FPGA CONTROLLER DESIGN AND SIMULATION OF A PORTABLE DOUGH MIXING MACHINE

Joseph O. Inwelegbu and **Thomas A. Nwodoh**

Department of Electrical Engineering, Faculty of Engineering, University of Nigeria, Nsukka, Nigeria

Abstract

Biscuit and chocolate cookies are normally produced by industrial based processes and machinery using analog and IC gate-based Controllers. With the advent of Microcontrollers, Application Specific Integrated Circuits, Digital Signal Processors (DSP) and Programmable Logic Devices, complex industrial systems and controls can now be integrated into portable embedded household electronic systems. In this paper, the design and simulation of a Dough Mixer Controller (DMC) with Proportional Integral Derivative (PID) closed-loop motor feedback control, is undertaken for a home-based biscuit Cookie machine. DC motor behaviour is modelled and simulated with Matlab/Simulink. Synthesizable VHDL Design and Simulation is carried out with Quartus II Web-Edition 9.0 and ModelSIM EDA software tools, using an Altera Field Programmable Gate Array (FPGA) development system to verify the PID algorithm applied. Simulation results show that the PID algorithm programmed into the FPGA-based controller, effectively maintained the Permanent Magnet (PM) DC motor of the mixer at constant torque over an operational 3-speed range. The mixer agitated the 10g dough ingredients fed into the mixing compartment. Subsequently, the mixture resulted in the formation of a proteinous and coherent viscoelastic dough structure, consistent with published works on dough mix-texture. The dough can subsequently be baked into biscuit cookies.

Keywords: FPGA, VHDL, PID controller, Pulse Width Modulation, Full H-Bridge DC motor driver

1. Introduction

The DC motor, a power actuator is widely used in industrial applications. The speed of a DC motor can be adjusted with electronic controllers to a great extent, so as to provide easy control and high performance [1,2]. By means of various combinations of shunt, series, and separately-excited field windings, DC motors (DCM) can be designed to display a wide variety of volt-ampere or speed-torque control characteristic; hence they have found use in many applications [3,4].

In general, electronic controllers can be classified as analog and digital. Analog controllers which only presented low cost and sim-

plicity of use characterized earlier controllers. With rapid advances in control technology, more and more research on digital controllers with computer-based systems have been done in recent years [5,6,7]. Usually, when a Microprocessor or Digital Signal Processor (DSP) is used for digital control, the control algorithm is executed sequentially. Migrating to programmable logic control, offers high speed, parallel processing, concurrent processing and short development time, resulting in a fast time to market.

FPGAs consist of three major configurable elements, namely: Configurable Logic Blocks (CLBs) that provide the functional elements; Input/Output (I/O) Blocks (IOBs) that provide interface between the package pins and internal signal lines; and Programmable interconnect resources that connect I/O of CLBs and IOBs onto the appropriate network. FPGA based digital controllers have become the most favorable choice for prototyping digital systems. The control algorithms are developed in VHDL which is now one of the most popular standard digital logic Hardware Description Languages (HDLs). It is supported by all major Computer Aided Engineering (CAE) platforms, making it very versatile [8,9,10].

FPGA advantage lies in customizing previously fixed generic hardware in Microcontroller units (MCUs) or DSP chips. Hence, the optimized application-specific PWM block in the DMC of this paper, can replace the standard PWM block found in a MCU or DSP-based motor-control chip, and reduce the Total Harmonic Distortion (THD) due to the motor, by nearly 50% at a high modulation index [11]. Here, an Altera FPGA controller is designed and programmed to control the speed of the Permanent Magnet (PM) DCM of a portable dough mixing machine, using the Proportional Integral Derivative (PID) algorithm with Pulse Width Modulated (PWM) signals.

2. Relevant Literature

FPGA-based PID control scheme have been widely applied to DCM control applications by many researchers. Abdelati presented the implementation of PID controller for a DCM on an FPGA board. Several modules necessary for building PID controllers on FPGAs which improve speed, accuracy, power, compactness, and cost effectiveness were outlined in the work [12,18].

Li et al implemented a parallel PID algorithm with fuzzy gain conditioner on an FPGA and conducted a simulation-based study [21]. Chen et al implemented a complete wheelchair controller on FPGA with parallel PID algorithm [13]. Correia et al also presented a standard DCM speed control design for a test car [14]. The LED, display and pushbutton modules were automatically generated by an EDA system. The implemented platform permits: (a) to control directly the real vehicle using control commands that are sent using a keyboard and (b) to simulate the control process in a virtual environment, using a virtual instrumentation approach.

Gras et al reported the development of a prototype Micro Z-arm Mixer for measuring flour [17]. The instrument consists of a temperature controlled mixing bowl and a speed-controlled DCM. Results obtained with the mixer were highly correlated with those obtained with the conventional equipment.

Mei et al reported the development of a real time simulation system for modeling electric motors based on a FPGA chip (Altera FLEX 10KEPF10K70). The internal design of the FPGA was carried out using a combination of VHDL and schematic/graphical entry methods [14].

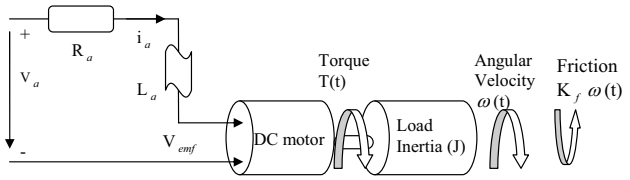

Figure 1: General model of a DCM.

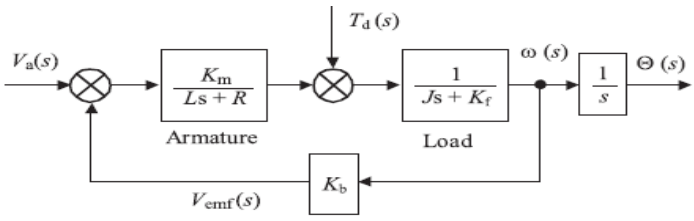

Figure 2: Mathematical model of a DCM.

Kowalski et al presented the position control of an unmanned electrical dual rotor helicopter. The experimental tests were performed with the National Instruments industrial computer with RIO PXI- 783 1R card using LABVIEW software. The rotor speed was measured by an encoder for comparison with the estimated speed. A prototyping board is the central piece with a PC and motor drive circuit attached [36].

