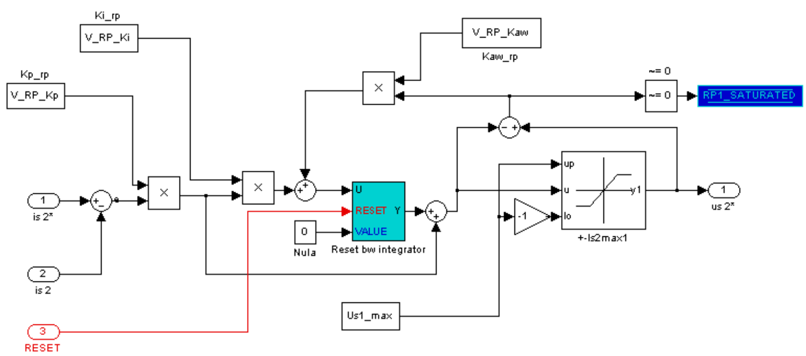

Figure 8 - Inner connection of the current controller block

dSpace model

The dSpace board allows real-time control of different systems. Its main advantage is integration into the MATLAB/Simulink environment. So instead of complicated time-intensive rewriting model to the C code, we developed the program in Simulink and then we just built and executed the finished model. Model is executed and controlled via the ControlDesk application. It is also very easy to export any graphs to MATLAB.

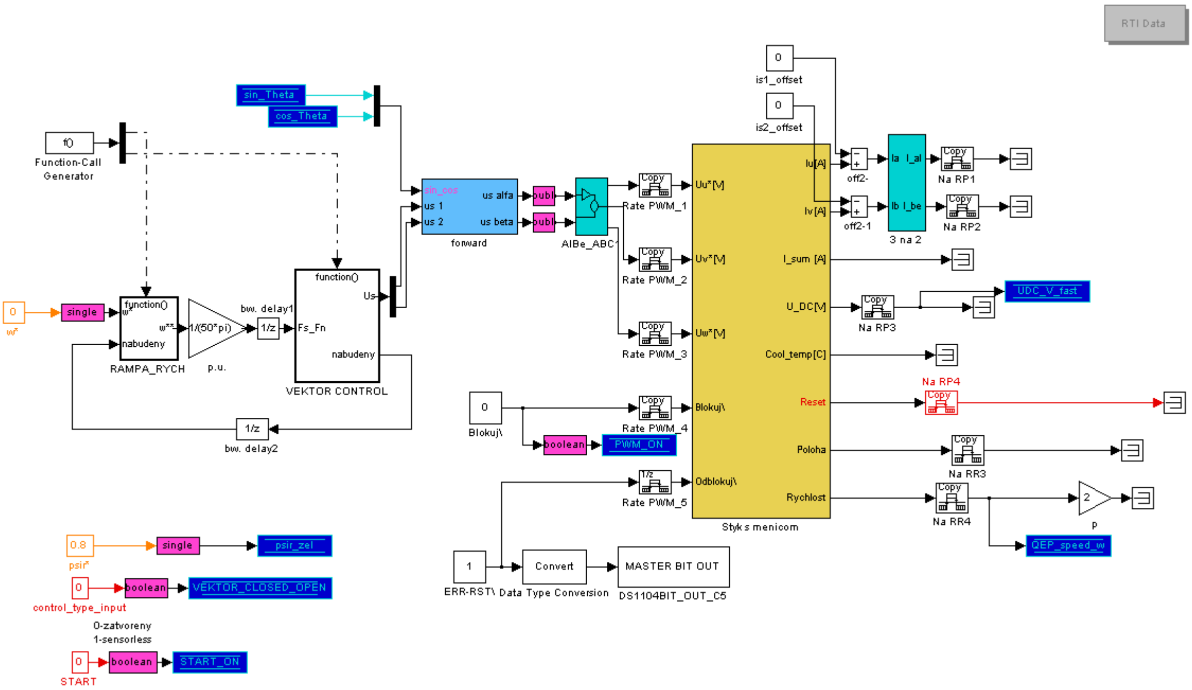

Figure 9 – Screen capture of proposed dSpace model

DSpace models and simulation models are pretty similar as can be seen in figure 9. Instead of induction machine and inverter blocks, the block of communication with inverter is used. Inputs represent reference voltages. The block of communication with inverter contains the compensation of DC voltage fluctuations as well and the computed duty cycles are inputs for the dSpace PWM block.

Figure 10 shows the basic schema of connection dSpace controller board to frequency converter and the induction machine as well.

