Programming of the DSP2 board with the Matlab/Simulink

Darko Hercog, Milan Čurkovič, Gregor Edelbaher, Evgen Urlep
University of Maribor
Faculty of Electrical Engineering and Computer Science
Smetanova 17, SI-2000 Maribor, Slovenia
darko.hercog@uni-mb.si, milan.curkovic@uni-mb.si, gregor.edelbaher@uni-mb.si, evgen.urlep@uni.mb.si

Abstract –the **DSP2 board, based on the digital signal processor, has been developed at the Institute of Robotic, FERI, University of Maribor. For mentioned board a set of Simulink blocks, so called the DSP2 Library for Simulink, was created under the Real-Time Workshop. These blocks enable easy graphical programming of the different control algorithms under the Matlab/Simulink. At the end, with the Real-Time Workshop and the Code Composer for the C3x4x digital signal processors, the binary executable code can be generated from the Simulink model and downloaded to the DSP2 board, where it is executed in real time. The DSP2 Library for Simulink in combination with the DSP Terminal, which was also developed at the at same institute, enables easy online monitoring of the DSP2 variables in the text or graphical mode and the parameter tuning, meanwhile the generated code is executed on the DSP2 board. With the DSP2 Library for Simulink, the development time of the different control algorithms that are executed on the DSP2 board is reduced.**

1 Introduction

DSP2 board, based on Texas Instruments (TI) floating point digital signal processor (DSP) TMS320C32 has been developed at the Institute of Robotic, Faculty of Electrical Engineering and Computer Science, Maribor (FERI). Until recently, the programming of the DSP2 board was possible only in the C and the assembler programming languages. For that reason a lot of work was needed for the realization of the control algorithms on the DSP2 board. In the text programming languages, also coding errors are very often present, which additionally extend development time of the control algorithms. On the other hand, a majority of the control algorithms are previously simulated and tested in the Matlab/Simulink [11], [9]. Matlab is well-known simulation program that is widely in use around the world in the researching area of the dynamic systems. Real-Time Workshop (RTW) is the Simulink add-on software that enables automatic C or ADA code generation from the Simulink model. To avoid here stated problems using the C code programming, we decided to take advantage of the Matlab, Simulink and the RTW, and develop some Simulink blocks, that will enable block programming of the DSP2 board with the Simulink. As a result of our work, the DSP2 library for Simulink [3] was created.
Nowadays several companies exist on the market ([12], [13], [14]…), that offer the hardware for the motor control, based on the digital signal processors. Programming of this hardware can be done using the Simulink. A majority of this hardware are PC boards that are plugged into PCI or ISA slot on the PC. Although, that these types of boards have some advantages (fast communication channel, they don't need additional power supply...), they also have two weaknesses. They can't operate without the PC and they are quite expensive.

The present article is divided into 6 sections. In the section 2 a few more words about the project are stated while in the section 3 some key features of the DSP2 board are described. Section 4 deals with the code generation process from the Simulink model, while in the section 5 the result of our work, so called DSP2 Library for Simulink is presented. Section 6 summarizes some preferences and weaknesses of the Matlab/Simulink. In this section also future work is mentioned.

2 Project overview

In the present time block programming of the microcontrollers and DSP's is more and more important and it plays a key role in the time to market of the new products. Block programming is much closer to the engineers and the scientists than other types of programming. Block programming enables that they can force their energy to the system design and not into the code development process, like it was in the practice in the previous era. The code generation process must be quick and transparent to the system designer. If so, they can quickly test and validate designed algorithms on the intended target (microcontrollers, DSP…).
Several companies exist on the market that offers software solution for block programming of the microcontrollers and DSP's. One of such is a company The MathWorks Inc., producer of the well known simulation program Matlab/Simulink. The Matlab/Simulink is an interactive tool for modeling, visualization, simulation and analysis of the dynamic systems. Simulink is Matlab add-on software which enables block programming and simulation of different signal processing algorithms. With the Real-Time Workshop (RTW) [5], the Matlab and Simulink has been evolved from the simulation program into the integrated development environment. RTW enables automatic C or ADA code generation from the Simulink model.
In the described project we wish to take advantage of all the mentioned benefits that Matlab/Simulink offers. On the Fig. 1, our desired aim is graphically presented. We wish to design and simulate the control algorithms for the motor control in the Simulink and after successful simulation convert the Simulink model with the RTW into the C code.