2.1. Mathematical model of a DCM

The general model of the DCM is depicted in Figure 1. The applied voltage V(t) controls the angular velocity (t). The relations for the armature controlled DC motor are shown schematically in Figure 2.

The dynamical model of the DCM is given by the following equations [15],

$$V_a(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + V_e mf(t) \quad (1)$$

$$T_m = J \frac{d\omega_m(t)}{dt} + B\omega_m(t) + T(t) \quad (2)$$

Where the parameters and variables are defined in Table 1 following.

Table 1: Motor Parameters.

V_a	Applied armature/input voltage (volts)
L_a	Armature inductance (H)
I_a	Armature current (A)
R_a	Armature resistance (Ω)
K_b	Back emf constant (volt.sec/rad)
K_f	Motor constant (kg m)
ω	Motor shaft angular velocity (rad/s)
T_m	Output torque (Nm)
I_m	Moment of inertia of motor & load (kgm)
B_m	Vicious friction constant of load and motor (Nm.sec)
J_m	Moment of inertia of load & rotor (kg m)

The output torque T_m of the motor is proportional to the armature current i_a, i.e. $T_m = K_t i_a$. By omitting the static friction, if one regards the applied armature voltage $V_a(t)$ as the input and the angular velocity $\omega_o(t)$ of the motor shaft as the output and also assume $L_a I_m \approx 0$, the transfer function from (1) and (2), is given as [15]:

$$G(s) = \frac{K\omega_o}{s + \omega_o} \quad (3)$$

DC Motor at Constant Torque Operation

Many industrial applications such as conveyors, mixers, squeeze rolls, processing machinery, etc., require nearly constant torque over their operating speed range [16]. A PM DCM was chosen for this research, because they exhibit an approximate constant torque characteristic over their speed range as shown in Figure 3, to deliver a constant torque to the agitator over varying speeds for uniform mixing of biscuit dough.

PWM Control

The Control of DCM and power electronics by PWM signal is very well known [17,7]. In

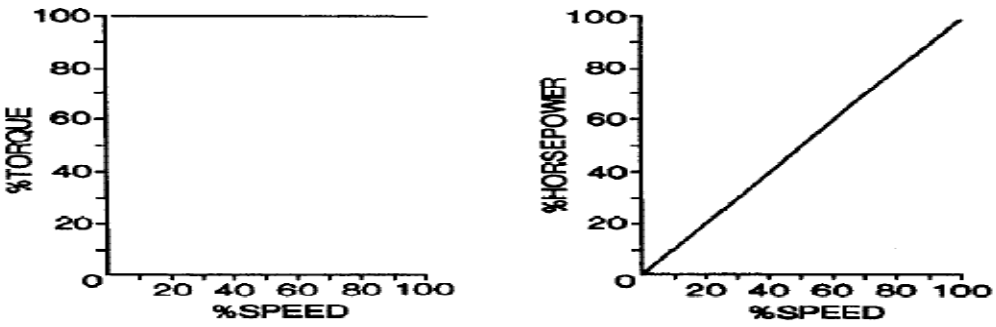

Figure 3: PM DCM Constant torque characteristics.

PWM, a time dependent varying output voltage is achieved in the process of varying the pulse by controlling the switching of the input voltage for the off and on duration. Figure 4 shows a square wave PWM pulse with 50% duty cycle.

As shown, if the input voltage V_{in} can be switched on and off frequently at the uniform rate then the total period T will be:

$$T = T_{on} + T_{off} \tag{4}$$

Where: T_{on} = ON time and T_{off} = OFF time. For a 50% duty cycle, the output voltage is 0.5 * V_{in}. In this design, the V_{in} = FPGA input voltage = 3V dc. This is boosted by the full H-bridge IC at 24V DC, to drive the mixer DCM. In general, the output voltage is:

$$\begin{aligned} V_{avg} &= \left\{ \frac{T_{on}}{T_{on} + T_{off}} \right\} * V_m \\ \rightarrow V_{avg} &= (D) * V_m \end{aligned} \tag{5}$$

Where: V_{avg} = average output voltage, and D = duty cycle. The PWM signal from the PWM block of FPGA controls on and off period of each terminal of the full H-Bridge transistor controlling the motor, and hence the speed.

Speed Control Methods of DC Motors

The speed of a motor can be controlled by open loop and closed loop control strategies. In the PID closed-loop control as applied in this research, the control value is dependent on the output speed of the motor. An open loop control does not. Figure 5, illustrates the losed-loop control of a DCM to maintain the controlled speed at the desired reference speed [18,19,20].

The control objective is to make the motor speed follow the reference input speed change by designing an appropriate controller. The goal is to eliminate the error between P the controlled variable (motor speed) and the desired speed P_d. The value of P is measured by the sensor, an optical speed encoder, which is compared with P_d to generate the error $e(t)$. The controlled output $u(t)$, is a function of $e(t)$. In this research, this is the PWM signal, which is fed to the full H-Bridge driver to power the mixer motor.

PID Control Algorithm

The PID algorithm has been demonstrated to be effective for DC servo motor speed control. The PID controller is used to reduce or eliminate the steady-state error between the measured motor speed and the reference speed to be tracked. It is therefore applied for control of the dough mixer PM DCM. The general equation of a PID controller, is the summation of the proportional, derivative and In-

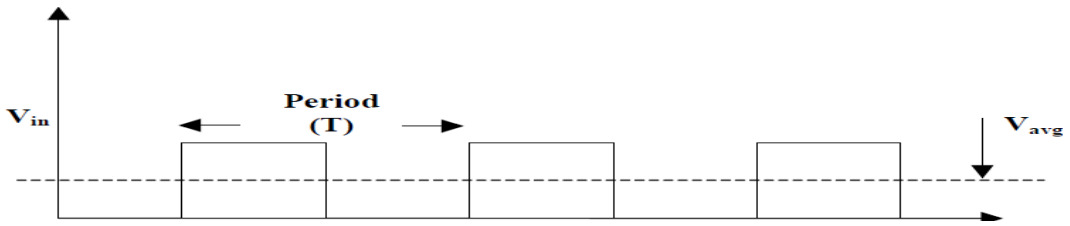

Figure 4: Square wave with 50% duty cycle.

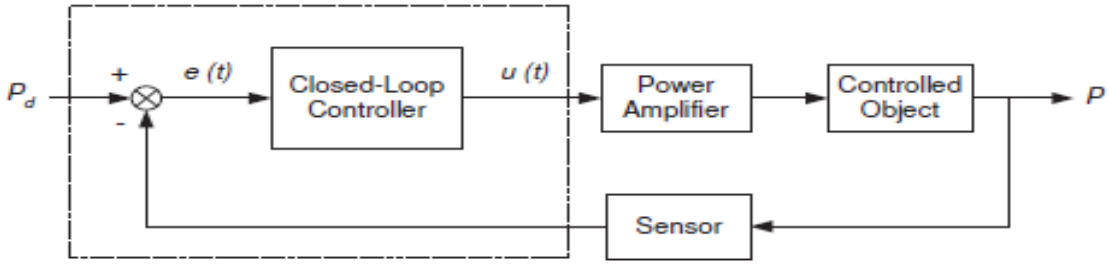

Figure 5: Closed-loop speed control system

tegral terms, giving $U(t)$ the output signal, as found in relevant literatures [21,22]. Consider the ideal PID controller written (assuming $U_o = 0$) in the continuous (analog) time domain form as:

$$U(t) = K_p e(t) + K_i \int_o^t e(t)dt + K_d \frac{de(t)}{dt} \quad (6)$$

Where: K_p = proportional gain; T_i = integral time; Td = derivative time; $e(t)$ = tracking error. There are several methods to calculate P, I and D terms. The references [22,23] provides the detail of all tuning methods, with their functional graphs. To design the digital PID controller in an MCU or FPGA device requires the standard PID controller with schematics in Figure 4, to be discretised. T_p, T_i, and T_d denote the time constants of the P, I and D terms.

To discretise the controller requires the integral and the derivative terms to approximate to forms suitable for computation by a computer. The transfer function of the system in

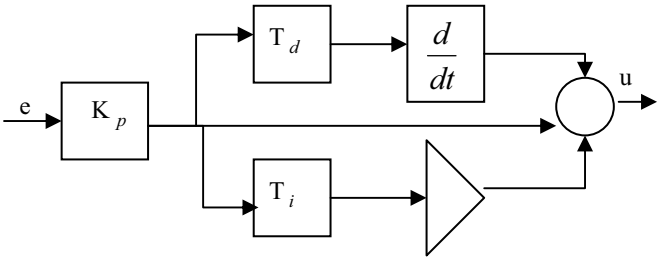

Figure 6: PID controller schematics.

Figure 6 is given as:

$$\frac{u}{e}(s) = H(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (7)$$

The discretized digital controller is given as a difference equation by [24]:

$$\begin{aligned} U(n) &= K_p e(n) + K_i \sum_{j=0}^{n} e(j) \\ &\quad + K_d(e(n) - e(n-1)) \end{aligned} \quad (8)$$

Where e = error, $K = \frac{K_p T}{T_i}$ is the integral coefficient, and K = is the derivative coeffi-

cient. To compute the sum, all past errors, $e(0) \ldots e(n)$, have to be stored. From equation (8), the digital controller $U(n)$ is given as:

$$K_p(e_n - e_{n-1}) + K_i e_n + K_d(e_n - 2e_{n-1} + e_{n-2}) \quad (9)$$

This form of the digitized PID equation will be used for implementation of the DMC.

PID Tuning Methods

Tuning is the process of calculating the P, I and D parameters for optimal gains to get an ideal response from a control system. The Trial and Error, Ziegler-Nichols, Cohen-Coon and Software tools, are Methods cited in several literatures to accomplish this [25]. Trial and Error Method was applied in this research.

Trial and Error Method

In this method, I and D terms are set to zero first and the proportional gain is increased until the output of the loop oscillates. Once the P and$I$ have been set to get the desired fast control system with minimal steady state error, the derivative term is increased until the loop is acceptably quick to its set point.

The Dough

Dough is a paste made out of cereals by mixing flour with a small amount of water and/or other liquid. It is precursor to making breads, pancakes, noodles, crusts, pastry, cookies and similar items [26]. Dough is usually a non-Newtonian and viscoeleastic material, exhibiting bimgham plastic properties[27]. Cookie dough refers to a blend of cookie ingredients which has been mixed into a malleable form, later to be baked to individual cookies [28]. Biscuit dough may be composed of ingredients listed in Table 2.

Dough can be mixed by an electric mixer [27]. Lindley has proposed a detailed review of mixing operations [29]. Biscuit cookies are made with the mixed dough by setting oven temperature of about 200°C and baking time of about 5 minutes, depending on the oven. The mixing time is related to the mixing index M, by the formula:

Table 2: Typical Biscuit dough ingredients.

Item	**% by weight**
Flour	45%
Soft white sugar	2.3%
Water	43%
Butter	2.6%
Eggs	3%
Baking powder	1%
Salt	1.1%
Oil	2%

$$\ln M = -Kt_m \quad (10)$$

(10) Where K = mixing rate constant, which varies with the type of mixer and the nature of the components, and t(sec.) = mixing time. The smaller the mixing index is, the better the mixing will be. The mixing rate constant K, depends on the characteristics of both the mixer and the liquids. The effect of the mixer characteristics on K is given by:

$$K \propto \frac{D^3 N}{D_T^2 Z} \quad (11)$$

Where D (metres) = the diameter of the agitator, N(rev/s) = the agitator speed, D_t(metres) = the vessel diameter and Z(metres) = the height of liquid.

3. Material use for implementation

The materials required to accomplish the design of the DMC Controller can be categorized into Software, hardware and the Mixer Compartment and Apparatus. They are described below.

3.0.1. Software:

(a.) FPGA design software Quartus II Web Edition/Ver. 4.0
(b.) The latest Quartus II service packs (For Cyclone II 2C35 FPGA FPGA)
(c.) A logic simulator (Model-SIM)/(Simulink)

3.0.2. Hardware:

(a.) DE2 FPGA development board from Altera
(b.) A computer system
(c.) 24V, DCM
(d.) (d.) L298 (46V, 4A) Full H-bridge motor driver IC
(e.) PC type ATX-300Watts power pack
(f.) Optical speed encoder
(g.) Dough mixer container and planetary mixing apparatus
(h.) Tachometer for measuring motor speed in RPM
(i.) Laboratory Microscope, for observing dough microstructure
(j.) Wattmeter, for measuring motor power consumption
(k.) Connectors and cables

3.0.3. Mixer Compartment and Apparatus

Aluminum alloy dough pan with non-stick coating is used. The dimensions of the pan are 5.29 X 5.29 X 5.49 (inches). The tapering and rounded corners of the pan produces an approximate volume of 2230 cm^3 [31].

4. Methodology

Figure 7 shows the design steps for any programmable logic device [32].

A detailed explanation of the design methodology steps can be found in relevant literatures [32].

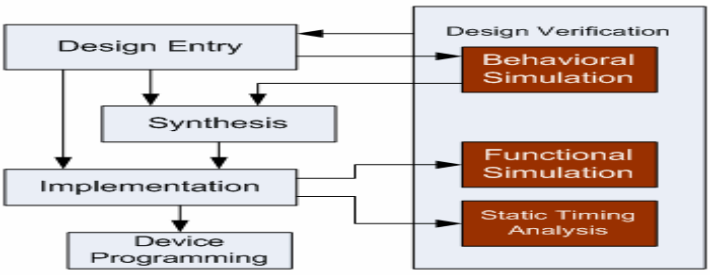

Figure 7: Programmable logic design methodology.

4.1. Hardware implementation

This section discusses the functional modules of the Dough-Mixer Controller shown in Figure 8.

DE2 FPGA development system

The Altera DE2 board block diagram shown in Figure 9 below features a state-of-the-art CycloneTM II EP2C35 FPGA in a 672-pin package [30]. All important components on the board are connected to pins of this device, allowing the user to control all aspects of the board's operation.

User Interface

In this research, through the 3-switch/key inbuilt in the DE2 system shown above, the user inputs an 8-bit value, that specifies the mixer motor LOW/MEDIUM/HIGH speeds of (100/200/300) RPM. This module also handles the bounce effect and associated scanning of the switches and the multiplexed 7-segment display units [18].

Optical speed Encoder function

The speed output of the motor is mechanically coupled through an optical encoder. The voltage encoded speed output (V_f), is fed back to FPGA through the ADC block. The reference input (V_r) from the speed switch, is then compared with V_f. The difference (e_f) (error signal), which is the control signal effects speed adjustment of the PM DCM through

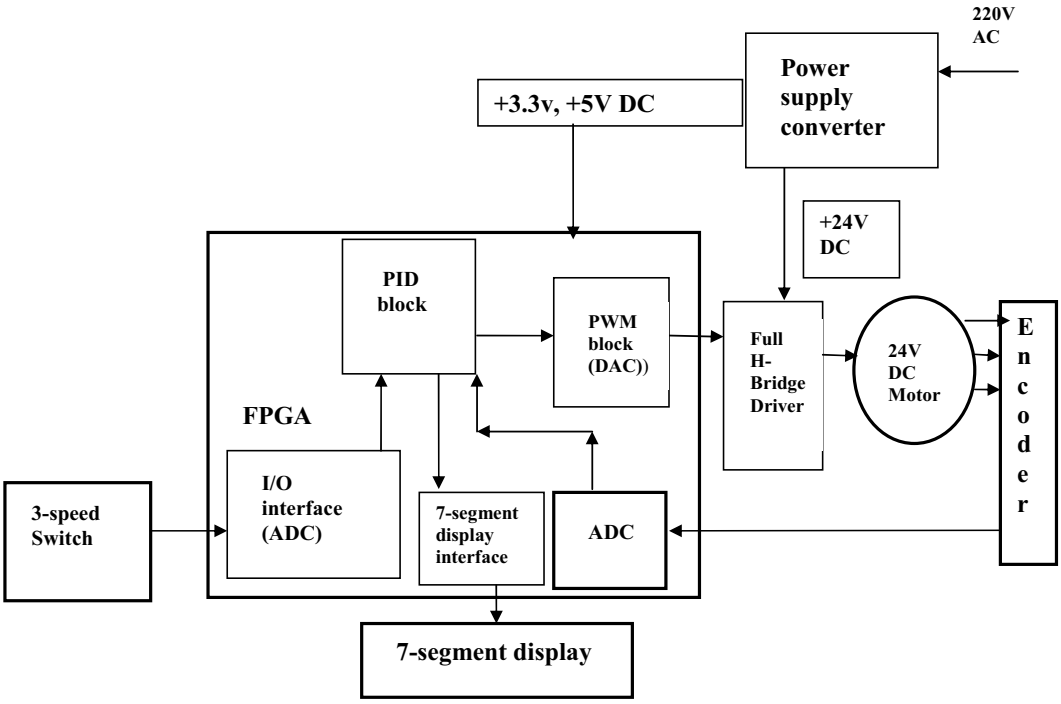

Figure 8: DMC Architecture and functional modules.

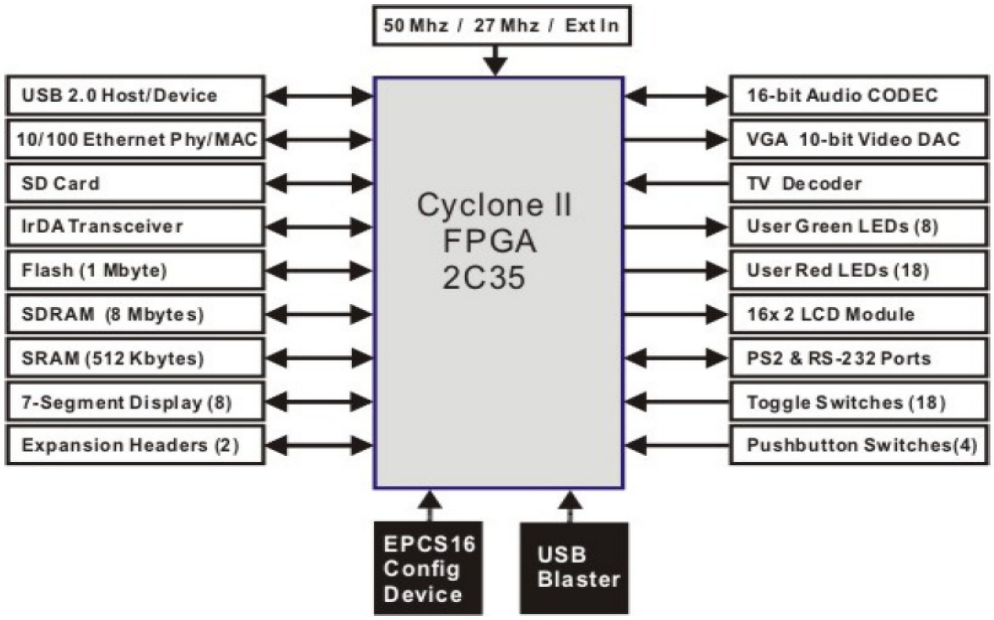

Figure 9: Altera DE2 FPGA development system.

PWM and PID algorithm. Speed control is achieved when the sensed speed equals reference speed and error signal (e) given by equation (12) equals zero.

$$e_f = V_r - V_f \tag{12}$$

In Figure 10(a) and (b), a rotating slotted disk was mounted on motor shaft with a fixed 10 tracks, slotted disk. The frequency of the output waveform f_out is given by:

$$f_{out} = \frac{N\ rpm}{60} \tag{13}$$

Where rpm = speed in revolutions per minute, and N = number of slots in disc. So, from equation (13), the maximum speed of the PM DCM is given by,

$$rpm = \frac{f_{out}60}{N} = \frac{50 * 60}{10} = 300RPM$$

From Figure 10(b), Chip LM324 is used to convert the output square pulses to digital form, readable by ADC block of FPGA. When the V_{out} of photodiode is less than V_{ref}, the output of LM324 will be 0V (Low) and 5V (High), if vice-versa. The output signal from LM324 has a frequency given by equation (13), and is read by the ADC block of the FPGA, as a representation of the actual switch programmed speed.

PM DC Motor (24V)

The mixer motor and dough container used is same as that used in [31]. Motor and Pulley specifications : ECM Motor CA-161200-T, 01R06, 24 V DC, 30 W.

Motor Shaft: 2275 RPM Full speed, 1800 RPM Pulse Speed

Pulley Ratio: (130 teeth on kneading blade shaft / 20 teeth on motor shaft) = 6.5 turn down ratio

Mixer Blade Rotation: 350 RPM, Maximum PWM Speed

Motor maximum Load = 0.58 Amps, 67 Watts.

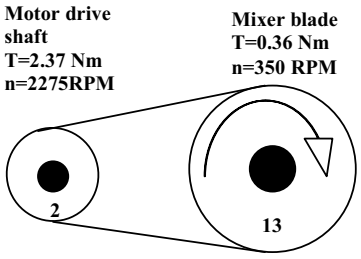

Figure 11: DCM drive shaft and Mixer agitator shaft and pulley system.

PWM Block PWM systems control direction, speed, and average torque of a motor. The PWM block of the FPGA interfaces to the motor through the H-bridge driver. It sends speed modulated signals to control duty cycle of full H-bridge DCM driver, through MOSFETs (IRFZ 740). Power is supplied to the motor in square wave of constant voltage but varying pulse-width, determined by equation (4). The speed of a DCM is a function of the input power and drive characteristics. The modulating signal is supplied in 8-bit digital format. With the 50MHz synchronization clock, a minimum pulse width of 100 ns for the synchronization clock period is obtained. A PWM frequency of about 50 Hz is found suitable. To generate this, the modulating signal is multiplied by 2^12 and a 20-bit counter with 50MHz clock is fired. This counter is compared with the scaled modulating signal. If the scaled modulating signal is larger, the PWM output is set to one, otherwise, it is set to zero, thus the PWM algorithm flowchart shown in Figure 12, is implemented successfully.

Mixing Apparatus

Dough and paste are mixed in machines, which of necessity must be powerful; based on the size of the materials to be mixed to form the dough. This research is based on mixing about 10g weight of material as shown in Table 2 to form dough for the biscuit cookie;

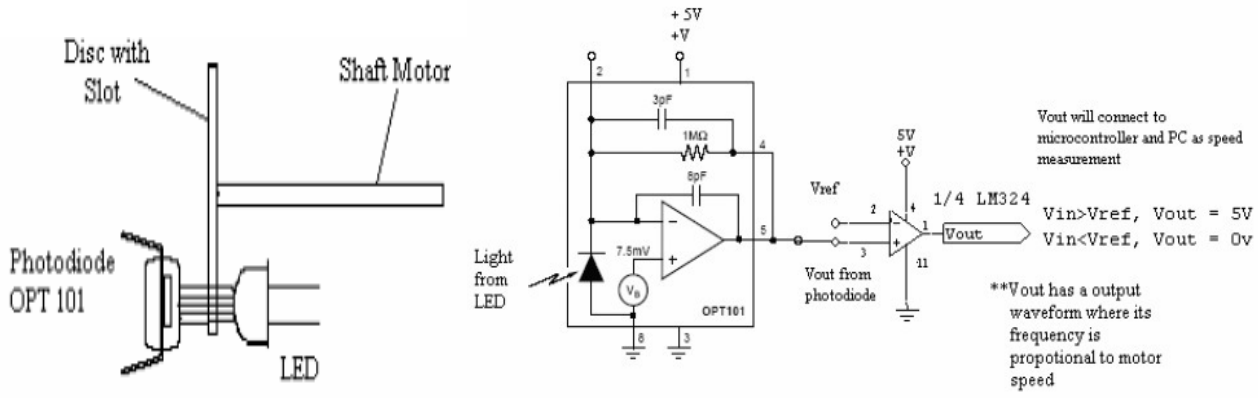

Figure 10: (a) Basic configuration (b) Schematic circuit.

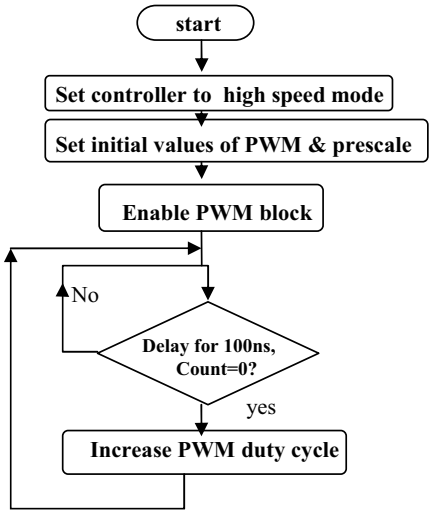

Figure 12: PWM flowchart.

therefore a small mixer with planetary agitator is adequate.

