Implementation of the emotional controller Using DSK TMSC320 Digital signal processor

S. Golabian[1], C. Lucas[2], M. R. Jamali[3], M. Afrasiabi[4]

[1]Electrical Engineering Department of Islamic Azad University, Central Tehran Branch

[2,3]Control and Intelligent Processing Centre of Excellence Department of Electrical and Computer Engineering University of Tehran

e-mail: [1]somayegolabian@gmail.com, [2]lucas@ipm.ir , [3]m.jamali@ece.ut.ac.ir, [4]m.afrasiabi@modares.ac.ir

ABSTRACT

In this paper an intelligent controller is implemented and verified on a digital signal processor (DSP) TMS320C7613 for controlling an embedded target application. The intelligent controller is brain emotional learning based intelligent controller (BELBIC), which is based on emotional learning process in Amygdala-Orbitofrontal system of the mammalians brain. The purpose of this paper is converting abstraction level, such as SIMULINK, to an executable code that runs in real time on the DSP hardware. The needed code for DSP is generated in code composer environment. At the experimental part, result of implementation phase verifies that implemented controller behaviors are similar to SIMULINK model.

Keywords

BELBIC controller, real-time implementation**,** Emotional learning, Intelligent Control, digital signal processor

1. INTRODUCTION

Computation complexity is a major problem of intelligent systems in their industrial aspect [1]. These days, implementation of intelligent systems on DSPs, FPGA, ASIC and other hardware platforms is possible and cost effective through developments in VLSI technology [1-4]. As an example, DSPs (digital signal processor) are used widely in communications, controls, and speech processing and alike applications [5]. As a matter of fact, DSP reduces noise susceptibility, chip count, development time, cost and power consumption. In addition it makes it easy to change, correct and to update applications. [6]

As the most popular approaches, intelligent control techniques such as various artificial intelligent algorithms and machine learning techniques can be named that manipulate and implement heuristic knowledge [1]. Artificial neural networks, fuzzy control, genetic algorithms and reinforcement learning control are examples of these control techniques among which BELBIC is an intelligent controller that has been applied successfully in some real and simulated control systems [7], [8], [9], however, for utilizing this controller for real-time applications it must be implemented in an embedded hardware platform.

In this paper abstraction level of this emotional controller model, such as SIMULINK model, is converted to an executable code that runs in real time on the DSP hardware, toward using this controller for an industrial real-time application. Using MATALB/SIMULINK and Embedded Target for TIC6000 DSK board has been benefited from implement this algorithm. Via The software tools, it has made it possible to save a lot of time in implementation, in addition to generation of the necessary C and Assembler code.

The DSK board is introduced in section 2. In section 3 the structure of the utilized emotional controller is introduced. In Section 4 the SIMULINK model and the real-time implementation of the proposed controller is described.

Implementation results of simulation model and the verification of implementation on DSP is presented in section 5, and finally section 6 summarizes the conclusions.

2. DSK TMS320C6713

The DSK TMS320C6713 is a development board from Texas Instruments. It is composed of the C6713 floating-point digital signal processor (DSP) and a 32 bit stereo codec (AIC23) to handle input and output signals (Fig. 1). The AIC23 codec uses a sigma-delta technology that provides Analogue to Digital conversion (ADC) and Digital to Analogue conversion (DAC) [5]. To develop this application the board must be connected to a PC host, and it uses a 225-MHz system clock, so the variable sampling rates can be set from 8 to 96 kHz.

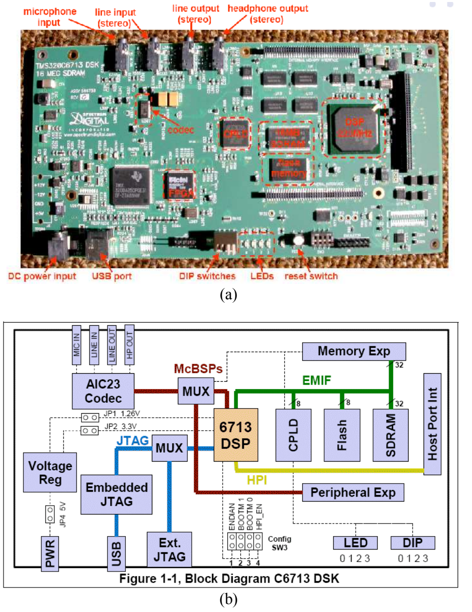

Figure1. TMS320C6723 based DSK board: a) board b) diagram [11]