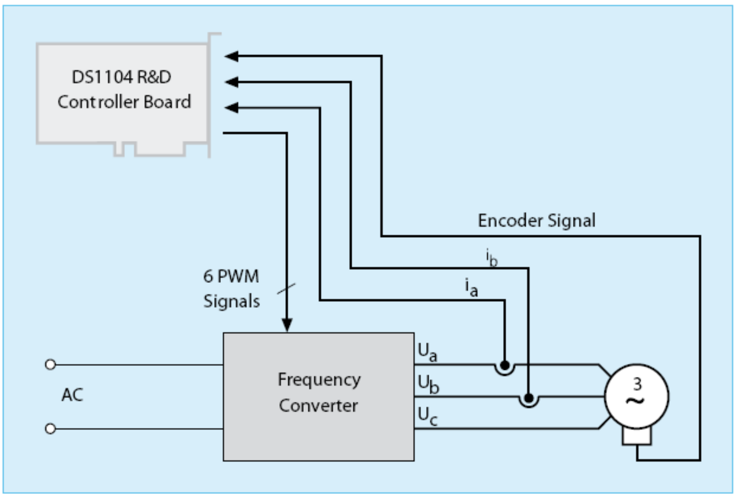

Figure 10 – Schema of connecting dSpace equipped PC with frequency inverter and induction motor

Comparison

Figures 11-14 show the comparison of simulation and real behaviour of modelled system. Vector control at low speed is demonstrated. Small differences and noise in dSpace figures are caused by non-calibrated current sensors, but indeed we can say that we reached comparative results within the toleration.

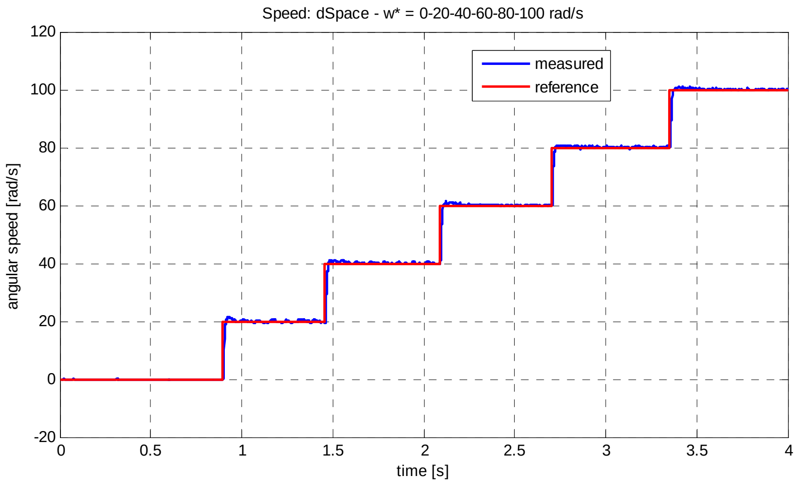

Figure 11 – dSpace experiment: steps of reference speed up to 100 rad/s

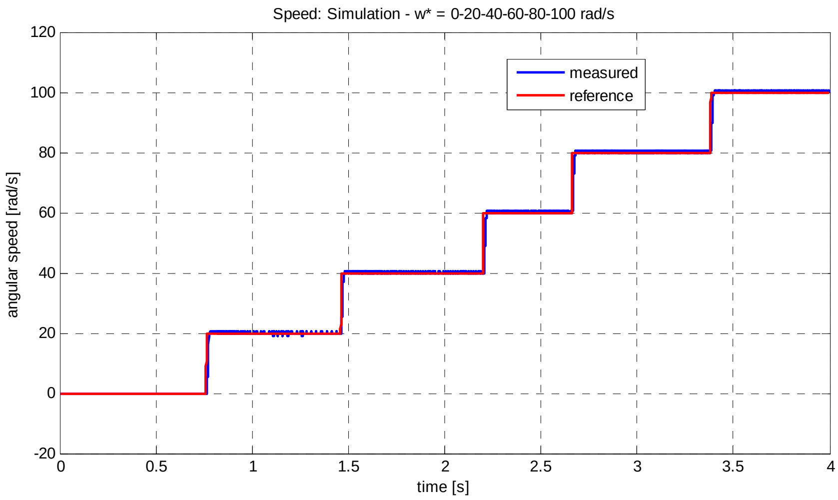

Figure 12 – Simulation: steps of reference speed up to 100 rad/s

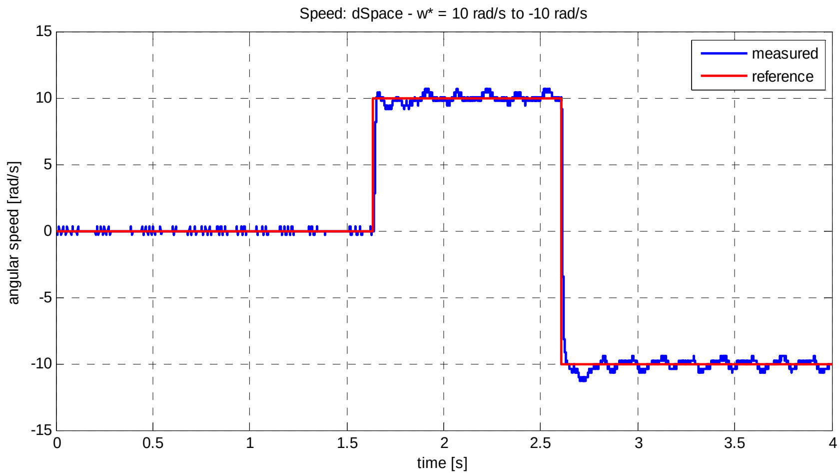

Figure 13 – dSpace experiment: performance of low speed and speed reversal

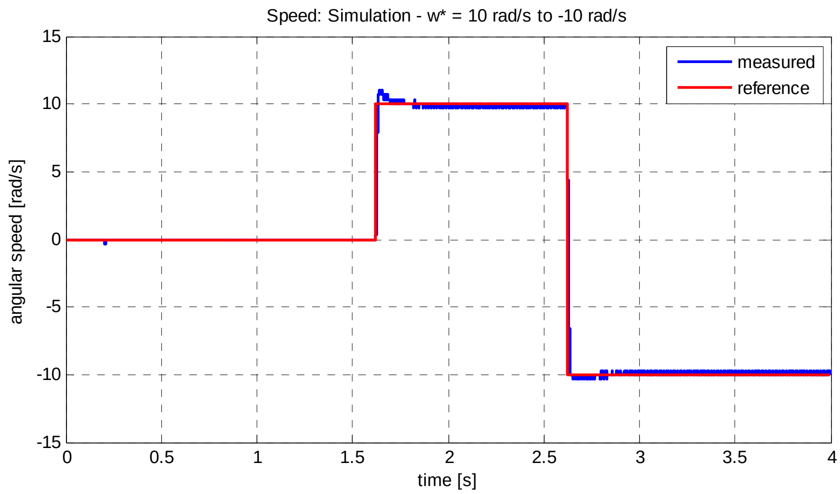

Figure 14 – Simulation: performance of low speed and speed reversal

Conclusion

In this paper we had focus on induction machine vector control development. Since the beginning we tried to keep the model simple, robust and fully discrete as well, because we planned to share the basic model with other platforms – dSpace in our case. The basic testing during the creation and development of the model was by simulation only, later we tried it also on a dSpace platform where real capabilities were demonstrated.

In the future we may use another platform and it can be used for educational purposes and also for testing some advanced control structures.

Acknowledgement

This work has received support from the Slovak Research and Development Agency. Project reference number is APVV-0530-07.

References

[1] Žalman M.: Akčné členy (Slovak language), STU Bratislava 2003

[2] Milan Žalman – lectures for Inteligent Servosystems, 2007/2008. http://servo.urpi.fei.stuba.sk/index.php?option=com_content&task=view&id=15&Itemid=26

[3] Doki, S.; Kinpara, Y.; Okuma, S.; Sangwongwanich, S., "Unified interpretation of indirect and direct vector control [of electric machines]," *Power Conversion Conference, 1993. Yokohama 1993., Conference Record of the* , vol., no., pp.297-302, 19-21 Apr 1993 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=264167&isnumber=6639

[4] DS1104 R&D Controller Board, dSpace Inc. 2009 Catalogue, URL:http://www.dspaceinc.com/ww/en/inc/home/medien/papers/download_page.cfm?FileID=778

[5] The MathWorks, SimPowerSystems Documentation, 2008 URL: http://www.mathworks.com/access/helpdesk/help/toolbox/physmod/powersys/index.html?/access/helpdesk/help/toolbox/physmod/powersys/

[6] The MathWorks, SimPowerSystems Demos, 2008

Ing. Jakub Vonkomer, Email:jakub.vonkomer@stuba.sk

Ing. Tatiana Radičová, Email: tatiana.radicova@stuba.sk

prof. Ing. Milan Žalman PhD, Email: milan.zalman@stuba.sk

Ing. Miroslav Suchánek, Email: miroslav.suchanek@stuba.sk

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava,