Proceedings of Engineering & Technology (PET)
pp. 1-5

2nd International Conference on Automation, Control, Engineering and Computer Science (ACECS-2015)
Proceedings of Engineering & Technology (PET)

FPGA Implementation of PID-Sliding Mode controllers

Fares Sassi
Laboratory of Analyze, Conception and Control of Systems (LACCS), Team: Modelisation and Control of renewable Energies Systems, University El Manar, Faculty of Sciences of Tunis, Tunis, Tunisia
Sassifares31@gmail.com

Abdelkader Mami
Laboratory of Analyze, Conception and Control of Systems (LACCS), Team: Modelisation and Control of renewable Energies Systems, University El Manar, Faculty of Sciences of Tunis, Tunis, Tunisia
mami.abdelkader@planet.tn

***Abstract*— In this paper, we propose an implementation of synthesizable VHDL programs of proportional-integral-derivative (PID) and sliding mode controllers (SMC) on a map XC3S700A Xilinx Starter Kit using the Xilinx ISE 10.1 software. The sliding mode control is applied to a second order state system and closed loop performance is compared with conventional proportional-integral-derivative controller.**

Keywords—FPGA; Matlab/Simulink; PID; Sliding mode; VHDL.

I. INTRODUCTION

With technological advancement in the field of microelectronics, new digital solutions such as FPGAs (Field Programmable Gate Array) are available and can be used as targets of the implementation of digital control algorithms [1]. The use of these hardware solutions allows finding some analog performance while keeping the advantages of digital solutions. In addition, these solutions can meet the new demands of modern controls. Indeed, in addition to improving control performance through the reduction of computation time, the parallelism of hardware solutions can be integrated on a single target a plurality of algorithms that provide different functions that can work independently of each other [6].

The target FPGA device used in this paper is Spartan-3A manufactured recently by Xilinx [7]. This board provides all the tools required to quickly beginning designing and verifying Spartan-3 platform designs. While the modules are implemented also suited to other high density FPGAs, designs are based on 50 MHz clock and should be updated if different frequency is used.

In this paper, we present an implementation of PID-SMC controllers applied to a second order state system using synthesizable VHDL integer programs [2]. The results are compared with those obtained in Matlab.

The paper is structured as follows: in section two, we present system to control and Simulink implementation of controlled system. In section three, we present simulation and implementation results. The last section is devoted to conclude this paper.

II. PID-SLIDING MODE CONTROLLERS

A. System to control

In this example, we consider a two coupled tanks system [4] [5]. The level of tank 2 can be described by the following discrete-time equation obtained from their continuous time counterparts by discretization, using a sampling time of 0.1s and Euler's first order approximation for the derivative and based on the observer canonical form [3]:

$$\begin{pmatrix} \dot{x}1 \\ \dot{x}2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -0.00416 & -0.516 \end{pmatrix} \begin{pmatrix} x1 \\ x2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.00416 \end{pmatrix} u$$
$$y = (1 \quad 0) \begin{pmatrix} x1 \\ x2 \end{pmatrix} \tag{1}$$

B. Simulink model of sliding mode control

The application of a SMC controller in a feedback control system is shown Fig. 1.

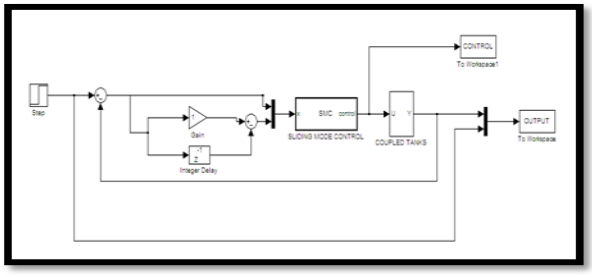


Fig. 1. Simulink implementation of controlled system

ISSN: 2356-5608

The MatLab function SMC is depicted in the appendix of this report. The global variables c, phi1, phi2 and kf are assigned the following values in MatLab. When this program is run, the two state variables, error and error derivative, are fed into the multiplexer. These states are multiplexed onto one line and used as the input data to the MatLab function entitled SMC. This program must be in the same working directory as the Simulink programs. This MatLab function calculates the switching line 's' using the command: sigma = state2 + (c×state1). The function then multiplies sigma by state1 and does a test on the result. If the result is ≥ 0, then teta1 is assigned the value 60. In the opposite case, teta1 is assigned the value 40. After that, the function multiplies this value of teta1 with state1 and this value is the control output, and, the control output is then fed to the system. This is how the variable structure is achieved with this sliding mode controller. The variable, kf, is the disturbance element that is initially set to zero.

