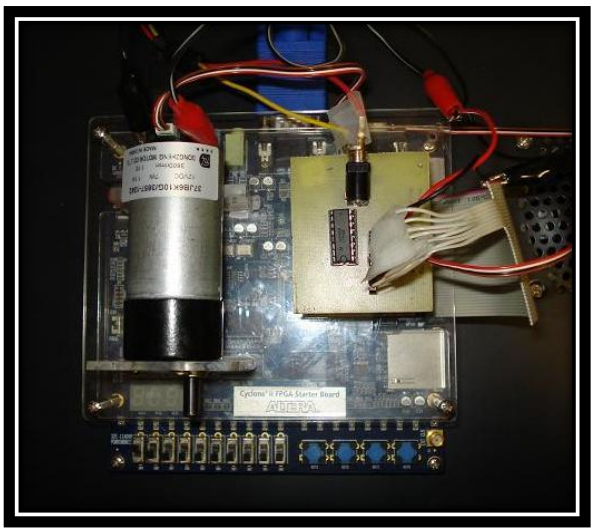
FPGA based Control Systems

DC Motor velocity, position control and white noise fIlter

Nikolaou Panagiotis
Computer Science and Engineering
Technical University of Kavala
Tripoli, Greece

***Abstract*— This paper presents the methodology development of (1) A real-time machine control application on a FPGA based motor driver with PID (Proposal - Integral - Derivative) controller for the velocity control of a DC Gear Motor connected through expansion header [JP2-GPIO 1] at Altera DE1 Board. The PID controller has been implemented at the Cyclone II FPGA and the embedded microprocessor NIOS II (SoPC - System on Programmable Chip technology) with programming language C. The motor driver (velocity variations) has been programmed by the Pulse Width Modulation (PWM) technique and driven with a L293D H -Bridge. The frequency method has been used as the RPM feedback measurement technique from the decoded measurements of the optical encoder (E4P) of the DC Gear Motor. (2) A NIOS II 2nd order system simulation of a DC Motor position control with PID controller connected through UART to a Personal Computer running Matlab and sending the unit step function as the desired position of the motor, (3) A NIOS II 1st order system simulation of a white noise filter to a noisy signal transferred through UART by Matlab software.**

Keywords — FPGA, PID, NIOS II , closed loop systems, DC Motor Control, Matlab



12V/1.5Kg-cm/365RPM 10:1 DC Gear Motor w/ Encoder
Altera DE1 - Cyclone II EP2C20F484C7

I. INTRODUCTION

A real-time system is one which "controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at that time.

Martin, James (1965) *Programming Real-time Computer Systems.*

Field programmable gate arrays are reprogrammable silicon chips and is a new technology that can comprise the base for the development of real time embedded systems based on SoPC implementations and the technology of FPGA's. Some of the applications that can be developed at these units with the hardware description programming languages VHDL, Verilog-HDL at Quartus environment or at NIOS II environment with high level programming language 'C' and HAL instructions, are the embedded control systems (opened / closed loop), digital and analog signal processing applications, robotics and graphics applications.

The approach that systems have been presented at the past for the DC Motor velocity control is usually implemented with microprocessor or microcontroller, analog chip arrays, ASIC, digital to analog (D/A) and analog to digital(A/D) converters and tachometer for the feedback voltage measurements to the closed loop control system.

Some advantages of systems based on FPGA in opposition to systems based on ASIC are the versatility in systems rapid prototyping, the total cost, the reliability and the maintenance. Reprogrammable chip have also the same software suppleness that is operated in a system based processor but it does not constrained from the number of available processing cores and overpowering on the performance which is based to the actual parallism on hardware.

The purpose of the experiment on the research that this conducted is the development of a FPGA based, real versatile and tunable closed loop control system that is designed at the architecture of embedded processor NIOS II for the control of a DC Motor velocity and graphical production and presentation of this response.

II. MAIN

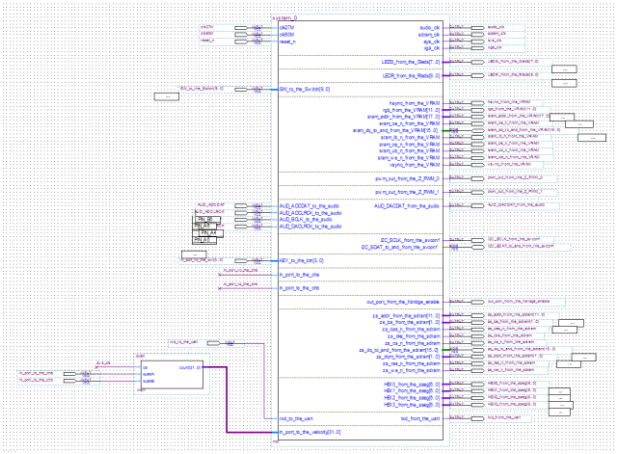
A. *ZNUC RTS, a DC motor driver, with velocity control and motor response vizualization at VGA screen*

In this experiment we will investigate the response of a 2nd order real time closed loop system in which a DC Gear Motor is connected through expansion header slot [JP2-GPIO 1] at the Altera DE1 Board. The velocity variation of the motor are controlled with the industrial PID controller is implemented on NIOS II microprocessor with C programming language and HAL instructions. The system is programmed also to produce at a connected VGA screen the response of the motor.

NIOS II processor architecture at SOPC Builder (developed based on DE1 Media computer)

sseg	Parallel Port	sys_clk	0x01405100	0x0140510f
btn	Parallel Port	sys_clk	0x014050e0	0x014050ef
Gleds	Parallel Port	sys_clk	0x014050f0	0x014050ff
onchip_memory	On-Chip Memory (RAM or ROM)	sys_clk	0x01402000	0x01403fff
cpu	Nios II Processor	sys_clk	0x01404800	0x01404fff
sdram	SDRAM Controller	sys_clk	0x00800000	0x00ffffff
systimer	Interval Timer	sys_clk	0x01405000	0x0140501f
stamptimer	Interval Timer	sys_clk	0x01405020	0x0140503f
jtag_uart	JTAG UART	sys_clk	0x01405130	0x01405137
uart	UART (RS-232 Serial Port)	sys_clk	0x01405040	0x0140505f
sysid	System ID Peripheral	sys_clk	0x01405138	0x0140513f
hbridge_enable	PIO (Parallel I/O)	sys_clk	0x01405060	0x0140506f
cha	PIO (Parallel I/O)	sys_clk	0x01405070	0x0140507f
chb	PIO (Parallel I/O)	sys_clk	0x01405080	0x0140508f
velocity	PIO (Parallel I/O)	sys_clk	0x01405090	0x0140509f
VRAM	chu_avalon_vga	sys_clk	0x01000000	0x013fffff
Z_PWM_0	Z_PWM	sys_clk	0x014050a0	0x014050af
Z_PWM_1	Z_PWM	sys_clk	0x014050b0	0x014050bf
External_Clocks	Clock Signals for DE-Series Board Peri...	*multiple*	0x01405140	0x01405141
audio	Audio	sys_clk	0x014050c0	0x014050ef
avconf	Audio and Video Config	sys_clk	0x014050d0	0x014050df
Switch	Parallel Port	sys_clk	0x01405110	0x01405111
Rleds	Parallel Port	sys_clk	0x01405120	0x0140512f

NIOS II Schematic and connections, decoder unit and connections at Altera Quartus



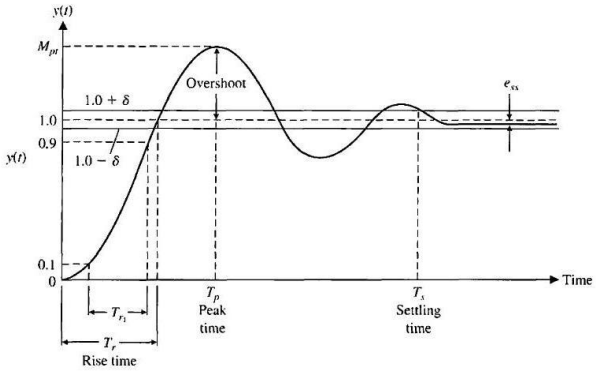
DC Motor transfer function from the input armature voltage to the resulting speed change

$$\frac{\omega(s)}{V_a(s)} = \frac{(K_{ma}/L_aJ)}{(s + R_a/L_a)(s + c/J) + (K_bK_{ma}/L_aJ)}$$

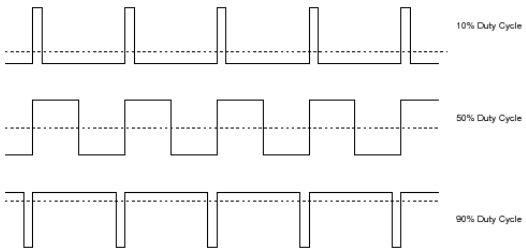
Discrete PID controller equation:

$$dCO = K_i edt - K_p dPV - K_d \frac{d^2PV}{dt}$$

PID graphical response:



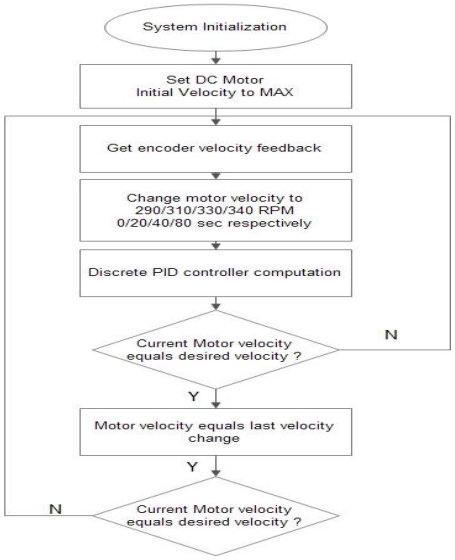
PWM (Pulse Width Modulation) is the technique to use a digital signal to generate an analogue output signal. This is usually used to control the average power to a load in a motor speed control circuit.



RPM frequency method measurement:

$$\frac{Revoltuions}{Minute} = \frac{\left(\frac{Pulse\ Frequency\ in\ pulses}{sec}\right) x \left(\frac{60\ sec}{min}\right)}{\left(\frac{Sensor\ pulses}{revolution}\right)}$$

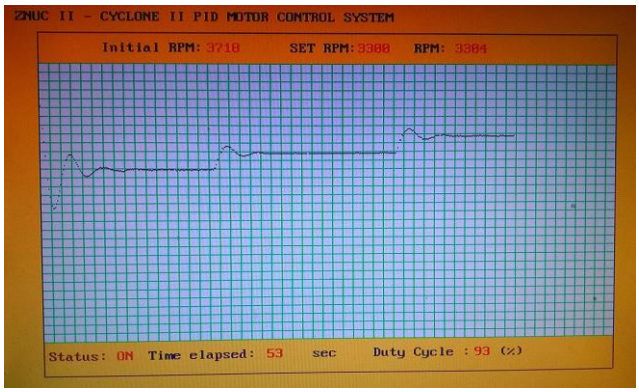
System software flowchart



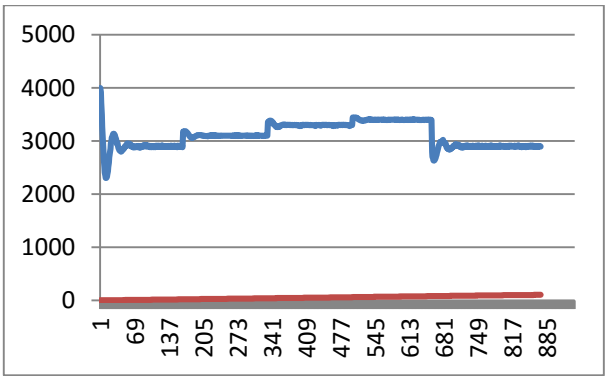
1st experiment results

I Gain = 5, P Gain = 15, D Gain =2

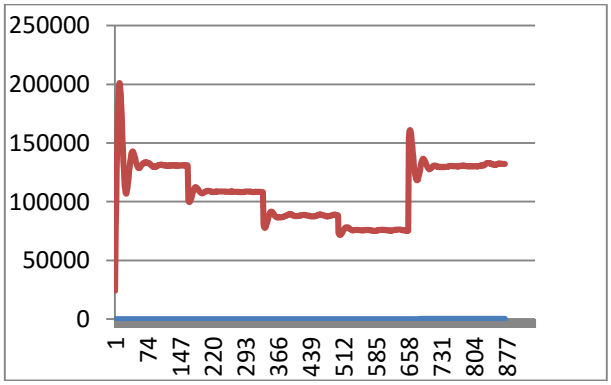
DE1 FPGA VGA OUTPUT (PID velocity response)



RPM (10:1) in relation to Time(s)

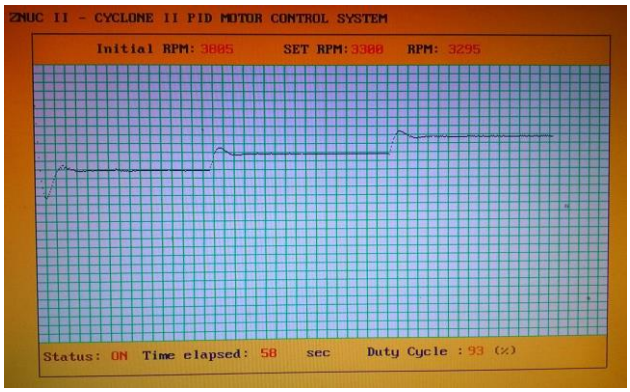


Duty cycle in relation to Time(s)

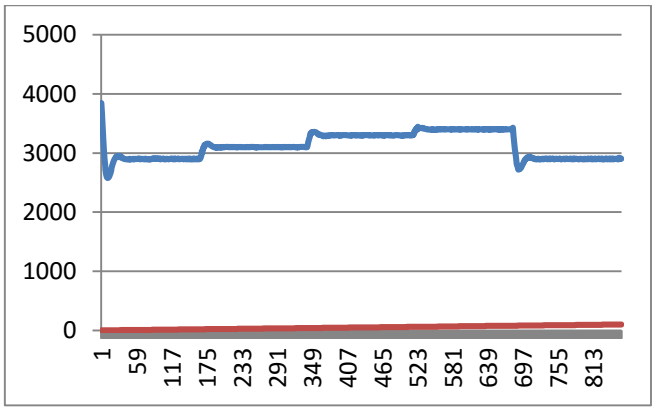


I Gain = 5, P Gain = 25, D Gain =8

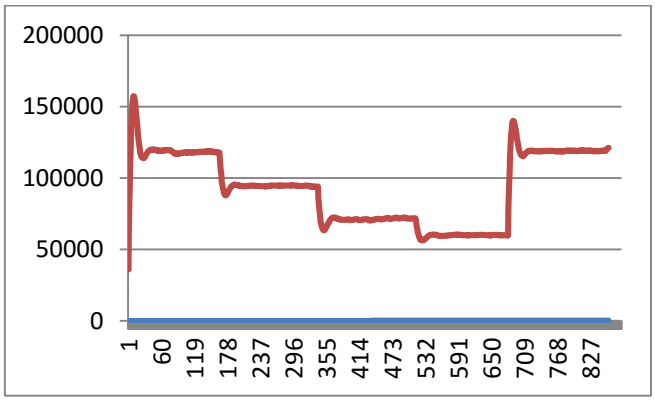
DE1 FPGA VGA OUTPUT (PID velocity response)



RPM (10:1) in relation to Time(s)



Duty cycle in relation to Time(s)



B. DC Motor position control

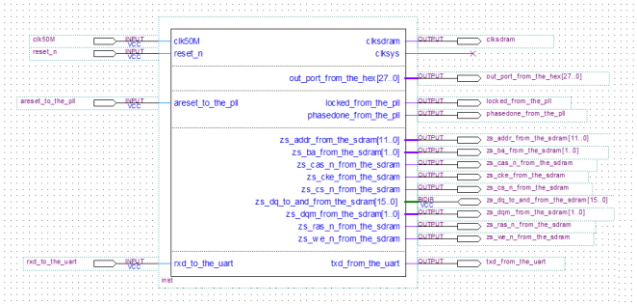
In this experiment we will investigate the response of a 2nd order closed loop system in which through Matlab environment we create a signal with transfer function the unit step function and we will simulate the position of a dc motor. In communication through RS232 UART at the FPGA Cyclone II a C code produced PID controller will approximate the current motor position at the given time and will return this measured data to the Matlab environment for plotting the response of the system.

Unit step function transfer function

$$\boldsymbol{u_c(t) = u(t-c) = H(t-c)}$$

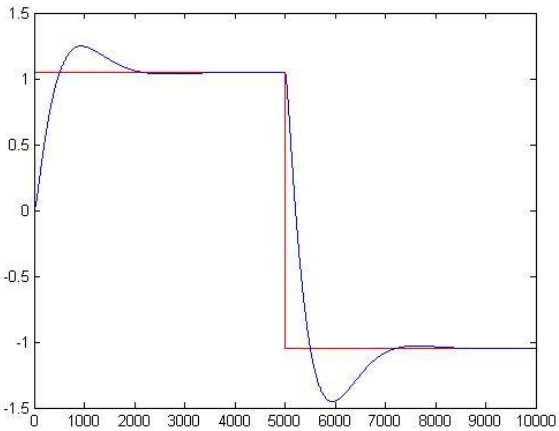
NIOS II processor architecture at SOPC Builder

NIOS II Schematic and connections at Altera Quartus II

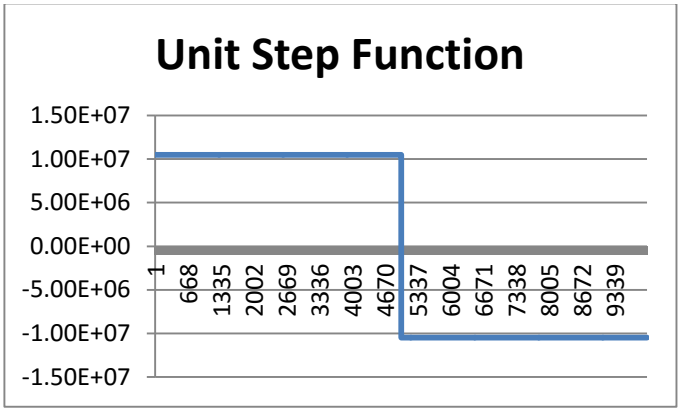


2nd experiment results

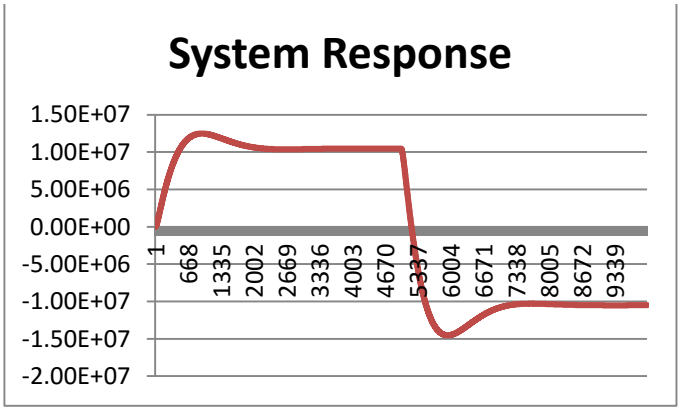
Matlab plot



Excel graphs from extracted values



Excel graphs from extracted values



CPU Benchmarks with different NIOS II implementations		
Processor	***PID time***	***Code time***
NIOS II/S Core 50Mhz	0.19 ms	6.1 ms
NIOS II/F Core 50Mhz	0.02 ms	4.2 ms

C. FOF – First Order Filter (Lowpass)

In this experiment we will investigate the response of a 1st order closed loop system in which through Matlab we create a signal with noise. In communication through UART(RS232) at the DE1 board and FPGA Cyclone II a C-code produced filter (White noise Filter) will return the signal at real time to the Matlab environment for plotting the response of the system.

Transfer function:

$$\boldsymbol{Y(s) = \frac{K}{S+K} X(s)}$$

$$\boldsymbol{SY(s) + KY(s) = KX(s)}$$

Laplace transformation from the field of frequency to the field of time:

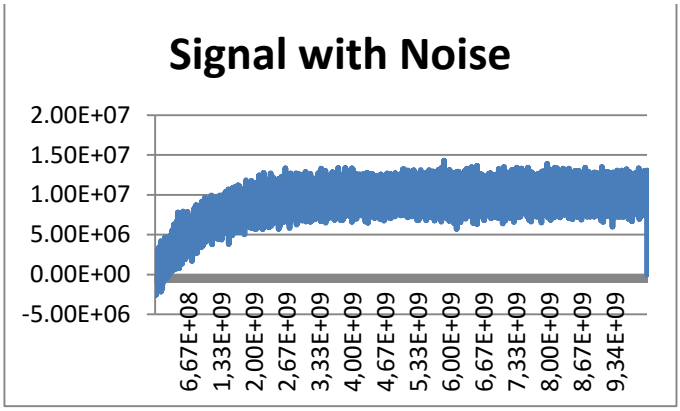
$$\boldsymbol{\frac{dy(t)}{dt} = -Ky(t) + Kx(t)}$$

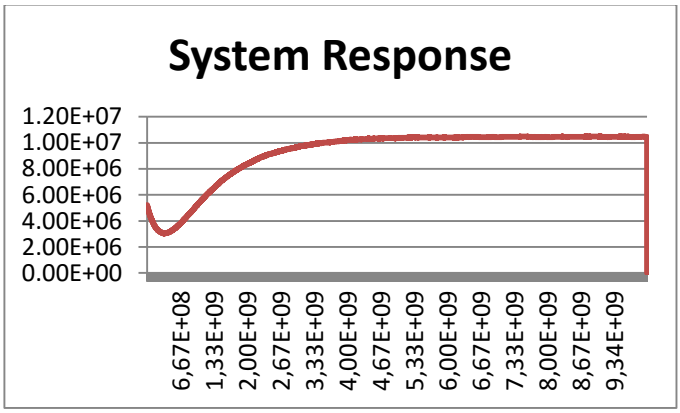
Sampling:

$$\boldsymbol{y[n+1] = (1 - tsK)y[n] + tsKx[n]}$$

$$\boldsymbol{y[n+1] = ay[n] + bx[n]\ ,\ a = 1 - tsK\ ,\ b = tsK, K = \frac{b}{ts}}$$

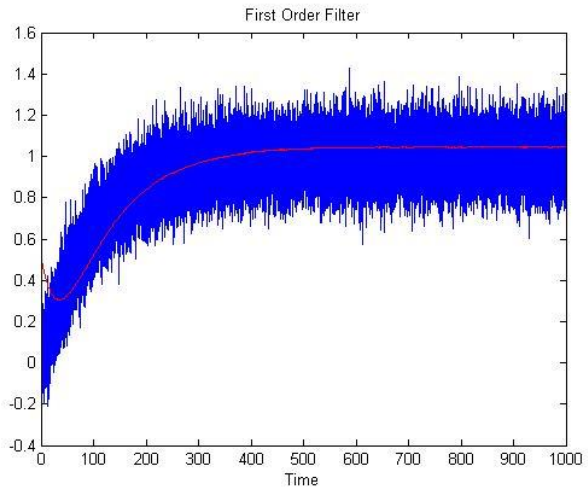
$$\boldsymbol{y = aybx}$$





3rd experiment results

Matlab plot



CONCLUSION

Through observations and benchmarks we can conclude that the systems responses and PID computations are operated with great speed and flexibility for the reason that all systems are hardware based to the FPGA logic.

REFERENCES

[1] Rapid Prototyping of Digital Systems, Second Edition

[2] Introduction to Quartus

[3] PID Controller Design Approaches – Theory, Tuning and Application to Frontier

[4] Modern Control Technology - Components and Systems

[5] Practical PID Control

[6] Modern Control Systems

[7] Quartus Reference Manual Vol. 1 , 2, 3

[8] Embedded Control Systems in C/C++: An Introduction for Software Developers Using MATLAB

[9] Embedded SOPC Design with NIOS II processor and verilog examples

[10] Modern Control Technology – Components and Systems