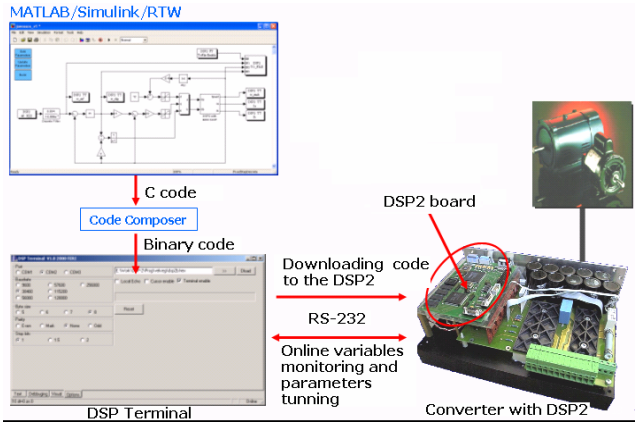

Fig. 1: Programming of the DSP2 board with the Simulink

Because the generated code will run on the DSP2 board, the TI Code Composer for C3x4x digital signal processors is used for generation of the binary executable code, from previously generated C code. After that the DSP Terminal [2], which was also created at the Institute of Robotics, FERI Maribor, is used for downloading the executable code to the DSP2 board, online monitoring of the selected DSP2 variables and the parameter tuning of the scalar Simulink blocks parameters. The DSP2 board is connected to the PC using the RS-232 serial connection. Mentioned board can be placed on the motor converter, like it is shown on the Fig. 1, and so can be applicable for torque, speed or position control of the DC or AC motors or it can operates as a stand alone unit and in such a manner used for general purpose applications.

To achieve described aim a set of the Simulink blocks have been developed. Short overview of these blocks is presented in the chapter 5.

3 DSP2 board

A few years ago was very hard to find the motor controller board based on the digital signal processor with the desired peripheral, performance, floating point arithmetic and the suitable price on the market, therefore on the Institute of Robotic, FERI, Maribor, decision was made to develop the custom DSP board. So called DSP2 board [1] has been developed, based on the Texas Instruments TMS320C32 floating point processor and the Xilinx FPGA XCS40PQ240 of the Spartan family. The board is mainly in use for the torque, speed and position control of the AC motors. On the Fig. 2 the photography of the DSP2 board is presented.

In the rest of this section some key features of the DSP2 board are stated. More information about the DSP2 board can be found in the literature [1].

DSP2 board has the following features:

- DSP TI TMS320C32 - 60 MHz;
- FLASH 256Kx8 - 70ns;
- SRAM 128Kx32;
- Xilinx FPGA XCS40PQ240
- 4 x 12 bit simultaneous A/D with serial output:
  - one channel with unipolar input range from 0 to 4.095V or 0 to 40,95mA with 100 Ω shunt resistor;
  - two input channels with bipolar input range from -2.048 to 2.047V or -20,48mA to 20.47mA with 100 Ω shunt resistor;
  - one input channel with the input multiplexer to select one of eight voltage input signals;
  - conversion and transfer to register in FPGA takes 2,6µs for all four A/D channels;
- two channels 12bit D/A converter with serial input and unipolar output 0 to 4.095 V;
- RS232 full duplex interface with fixed Baud Rate (57600kBd 8bits ,1stop , no parity);
- RS485 interface;
- RS422 receiver for incremental encoder. Speed measurement with improved MT method [4];
- three logic inputs and one logic output – all optically isolated (12V passive);
- three phase synchronous pulse width modulator (PWM);
  - symmetrical output pulses;
  - 66 ns time resolution;
- DSP MPSD interface for XDS510 emulator;
- interlock between bottom and top IGBT activation and dead time
- etc.

4 System functions and code generation process

Simulink open architecture enables inclusion of the custom block with the custom algorithm to the Simulink model. This can be done with the system function or so called S-function that can be written in C programming language (C-MEX S-functions). S-function defines number of the inputs, number of the outputs, algorithm for calculation of the block output values, sample time, etc for the custom Simulink block.

Fig. 2: Photography of the DSP2 board

When the custom S-function algorithm is written, dynamic library must be created from the S-function source code. After that, it can be included in the Simulink with the S-function block [8]. During the simulation, in each simulation step Simulink calls the functions placed within created dynamic library and based on the block parameters, states and the inputs it calculates block new states and the block outputs. In our case, for each block in the DSP2 Library for Simulink (Fig. 5) (DSP2 library is described in section 5), an S-function has been written. More about S-functions can be found in [8].

Real-Time Workshop (RTW) [5] enables automatic ANSI-C or ADA code generation from the Simulink model. In general the RTW generates two types of the C codes:

- generic C code
- embedded C code

Embedded C code is much more optimized in performance and the space usage and consecutively it is more useful for the embedded targets, such as the digital signal processors (DSP) and the micro controllers. In spite of some restrictions in the embedded C code (only the discrete Simulink blocks can be used in the Simulink model) and some additional work that must be done to develop the custom embedded target (for each custom S-function corresponding .tlc file must be written [7]), we decided for this type of C code generation.

On the Fig. 3 C code generation process from the Simulink model is shown. Real-Time Workshop converts the Simulink model stored in the .mdl file into .rtw file. This file contains all the necessary information's about the Simulink model that are needed during code generation process. RTW also generates makefile (model.mk) from the system template makefile (system.tmf). After that, the Target Language Compiler (TLC) [7] is invoked into the code generation process and it converts Simulink model to the C code. When the TLC concludes with the C code generation process, *make* program based on previously generated C files, makefile and using the appropriate compiler generates executable code. Several makefile programs exist on the market. In our case the *gmake* program from the Free Software Foundation [16] was used, because it is free and it is also enclosed with the Matlab.

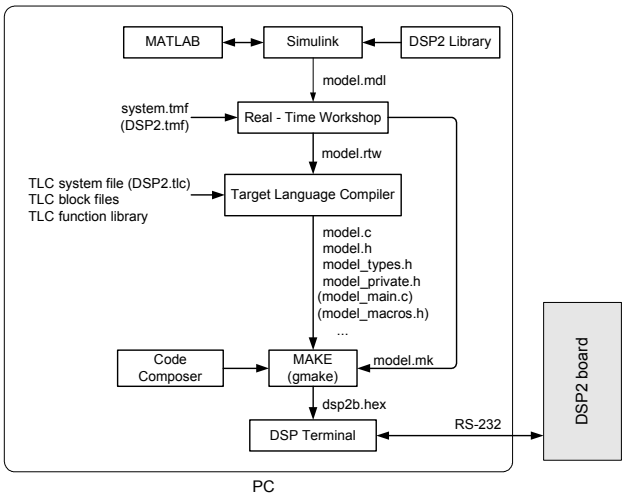

Fig. 3: C code generation process from the Simulink model

To support DSP2 board with the C code generated from the Simulink model, besides the DSP2 Simulink blocks, the system target file (dsp2.tlc) and the template make file (dsp2.tmf) must have been developed.

5 DSP2 Library for Simulink

On the Fig. 4, the result of our work, the DSP2 Library for Simulink is presented. It is composed out of a few subsystems, where the most important subsystem is the DSP2 I/O blocks (Fig. 5). This subsystem contains the input/output blocks for the DSP2 board that enable graphical programming of the DSP2 board with the Simulink. The DSP2 I/O subsystem contains the following blocks:

- Analogue input/output
- Digital input/output
- From address/To address
- From Terminal/To Terminal
- PWM output
- Encoder
- To file
- PRBS (pseudo random binary signal)
- Pulse generator
- Transformations:
  - from a-b to d-q system of coordinates
  - from d-q to a-b system of coordinates
  - rad/s to RPM
  - RPM to rad/s

The meaning of each block of the DSP2 Library for Simulink is in great deal explained in literature [3]. With these blocks and Simulink built-in blocks almost any type of the controller or signal processing algorithm can be designed and simulated under the Simulink. Note, that all Simulink DSP2 blocks are grounded in the simulation because Simulink can't access to the DSP2 I/O peripheral meanwhile the simulation is executing. But, when the executable code is generated from the Simulink model and downloaded to the DSP2 board (code generation and downloading process takes approximately 20s), DSP processor actually performs reading from and writing to the IO peripheral of the DSP2 board, depends on the DSP2 blocks used in the Simulink model.

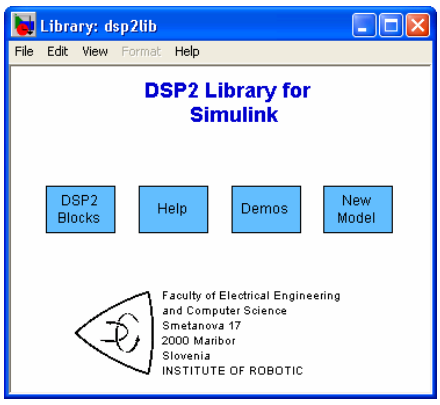

Fig. 4: Screen shot of the DSP2 library for Simulink

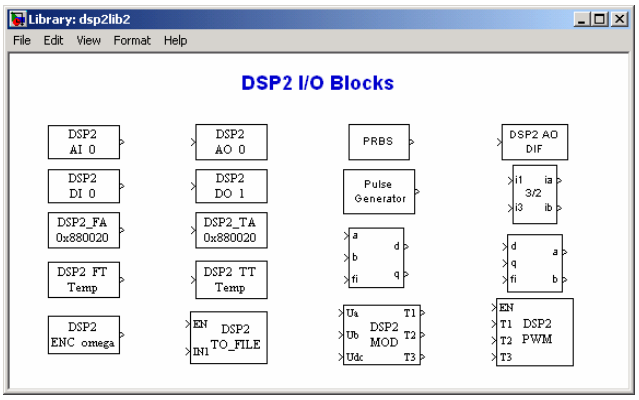

Fig. 5: Screen shot of the DSP2 I/O blocks

DSP2 Library for Simulink works in a combination with the DSP Terminal [2] that runs on the PC. DSP Terminal [2] is used for binary code download to the DSP2 board, online changing and monitoring of selected DSP2 variables and parameters tuning of the scalar Simulink block parameters. Graphical user interface (GUI) in the DSP Terminal (Fig. 6) is automatically generated when the executable code is downloaded to the DSP2 board. GUI in the DSP Terminal depends on two DSP2 blocks used in the Simulink model. These blocks are so called "From Terminal" and "To Terminal". Block "From Terminal" is a Simulink source block and it enables changing of the variable within the DSP Terminal meanwhile the generated code is executing on the DSP2 board. Block named "To Terminal" is a Simulink sink block and is used only for the variable monitoring. For each of these blocks an option box, text label with variable name, numeric box for the variable value presentation and the unit label appear on the upper side of the DSP Terminal. When the option box by the variable name is selected, the variable also appears in the graphical mode in the graph placed on the bottom side of the DSP Terminal.
In addition, all or only selected scalar block parameters of the Simulink blocks appears in the "Parameter Inspector" window of the DSP Terminal. These parameters are changeable in the online mode, therefore fine parameter tuning of the controller can be made meanwhile the generated code is executing on the DSP2 board.

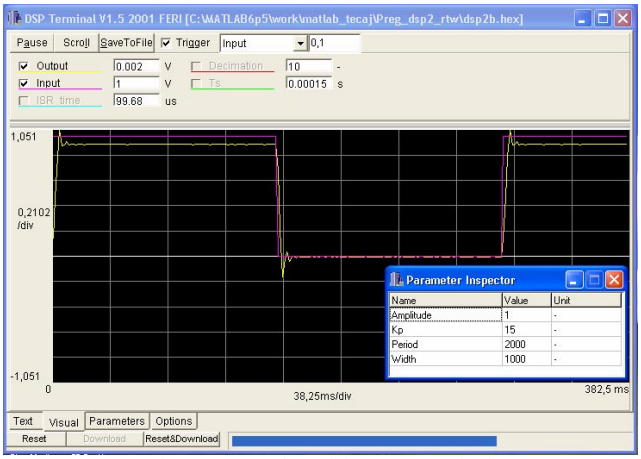

Fig. 6: Screen shot of the DSP Terminal

Because we decided for production C code generation, one restriction must be considered in the development phase of the signal processing algorithm. Simulink model can contain only the Simulink discrete blocks or blocks that are supported with ERT format [6].
In the paragraphs beneath some Simulink additional toolboxes that are supported with the ERT format are stated:

- DSP Blockset
- StateFlow
- Fixed-Point Blockset
- Fuzz Logic Blockset
- Neural Networks Toolbox
- etc.

A majority of motor controllers are based on fixed point processors. These processors like TI TMS320LF2407A contain all the peripheral that is needed for 3-phase motor control on a chip. Fixed point processors are much cheaper than floating point processors and are for this reason more appropriate for massive production of the motor controllers.
Although the DSP2 board is based on the floating point processor, also fixed point algorithms can be tested on this board. The MathWorks Inc. developed a powerful Simulink toolbox for fixed point arithmetic named Fixed-Point Blockset. This blockset contains a lot of fixed point blocks that allow design of the control and signal processing algorithms that will be implemented using fixed-point arithmetic. By using the Fixed Point Blockset and the DSP2 Library for Simulink the development time of the fixed point algorithm can be drastically reduced. Designed fixed point control algorithm can be quickly tested and validated on the DSP2 board.

Also the setup program for the DSP2 Library for Simulink has been developed. This program copies all the necessary files to the PC and adds new folders to the Matlab search path. Prerequisite for this installation is previously installed the following software:

- Matlab 6.5 with RTW and Embedded Real Time Coder
- TI Code Composer for C3x4x processors

A few of screen capture flash movies have also been created where the usage of the DSP2 Library for Simulink in combination with the DSP2 board is shown. These movies of total length approximately 30min are very appropriate for new users of the DSP2 board. Mentioned movies are in Slovene language and are accessible on the DSP2 web page [15].

6 Conclusion

In this article the DSP2 board and the DSP2 Library for Simulink was presented.
Real-Time Workshop is powerful and applicable tool that enables automatic C code generation from the Simulink model. Generated code is well optimized and it's comparable with the hand written code. Superior property of the Simulink and the RTW is its open architecture that

enables adding custom blocks to the Simulink and custom target development, like it was presented in this article for the DSP2 board.

DSP2 library for Simulink in combination with the DSP2 board is very useful in the teaching and researching area of dynamic systems. With this blockset and Simulink built-in blocks almost any type of control algorithms can be implemented in a very fast and simple way on the DSP2 board. Although the DSP2 board is mainly in use in the researching area of the motor control, it is also very suitable for the pedagogic purposes, because deep knowledge of the programming of the DSP processors is not needed. In the pedagogic area is very useful that the students test their designed control algorithms not only by using the simulations, but also on the real system. In such a way, the students are also acquainted with some limitations, nonlinearity, saturation... that appear in the real world and are usually not considered in the simulation.

The main weakness of the DSP2 board is a serial connection with the PC. In the future we'll try to replace an existing RS-232 serial communication with the TCP/IP or in the lasted time more and more applicable universal serial bus (USB). Simulink also enables so called "External mode" of operation. In this mode, Real-Time Workshop establishes a communication link between a model running in the Simulink and code executing on a target system [5]. Whenever the parameters in the Simulink block diagram are changed, Simulink automatically downloads them to the executing target program. In the future we'll try to accommodate this superior feature of External mode to our DSP2 board.

References

[1] FERI Maribor: *DSP2 User's Manual,* version t3, march 2001, Institute of Robotic, FERI Maribor.

[2] FERI Maribor: *DSP Terminal User's Manual,* version t3, mach 2001, Institute of Robotic, FERI Maribor.

[3] FERI Maribor: *DSP2 Library for Simulink User's Manual*, September 2003, Institute of Robotic, FERI Maribor.

[4] ČURKOVIČ, Milan, JEZERNIK, Karel, CAFUTA, Peter*:* "Digitalno merjenje hitrosti", V: *Zbornik radova JUREMA, svezak 3*. Zagreb, 1986: [s. n.], pp. 21-24.

[5] The MathWorks, Inc.: *Real-Time Workshop User's guide (rtw_ug.pdf),* version 3, January 1999, www.mathworks.com.

[6] The MathWorks, Inc.: *Real Time Workshop Embedded Coder (ecodder_ug.pdf)*, version 3, July 2002, www.mathworks.com.

[7] The MathWorks, Inc.: *Target Language Compiler Reference Guide (tlc_ref.pdf),* version 1.2, January 1999, www.mathworks.com.

[8] The MathWorks, Inc.: *Writing S-Functions (sfunctions.pdf),* version 3, October 1998, www.mathworks.com.

[9] The MathWorks, Inc.: *Using Simulink (sl_using.pdf),* version 3, October 1999, www.mathworks.com.

[10] Texas Instrument, Inc.: *Code Composer User's Guide (SPRU296A.pdf),* October 1999, www.ti.com.

[11] The MathWorks, Inc, www.mathworks.com

[12] dSPACE, Inc, www.dspaceinc.com

[13] Wind River System, www.windriver.com

[14] Humusoft s.r.o., www.humusotf.cz

[15] DSP2 web page, www.ro.feri.uni-mb.si/projekti/dsp2

[16] Free Software Foundation, www.gnu.org