4.2. Software implementation

The controller software embedded within the target Cyclone®II 2C35 FPGA running at 50 MHz clock calculates the necessary duty cycle and generates PWM signals for the full H-bridge driver through the PWM block. Using the PID algorithm (shown in Figure 13) implemented in the PID block, the DCM speed is maintained at the required constant value. Figure 14, shows the DMC system control blocks, Figure 15, shows the digital PID controller equation block. Figure 16 shows the hierarchical diagram of the PID controller Implementation. Quartus II Ver. 4.0 and ModelSim XE III 6.3c software tools were used for building and testing these modules.

PID Algorithm Description

The PID block is implemented in VHDL language using dedicated libraries from Altera. The flowchart of the algorithm is shown in Figure 13.

Software Implementation of the PID Algorithm

VHDL software modules used include the key scanning and 7-segment multiplexed display routines, PWM block and the PID block and the system reset. To resolve the digital PID equation (8) of the PID block, VHDL libraries built for algebraic manipulation in VHDL are used. Several works aided this research [33, 34]. A digital PID controller can be represented by the following expression [33]:

$$\begin{aligned} u[k] &= u[k-1] + e[k](k_p + k_d + k_i) \\ &\quad + e[k-1](k_i - k_p - 2k_d) \\ &\quad + e[k-2]k_d \end{aligned} \tag{14}$$

This is another form of equation (8). Equation (14) is represented in Figure 15. The register error block stores values of $e[k]$, $e[k-1]$ and $e[k-2]$ (error), and makes shift operations

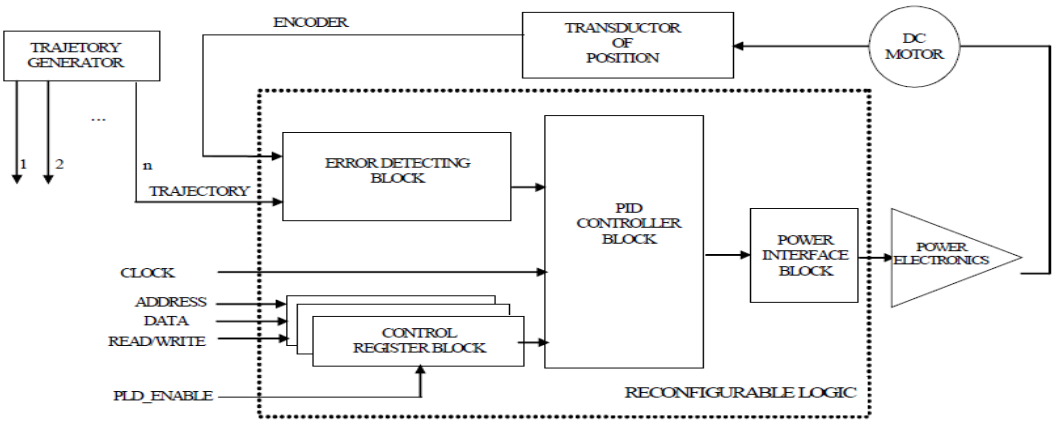

Figure 14: DMC system control blocks.

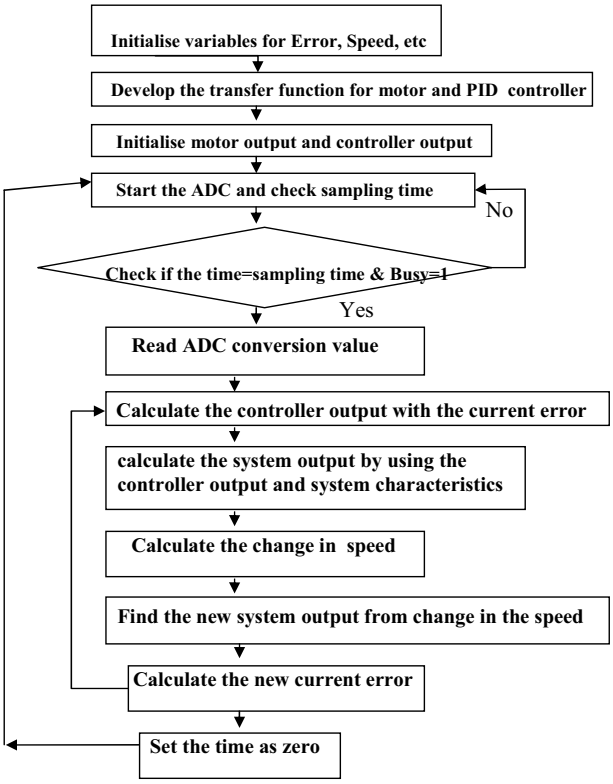

Figure 13: Flowchart for PID algorithm.

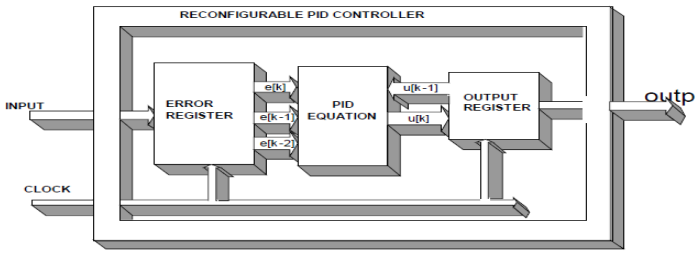

Figure 15: Digital PID controller equation block implementation in programmable logic.

$(e[k-1] = e[k]$ and $e[k-2] = e[k-1])$. The output register block stores $u[k]$ and $u[k-1]$. The implemented controller can be visualized in the blocks presented in Figure 14. Four main blocks are observed:

- Error Detecting Block: This block is used for the comparison of the signs TRAJECTORY (user motor speed, through 3-switches) and ENCODER (Feedback speed read from optical encoder). (+ or sign).
- PID Controller Block: it implements a PID digital controller, using the gain parameters contained in the control registers.

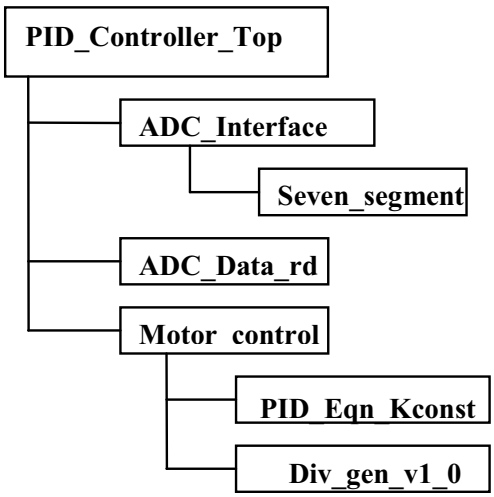

Figure 16: Hierarchical Diagram of PID controller Implementation.

- Control Register Block: it implements the control registers, responsible for the programming of parameters in the FPGA.

- Power Interface Block: it converts the binary word supplied by PID controller in a pattern of digital signs to control the PWM block [36].

4.2.1. Hierarchical structure of the PID controller Software

The hierarchical structure of the PID controller software implementation is shown in Figure 16. It consists of a top level module called PID _ controller _ Top, with functional sub-modules.

PID _ Controller _ Top Module
The PID Controller Top module, instantiates the sub modules ADC _ interface, ADC _ Data _ Read and Motor _ control. It interconnects all the signals and interacts with the external world.

Coregen Divider Module
This is Xilinx/Altera specific module Div _ gen _ v1 _ 0, used in the present design. Instantiated in Motor Control Module to divide the calculated PID value Vn to get the equivalent binary value, which has to be sent to DAC. **PID Equation Calculation Module** The calculated errors en, en-1 and en-2 with their polarities, is fed to this module from motor control module. This module calculates the PID equation,

$$K_p(e_n - e_{n-1}) + K_i e_n + K_d(e_n - 2e_{n-1} + e_{n-2}) \quad (15)$$

(15) This is a form of the discretized digital controller in equation (8). The constants K_p, K_i and K values are 3, 2 and 1 respectively, calculated by trial and error method of PID described in section (1.6.1).

5. Simulation of the Controller

The complete design is simulated for verification using Modelsim 6.5 Simulation tool, which has pre-compiled libraries for all Altera FPGAs. The inputs like Clock, Reset, Switch data and ADC data are defined and the output can be observed in the simulation window. The sub modules are instantiated in Top module and internal signals are also observed in the waveform window of the simulator. Once all the signals are taken into the waveform window, the simulation is run for 1000 microsecond and the required changes in the signals are observed. Observations from the software simulation window for the set speed of 300 RPM shows that after certain transitions, when the errors en, en-1 and en-2 become zero, the current speed locks at 300 RPM, which is equal to set speed through application of the PID control algorithm. Table 3, shows the Design Summary, Altera tool device utilization summary and reports the percentage of available resources used for the current FPGA design.

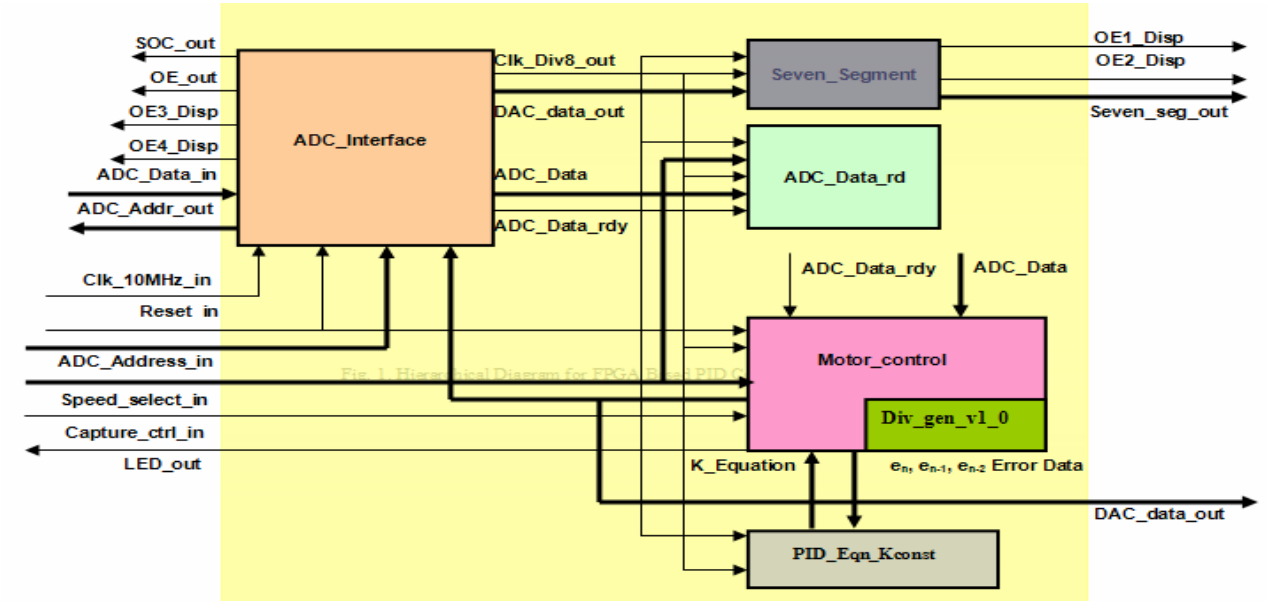

Figure 17: PID Controller Top module with sub modules.

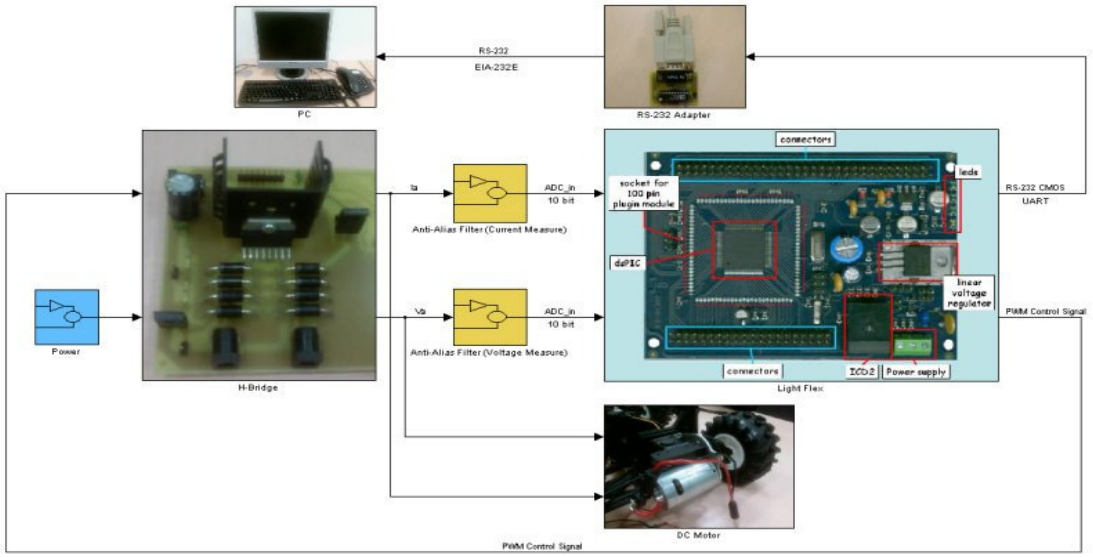

Figure 18: DMC - FPGA experimental setup. Courtesy: http://dev.emcelettronica.com/print/51811.

Table 3: Synthesis report of the controller (Cyclone^{TM} II EP2C35 FPGA logic elements utilization).

Device	**Spec. of the Cyclone^{TM} II EP2C35 FPGA**	**Used by this design**
Number of CLBs used	1536	757
Number 4- input LUTs	6144	2648
Number of IOBs	142	39
Number of gates used for design	300000	24585

6. Results and Discussions

6.1. Experimental setup

Figure 18 in the next page, shows the experimental setup. The HP laptop PC running SIMULINK, Quartus II and ModelSIM softwares is connected to the DE2 board, through a serial (RS-232) interface. The input interfaces to the DE2 development system includes the optical speed encoder circuit, the mixer 3-speed toggle switch and the power supply pack, providing +5V DC for logic, +3V DC for the FPGA and +24V DC for the Mixer motor and H-bridge driver. The DE2 output interface includes the 3-element seven-segment display and the full H-Bridge motor driver, activated by the controllers PWM block.

Dough Mixing Results

Approximately 10 grams of dough ingredients listed in Table 1 was placed within the mixer chamber. While mixing, torque responses and shaft speeds were collected with tachometer at mixing speeds of 300, 200 and 100 RPM, representing the 3-speed range of the controller. For each scenario, torque responses and agitator speeds were collected over a 10-minute interval. Results show that typical dough development times at 100 RPM (LOW-speed) are about 12 minutes, 8 minutes at 200 RPM (MED-speed) and about 5 minutes at 300 RPM (HIGH-speed), averaging optimum dough development time at about 8 minutes. This result agrees fairly well with reviewed literatures, using horizontal and planetary mixers [35].

Dough Tensile Properties

The tensile properties of the dough also changed systematically as the dough develops. The maximum tensile force that the dough exerts on the agitator as it adheres to the dough surface gets smaller. This is a combination of surface thickness and dough's cohesive properties. This is supported by results of fundamental rheological measurements [35], and also observed from the mixing curve shown in Figure 19.

Mixer power consumption profile

The dough mixing curve also shows that the motor power consumption reduced (measurements with a wattmeter) after about 6 minutes, typical of dough when the tensile force exerted on the agitator declined as dough complete development time was approached.

Dough Microscopy

Dough structure was observed under a microscope. The microstructure of the dough samples mixed at 100 RPM was quite different from the microstructure of dough mixed at 300 RPM. The 100 RPM set was less developed with coarser protein network, than the dough developed at 300 RPM. Indeed by comparison, the general effect of higher mixing speed is a finer, more homogenous protein network [35].

6.2. Experimental results

Experiment is conducted to verify the conventional PID control of the PM DCM of the DMC for speed control. Trial and Error Method was used to tune the PID controller. Effective motor speed control was achieved with optimum PID coefficients of P=4, I=0.001 and D=4. In Figure 20, the result of the speed control experiment showed noise behaviour. This is due to the change in sign of error as the motor speed changes to gradually lock at 300 RPM when speed error approaches zero.

To verify the performance of the controller hardware design, the VHDL code (Bit file) was downloaded into the target FPGA device and the complete system was reset. The set speed of 300 RPM was assigned to switches and the capture control switch was enabled. ADC block reads the set speed and PID block calculates the equivalent PID value, which

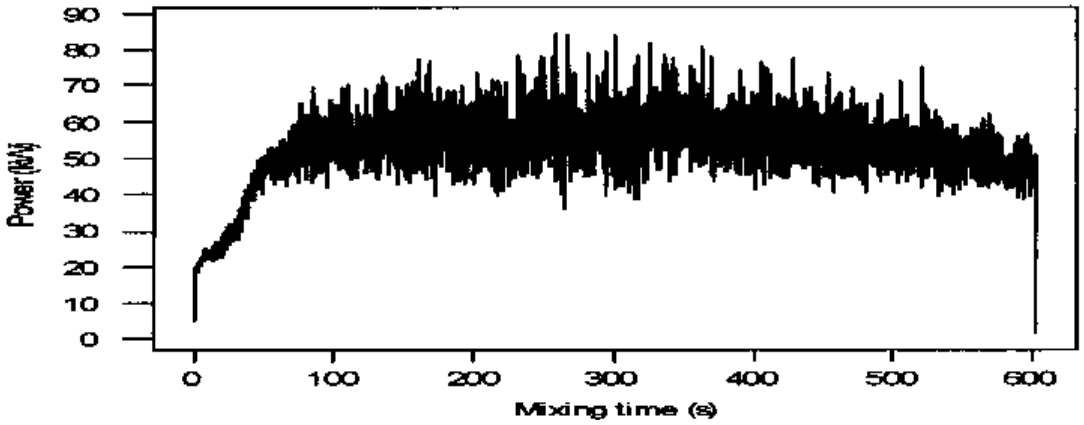

Figure 19: Dough Mixing curve.

feeds back to the motor through DAC block. The motor started running at set speed as observed from a tachometer when the current output speed equals the set speed. For another speed, the above procedure is repeated by changing the switch selection. Changing the speed, the ADC voltage also changed. Measured ADC and equivalent Hex values for different set speeds were tabulated in Table 4. It was observed that the current speed, which displays on the onboard 7- segment display, equals the set speed value. The change in the motor speed for different switch combinations can also be observed accordingly.

Table 4: Results of DCM speed control system for various set speeds.

SW. No	**Toggle Switch Position**	**Set Speed (rpm)**	**Equivalent HEX value**	**Measured ADC voltage(V)**
1.	000	300	12C	4.5
2.	001	200	C8	2.4
3.	010	100	64	0.64

7. