International Conference on Automation, Control, Engineering and Computer Science (ACECS'14)
Proceedings - Copyright IPCO-2014, pp.1-4
ISSN 2356-5608

FPGA Implementation of Sliding Mode Control

Fares Sassi #1, Abdelkader Mami #2

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

[1]sassifares31@gmail.com

[2]mami.abdelkader@planet.tn

ABSTRACT

In this paper, we propose an implementation of a synthesizable VHDL program of Sliding Mode control (SMC) on a map XC3S700A Xilinx Starter Kit using the Xilinx ISE 10.1 software. The control strategy was applied to a second order state system. The VHDL language was used as a programming tool. The use of FPGA circuits presents a good choice regarding to the problem of execution time. A SMC matlab program was also implemented in order to make a performance comparison.

Keywords— MATLAB/SIMULINK; Sliding Mode Control;

VHDL

I. INTRODUCTION

The first realizations of digital implementation of control algorithms were performed using microprocessors. They also have a large economic interest and a better design flexibility. 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 for the implementation of digital control algorithms. 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. The target FPGA device used in this paper is Spartan-3A manufactured recently by Xilinx [6]. Design development and debugging is carried on a low-cost, full featured kit provided by Digilent. 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 a sliding mode control applied to a second order state system using a synthesizable VHDL integer program. The results are compared with those obtained in Matlab.

The paper is structured as follows: in Section II and Section III, we present a theoretical review of SMC and system to control. In Sections IV AND V, we present the controlled state system VHDL design, their simulations and implementation results. The last section is devoted to conclude this paper.

II. SLIDING MODE CONTROL

The term 'sliding mode control' first appeared in the context of variable-structure systems [5]. Soon sliding modes became the principal operational mode for this class of control systems. The following figure illustrates an example of a variable structure system [4] [1]. The horizontal axis should be denoted by x.

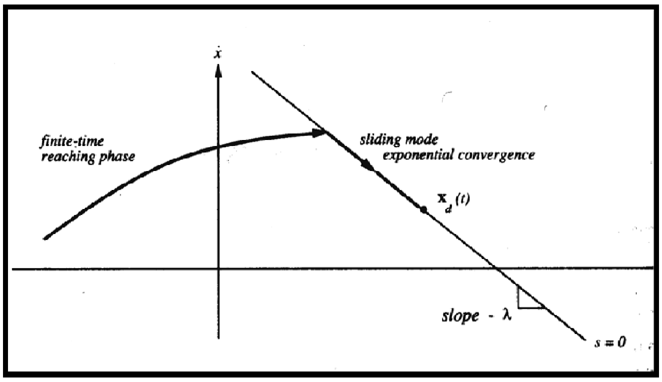
$\dot{x}$
finite-time reaching phase
sliding mode exponential convergence
$x_d(t)$
slope - λ
s = 0

Fig. 1 Sliding Mode in the Phase Plane

The phase plane is shown on the axis of the two state variables, x and $\dot{x}$. The switching line is drawn with a slope of λ, as shown. The trajectories are plotted on the phase plane. When the trajectory reaches the switching plane, the structure of the feedback loop is adaptively altered to slide the system state along the switching plane.

$$u = \theta \times state1 \tag{1}$$

where θ has the following structure

$$\theta = 60 \text{ if } s \times state1 \succ 0,$$

or

$$\theta = 40 \text{ if } s \times state1 \prec 0,$$

and

$$s = x\dot{1} + (c \times x1)$$
$$s = x2 + (c \times x1)$$

The variable, c, in the equation computes the slope of the switching line in the phase plane. This controls the switching logic to switch between the two structures in the control system. In Fig. 2, the slope is denoted as -λ. In this project, a MatLab program (function) is used to test the two states of the system and to return a value that depends on the position of the trajectory in the phase plane.

III. SYSTEM TO CONTROL

In this example, we consider a two coupled tanks system [2] [3]. 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 :

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

$$y = (1 \quad 0) \begin{pmatrix} x1 \\ x2 \end{pmatrix}$$

IV. IMPLEMENTING A SLIDING MODE CONTROL ON THE MAP XILINX SPARTAN 3A STARTER KIT XC3S700A

A. SIMULINK MODEL

The State Space representation of the system is written into the Simulink model in Fig. 2.

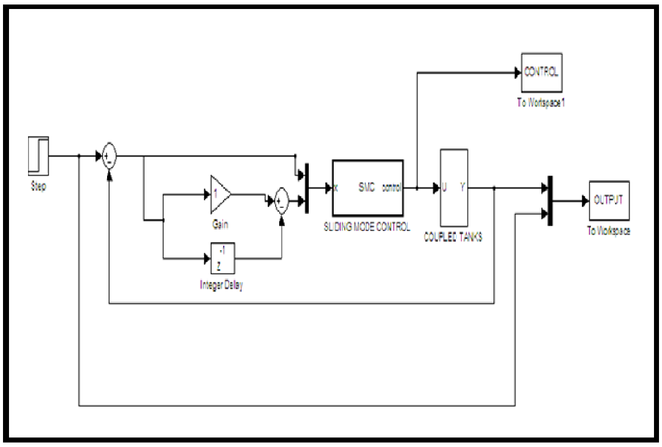

Fig. 2 Model of Sliding Mode control system

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.

B. VHDL Programming

The design includes:

- The "Set_point": integer type written in VHDL as a state machine.
- . The "SMC": the VHDL algorithm block written in integer.
- . The "system": the VHDL algorithm block written in integer.

The three blocks are synchronized by the same clock "clk".

V. SIMULATION AND IMPLEMENTATION RESULTS

In this section, we present the simulations of the programs carried out on Simulink and Xilinx ISE 10.1 with Xilinx ISE simulator as well as the interpretation results.

A. Simulation Results By Simulink

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

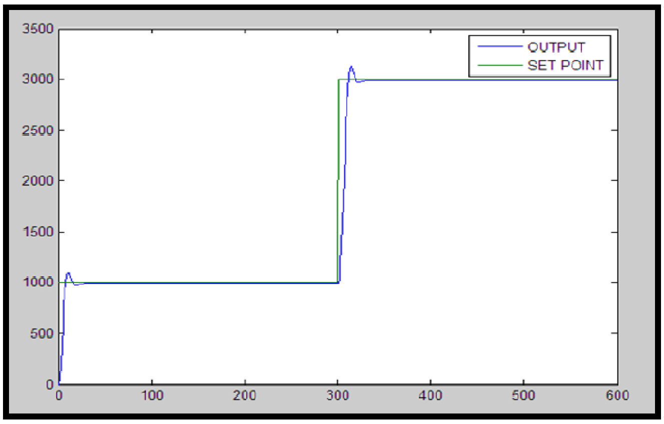

Fig. 3 Set point and output signals of the matlab

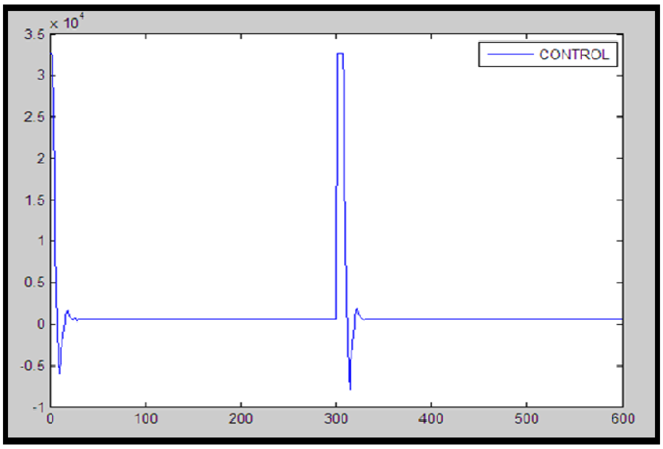

Fig. 4 Matlab control signal of Sliding Mode control

B. *Simulation By The ISE Simulator*

The simulation using the ISE simulator is shown in Fig. 5. The set point varies between 1000 and 3000.

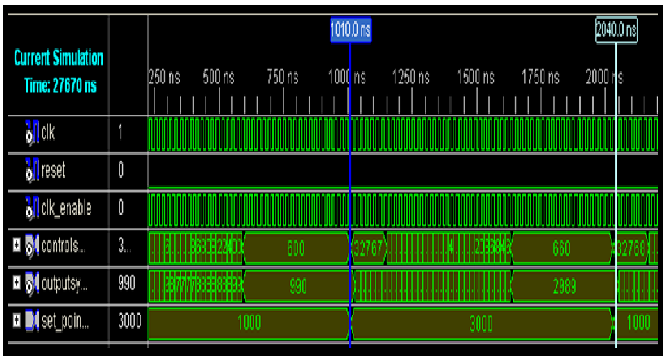

Fig. 5 Simulation of sliding mode control system by the ISE simulator

The clock frequency is fixed at 50 MHz (20ns) which is equal to the clock frequency of the XC3S700A Starter Kit.
The output reaches the set point value after 20 iterations.
We note that the value of the output is stabilized with the first set point value (1000), when there is a change in the set point value, the value of the output starts to increase until it reaches the second value set point (3000).

C. *Implementation Results*

The VHDL program "SMC, system and 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. 10.
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. 11.

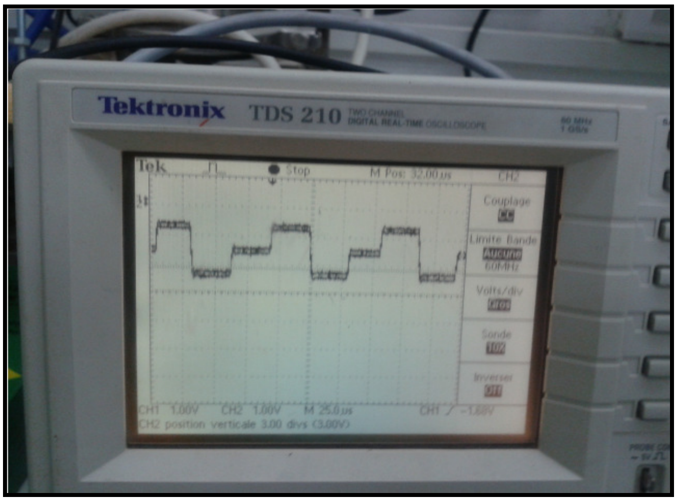

Fig. 10 Result of the implementation of the oscilloscope set point and output

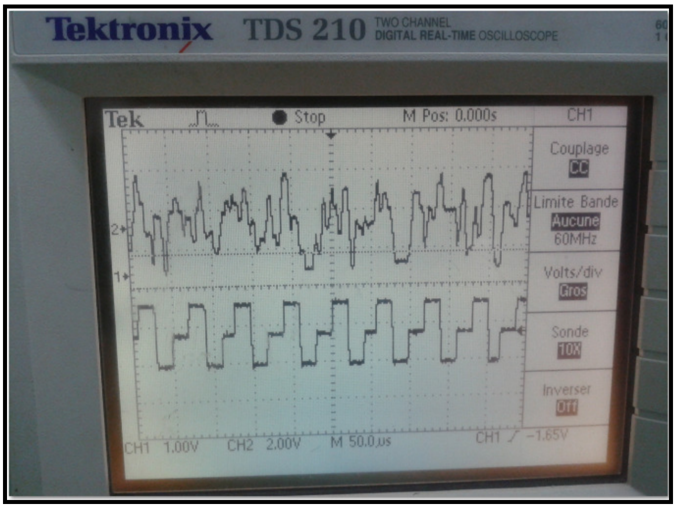

Fig. 12 Result of the implementation of the oscilloscope control and output

D. *Interpretation Results*

The difference between the results of simulation and implementation on FPGA is due to the difference interpretation of integer type used in VHDL program and real type used in the Matlab program.

The implementation of sliding mode control on a FPGA to the autonomous system and provides enhanced reliability due to the fact that this map contains analog blocks.

VI. CONCLUSIONS

In the current investigation, an implementation of sliding mode control on a map XC3S700A, FPGA-based, is performed by writing a synthesizable VHDL integer program. A comparison of VHDL signals to those obtained by Matlab is carried also out.

As a future extension of the current investigation, we will try to make a comparison between the VHDL program and the Matlab program, to control a real system.

REFERENCES

[1] Applied Non Linear Control. Jean-Jacques E. Slotine & Weiping Li. 1991. Prentice Hall.

[2] 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.

[3] 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.

[4] Sliding Mode Control in Electromechanical Systems. Vadim Utkin, Jurgen Guldner & Jingxin Shi. 1999. Taylor & Francis Ltd.

[5] Sliding Mode Control. Theory and Applications. Christopher Edwards & Sarah K. Spurgeon. 1998. Taylor & Francis Ltd.

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