C. Simulink model of PID controller

The application of a PID controller in a feedback control system is shown Fig. 2.

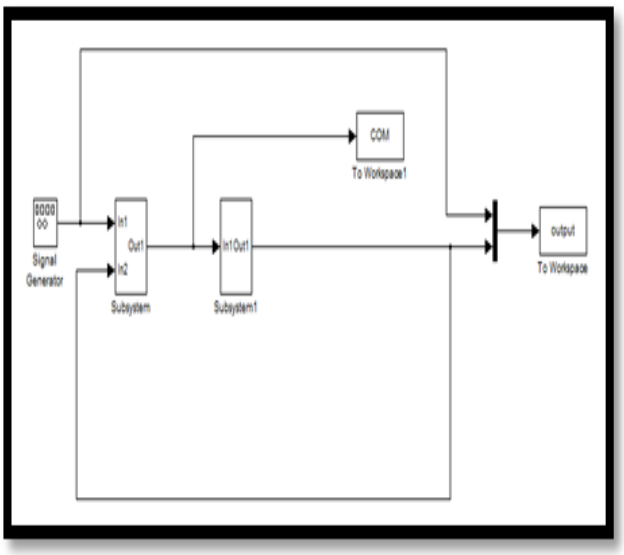


Fig. 2. Simulink implementation of controlled system

The corresponding discrete PID equation is given [12]:

$$U(z) = K_i T_e \left(\frac{z}{z-1}\right) E(z) - \left[K_p + \frac{K_d (z-1)}{T_e z}\right] y(z) \qquad (2)$$

Where

$$E(z) = Y_c(z) - Y(z) \qquad (3)$$

$Y_c(z)$: The set point signal.

$Y(z)$: The feedback signal.

$E(z)$: The error signal.

$U(z)$: The control signal.

III. SIMULATION AND IMPLEMENTATION RESULTS

A. Simulation results by Matlab

The resulting curves of the Matlab simulation are illustrated in Fig. 3 and Fig. 4.

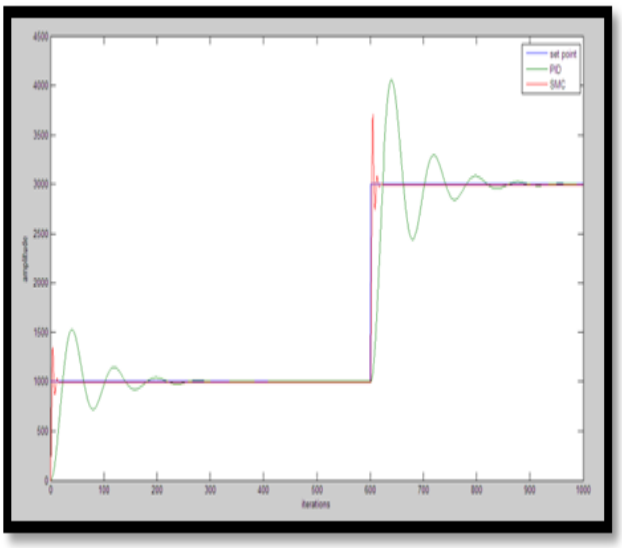


Fig. 3. Set point and output signals of the matlab

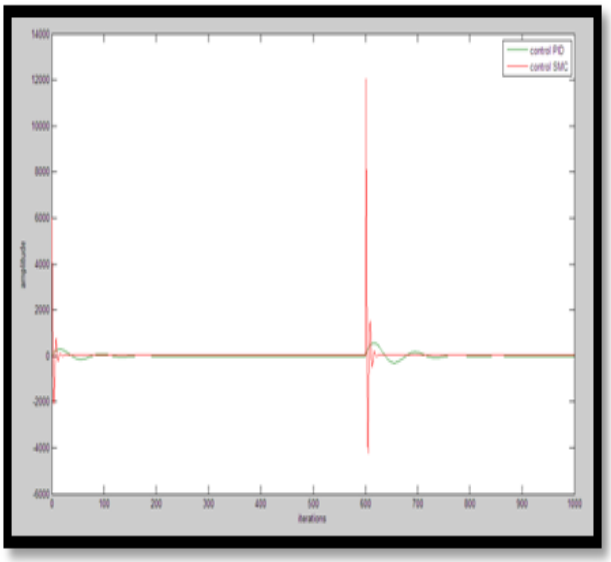


Fig. 4. Matlab control signals of PID-SMC controllers

B. Implementation results

The VHDL programs "PID-SMC, system and digital to analog converter (DAC)" is implemented in Spartan 3A map using the Xilinx ISE 10.1 software and JTAG cable.

The output signal on channel 2 and the set point signal on the chain 1 are then displayed together on the screen of the oscilloscope. The result of the implementation is depicted in Fig. 5 and Fig. 6.

The output signal on channel 2 and the signal control of the chain 1 are then displayed together on the screen of the oscilloscope. The result of the implementation is shown in Fig. 7 and Fig. 8.

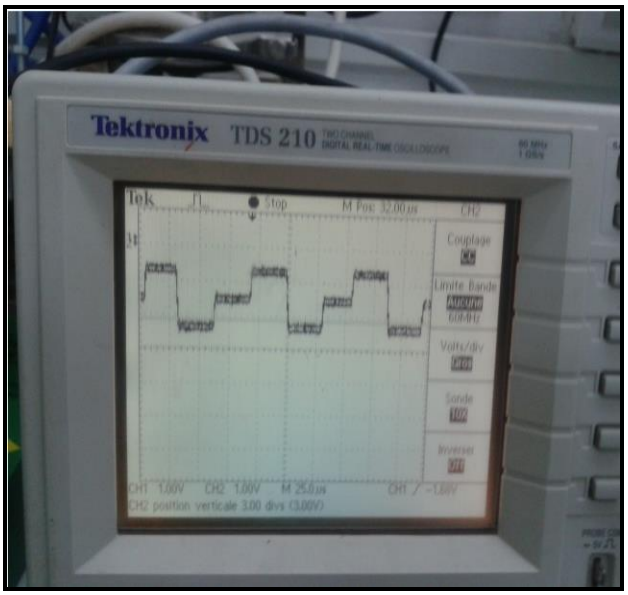


Fig. 5. Result of implementation of the oscilloscope set point and output signals "SMC, system and DAC" design

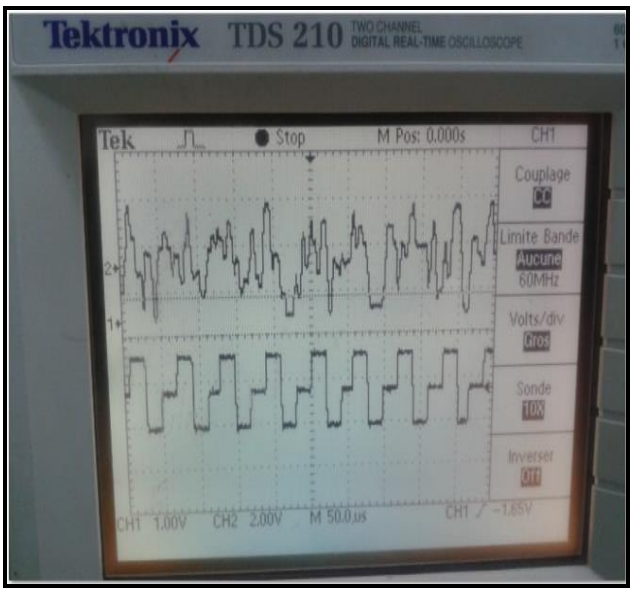


Fig. 6. Result of implementation of the oscilloscope control and output signals "SMC, system and DAC" design

Fig. 7. Result of implementation of the oscilloscope set point and output signals "PID, system and DAC" design

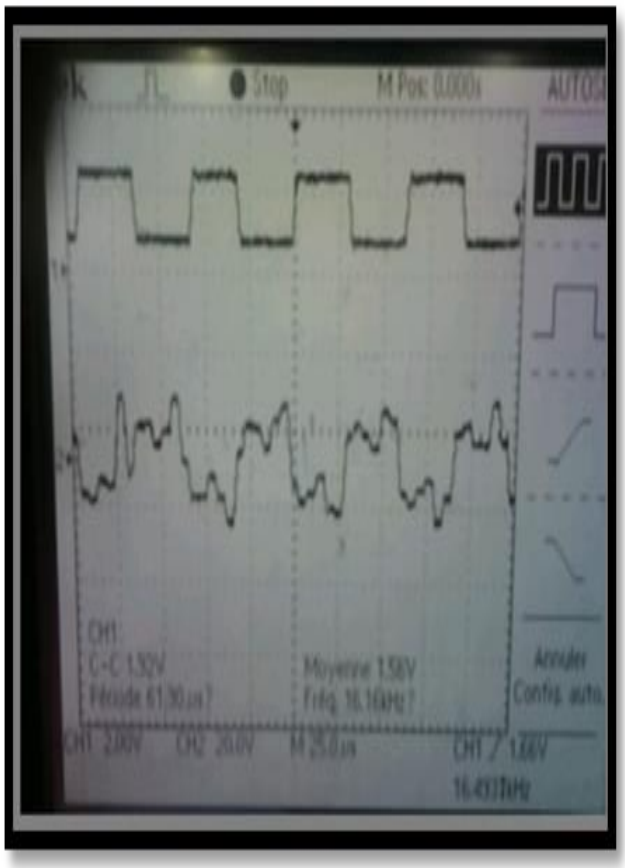
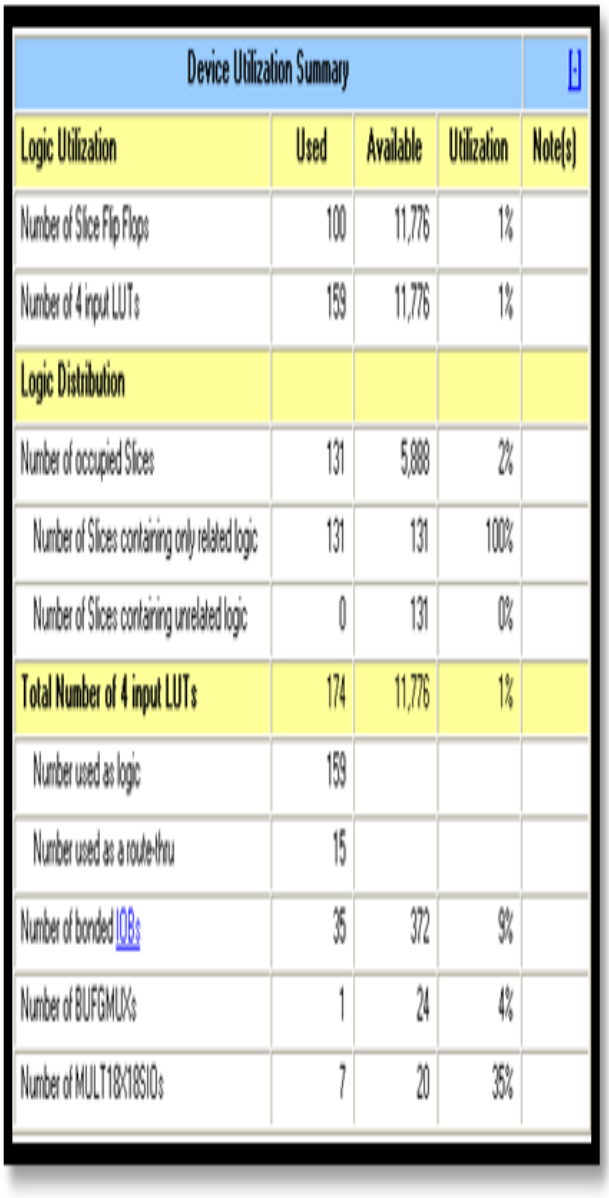


Fig. 8. Result of implementation of the oscilloscope control and output signals "PID, system and DAC" design

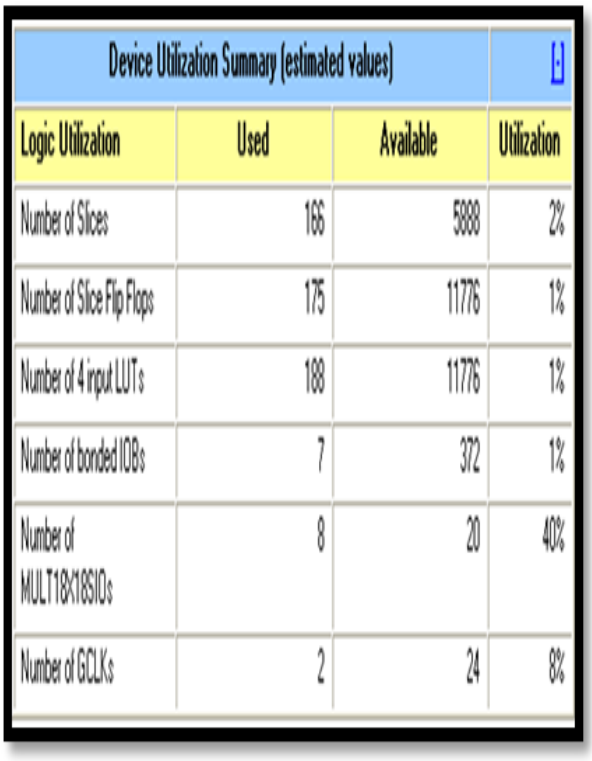
C. Interpretation results

After the execution of the synthesis process, the Xilinx ISE software generates the statistics on the number of used hardware resources which are depicted in Fig. 9 and Fig. 10. The figures show that statistics on the materiel resources used to implement the "SMC, system and DAC" design is less than the "PID, system and DAC" design. The table. 1 shows that the SMC controller can yield a better dynamic performance than the PID controllers in terms of rising time, setting time, maximum overshoot.



Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	100	11,776	1%	
Number of 4 input LUTs	159	11,776	1%	
Logic Distribution				
Number of occupied Slices	131	5,888	2%	
Number of Slices containing only related logic	131	131	100%	
Number of Slices containing unrelated logic	0	131	0%	
Total Number of 4 input LUTs	174	11,776	1%	
Number used as logic	159			
Number used as a route-thru	15			
Number of bonded IOBs	35	372	9%	
Number of BUFGMUXs	1	24	4%	
Number of MULT18X18SIOs	7	20	35%	

Fig. 9. Statistics on the material resources used to implement the "SMC, system and DAC" design



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	166	5888	2%
Number of Slice Flip Flops	175	11776	1%
Number of 4 input LUTs	168	11776	1%
Number of bonded IOBs	7	372	1%
Number of MULT18X18SIOs	8	20	40%
Number of GCLKs	2	24	8%

Fig. 10. Statistics on the material resources used to implement the "PID, system and DAC" design

Characteristics	SMC	PID
Overshoot	7.06	10.56
Peak	3706	4055.8
Setting Time	2.6	37.8

Table. 1. Performances summery

IV. CONCLUSION

During the present investigation, an implementation of PID-SMC controllers on a map XC3S700A, FPGA-based, is performed by writing synthesizable VHDL integer programs. We concluded that the sliding mode control performs better than PID controller.

REFERENCES

[1] E. Monmasson and Y. A. Chapuis, "Contributions of FPGAs to the control of Electrical System, A Review," IEEE Trans. Ind. Electron. Society Newsletter, vol.49, no.4, pp.8-15, December. 2002.

[2] D.L. Perry: “VHDL”, McGraw-Hill, 2004.

[3] Control System Design. An Introduction to State Space Methods. Bernard Friedland. 1987. McGraw-Hill.

[4] Improvement of mold level control system in continuous caster using sliding mode control for transitional operation (Journal paper) Transactions of the Society of Instrument and Control Engineers. Watanabe, T.;Omura, K.; Furukawa, K.; Watanabe, S.; Konishi, M. Vol. 36, Iss. 10, Pg.884-6. October 2000.

[5] MK. Khan, and SK. Spurgeonm, 2006.“Robust MIMO water level control in interconnected twin tanks using second order sliding mode control *Control Eng Pract J.* vol. 14, pp. 375–86.

[6] M.W. NAOUAR, « Commande numérique à base de composants FPGA d'une machine synchrone », algorithmes de contrôle du courant, Thèse de Doctorat, Ecole Nationale d'Ingénieurs de Tunis et l'Université de Cergy Pontoise, Tunis, 2007.

[7] Spartan-3A/3AN FPGA Starter Kit Board User Guide, Xilinx, 2008.