There are two 80-pin connectors which play the role of interface for external peripheral and external memories; however the DSK itself consists of 16MB (megabytes) of synchronous dynamic random access memory (SDRAM) and 256kB (kilobytes) of flash memory. [10] This board functions on the basis of VLIW architecture, which is significantly appropriate for numerically intensive algorithms. The structure of the internal program memory is in a way that a sum of 8 instructions can be fetched every cycle. Features of the C6713 include 264 KB of internal memory (8kB as L1P and L1D Cache and 256KB as L2 memory shared between program and data space), eight functional or execution units composed of six arithmetic-logic units (ALUs) and two multiplier units, a 32-bit address bus to address 4 GB (gigabytes), and two sets of 32-bit general-purpose registers. The C6713 is capable of both fixed- and floating point processing [10].

3. Emotional controller model

In this section, the structure of BELBIC is introduced. A brief structure of this controller is shown in Figure (2) [9].
BELBIC is a simple composition of Amygdala and Orbitofrontal cortex in the brain.

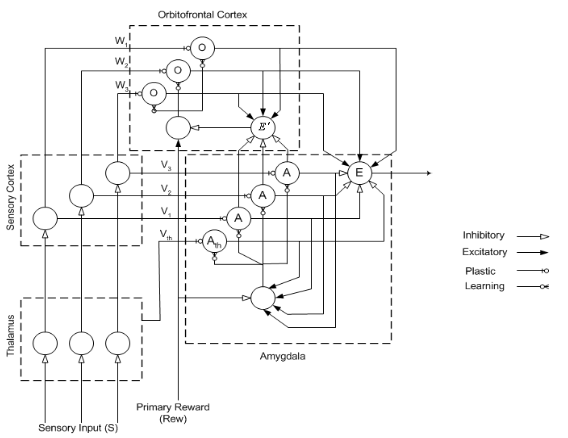

Figure2. Structure of BELBIC [8]

In Thalamus, some poor pre-processing on sensory input signals such as noise reduction or filtering can be done in this part. As a matter of fact, Thalamus is a simple model of brain real thalamus. The Thalamus part prepares Sensory Cortex needed inputs which to be subdivided and distinguished [9].
Based on the context given by the hippocampus, the Orbitofrontal Cortex part is supposed to inhibit the inappropriate responses from the Amygdala, [9].
The emotional evaluation of stimuli signal is carrying out through the Amygdala, which is a small part in the medial temporal lobe in the brain. As result, this emotional mechanism is utilized as a basis of emotional states and reactions. [9].
At first, Sensory Input signals are going into Thalamus for pre-processing on them. Then Amygdala and Sensory Cortex will receive their processed form and their outputs will be computed by Amygdala and Orbitofrontal based on the Emotional Signal received from environment. Final output is subtraction of Amygdala and Orbitofrontal Cortex [9].
One of Amygdalas' inputs is called Thalamic connection and calculated as the maximum overall Sensory Input S as equation (1). This specific input is not projected into the Orbitofrontal part and cannot by itself be inhibited and therefore it differs from other Amygdalas' inputs.

$$A_{th} = \max_i (S_i) \quad (1)$$

Every input is multiplied by a soft weight V in each A node in Amygdala to give the output of the node. The O nodes behaviours produce their outputs signal by applying a weight W to the input signals as well as A nodes. To adjust the V_i, difference between the reinforcement signal and the activation of the A nodes is been made use. For tuning the learning rate the parameter α is used and it sets to a constant value. As shown in equation (2) Amygdala learning rule is an example of simple associative learning system, although this weight-adjusting rule is almost monotonic. For instance, V_i can just be increased.

$$\Delta V_i = \alpha (S_i \max(0, rew - \sum A_j)) \quad (2)$$

The reason of this adjusting limitation is that after training of emotional reaction, the result of this training should be permanent, and it is handled through of the Orbitofrontal part when it is inappropriate [9].
Subtraction of reinforcing signal rew from previous output E makes the signal of reinforcement for O nodes. To put it another way, comparison of desired and actual reinforcement signals in nodes O inhibits the model output.
The learning equation of the Orbitofrontal Cortex is drawn in Eq. (3).

$$\Delta W_i = \beta (S_i \sum (O_j - rew)) \quad (3)$$

Amygdala and Orbitofrontal learning rules are much alike, but the Orbitofrontal weight *W* can be changed in both ways of increase and decrease as needed to track the proper inhibition. And rule of β in this formula is similar to the α ones.

$$\begin{aligned} A_i &= S_i V_i \\ O_i &= S_i W_i \\ E &= \sum A_i - \sum O_i \end{aligned} \quad (4)$$

As presented in equation (4) the difference between *A* nodes and *O* nodes computes output *E*. The *A* nodes outputs are produced according to their rule in prediction of rew signal (reward or stress), though the responsibility of *O* nodes are inhibition of output *E* in while it is necessary.

4. SIMULINK model and the real-time implementation of the proposed controller

In the first part of this implementation, the mentioned controller is implemented for a transfer function system. This implementation is just used as an example and its control response is used to verify the real time implementation part. The second part will describe the implementation process and the connection of SIMULINK to CCS.

4.1. SIMULINK model

The system, which is used as a case study is a linearized model of an AVR (Automatic Voltage Regulator). An AVR is utilized to hold the magnitude of the terminal voltage of a synchronous generator at a specified level. A linearized example of an AVR system with its four components is:

$$G(s) = \frac{10}{0.1s+1} * \frac{1}{0.4s+1} * \frac{1}{s+1} \ , \qquad S(s) = \frac{1}{.01s+1} \tag{5}$$

As the first step, above system is controlled by SIMULINK model of described BELBIC controller. This implementation is carried out in SIMULINK environment and figure (3) shows the related block diagram.

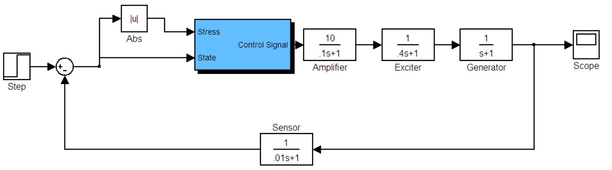

Figure 3 – Block diagram of the control loop composed of linear AVR system and SIMULINK model of BELBIC controller

BELBIC parameters are tuned through some experiments on the control loop and the relation of BELBIC parameters to system states. The result of this control loop is presented in section 4 and it is compared with the real time result.

4.2. Real time implementation

The BELBIC controller is implemented in high-level MATLAB/SIMULINK programming environment, and Real Time Workshop (RTW) and its Target Language Compiler is utilized to generate the required C code. First MATLAB link for CCS integrates C code into the CCS environment, and then CCS compiles it, prepares necessary links, and loads it into the target processor. At last the DSK can process the implemented algorithm.

Now the real-time data exchange can be performed with the help of real-time data exchange (RTDX) channels which can send and receive information between target hardware (DSK) and SIMULINK environment (Figure 4). This data exchange is in real-time and without any interrupt in target processes, and also the performance measures can be monitored as well. Controlling and monitoring of this procedure is handled by on-chip emulation support and with the help of the joint team action group (JTAG).

In order to control the AVR system by using the generated code in CCS, some interface SIMULINK blocks are used through which the data can be transferred to and from the CCS environment. Figure (5) shows the implementation procedure from the SIMULINK model to the target DSP.

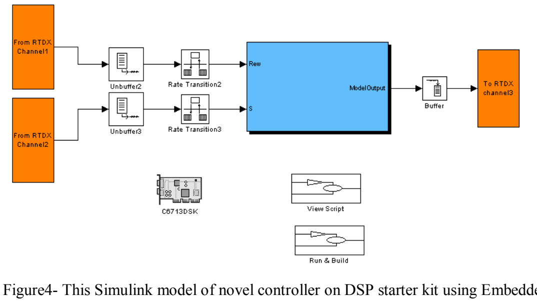

Figure4- This Simulink model of novel controller on DSP starter kit using Embedded Target

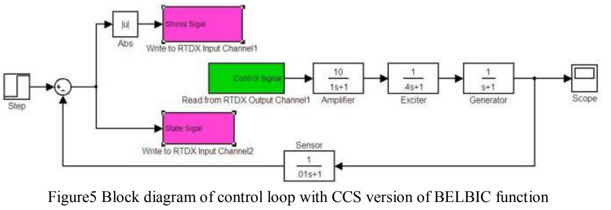

Figure5 Block diagram of control loop with CCS version of BELBIC function

5. Implementation results

In this section the results of SIMULINK and CCS version of BELBIC is presented and compared with each others.

5.1 SIMULINK results

The control response of SIMULINK model to the step reference signal is demonstrated in figure (6).

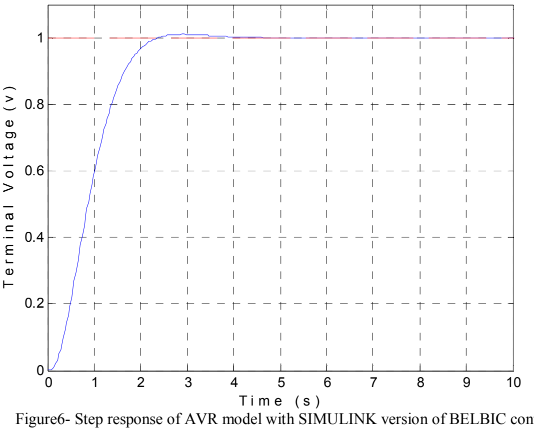

Figure6- Step response of AVR model with SIMULINK version of BELBIC controller

The BELBIC parameters which are tuned by an optimization algorithm are presented in table 1.

Table1- BELBIC parameters which are achieved through an optimization algorithm.

k_a	k_o	k_{th}	m_a	m_o	m_{th}
0.422	0	0.558	0.213	0.939	0.524

Table 2 demonstrates the detail of the control response performance and this information is used as verification measures in section 4.2

Table 2- Control performance of SIMULINK version of BELBIC controller

Overshoot	T_{rise} *(s)*	$T_{settling}$ *(s)*	$Error_{ss}$ *(v)*
1%	1.303	1.886	31e-5

5.2 Verification of model

The step response of control loop with CCS generated code is compared with SIMULINK results and they are presented in figure (7). As shown in this figure, the generated code has close behavior to the original BELBIC function and is capable of controlling the AVR linear model with the original BELBIC parameters.

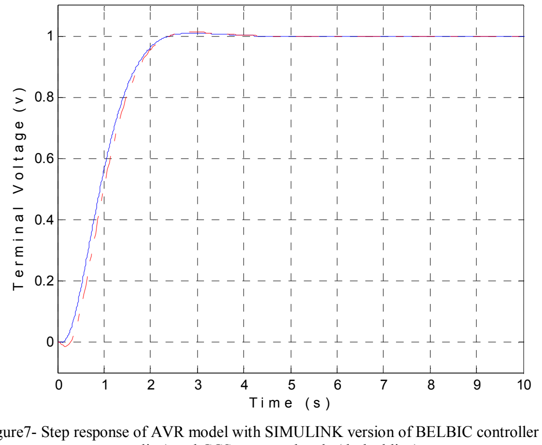

Figure7- Step response of AVR model with SIMULINK version of BELBIC controller (solid line) and CCS generated code (dashed line)

Detail of control performance and the comparison are presented in table 3.

Table3- Control performance of CCS version of BELBIC controller

Overshoot	T_{rise} *(s)*	$T_{settling}$ *(s)*	$Error_{ss}$ *(v)*
1.34%	1.26	1.98	32e-5

As seen in both table (3) and figure (7), there is a slight difference between the two step responses. This disparity is due to different sampling times of CCS and SIMULINK version of BELBIC. Due to DSK board limitations in real time data exchange the RTDX channels are used with 100 Hz frequency while the SIMULINK model can work with higher frequency values, even though the CCS generated code can control the AVR model with in satisfactory manners.

5.3 Experimental results of implemented BELBIC on TMS320C6713

After BELBIC function has been generated in CCS environment, the resulted code is implemented on TMS320C6713 DSK board. The detail of this implementation is calculated and summarized in table (4). In this table values are calculated based on sampling frequency equal to 100Hz.

Table4- Experimental results related to Implementation of BELBIC on DSK board

Result of software pipeline & optimization	
Consumed FLOPS	0.2708 $FLPOS^{*}$
Total time to complete	3.01 μs
Maximum allowable frequency	2.65MHz
Cycle needed for Function1(Amygdala)	461
Cycle needed for Function 2(Orbitofrontal)	216
Total cycle needed	677
*Floating Operation Per Second	

6. Conclusions

In this paper, an emotional controller, BELBIC that is based on mammalian limbic learning algorithms is implemented on a digital signal processor (DSP) TMS320C7613. In order to verify the implementation result and compare it with SIMULINK model the BELBIC controller is utilized to control an AVR system. As presented in section 5.1 BELBIC controlled the linear model of AVR system, and in the section 5.2 it shows that the implementation results have been verified by comparison with SIMULINK version in satisfactory manners.

Converting the BELBIC function from the SIMULINK environment to an executable code to run in real time on the DSP hardware was done fully-automated. This conversion can be done for other similar intelligent algorithms such as GA, Neural Networks and Reinforcement Learning (RL) based algorithms, and there is no need of expert programming skills to do this conversion.

7-REFERENCES

[1] M. R. Jamali, M. Valadbeigi, M. Dehyadegari, Z. Navabi and C. Lucas, "Toward Embedded Emotionally Intelligent System", *IEEE EWDTS*, Yerevan, September. 2007

[2] L. M. Reyneri, "Implementation issues of neuro-fuzzy hardware: Going toward hw/sw co design", *IEEE Trans Neural Networks*, vol. 14, no. 1,pp. 176–194, Jan. 2003.

[3] S. M. Fakhraie and K. C. Smith, "VLSI-Compatible Implementations for Artificial Neural Networks". *Kluwer Academic Publishers*, Norwell, Massachusetts, 1997.

[4] A.Pedram, M. R. Jamali, C. Lucas, and S. M. Fakhraie, "Reconfigurable Parallel Hardware for Computing Local Linear Neuro-Fuzzy Model", *International Conference on Parallel Computing in Electronics*, Poland, 2006.

[5]R. Chassing, "Digital Signal Processing and Applications with the C6713 and C6416 DSK", *John Wiley*, 2005

[6] Zarzo Fuertes, "OFDM PHY Layer Implementation based on the 802.11 a Standard and system performance analysis", *Student thesis*, Linköping University, Department of Electrical Engineering. 2005

[7] R. Milasi, C. Lucas, and B. Araabi, "Speed Control of an Interior Permanent Magnet Synchronous Motor Using BELBIC ", In M. Jamshidi, L. Foulloy, A. Elkamel, and J. S. Jamshidi (eds.), *Intelligent Automations and Control- Trends, Principles, and Applications*, Albuquerque, NM, USA: TSI Press Series: Proceedings of WAC, Volume16, M. Jamshidi (series editor), Pages: 280- 286, 2004.

[8] C. Lucas, D. Shahmirzadi, N. Sheikholeslami, "Introducing BELBIC: Brain Emotional Learning Based Intelligent Controller", *International Journal of Intelligent Automation and Soft Computing*, pages11- 22, 2004.

[9] J. Moren, C. Balkenius, "A Computational Model of Emotional Learning in the Amygdala: From animals to animals ", *6th International Conference on the Simulation of Adaptive Behavior*, Cambridge, MIT Press, 2000.

[10] Rev. B, "TMS320C6713 DSK Technical Reference", *SPECTRUM DIGITAL*, INC. November 2003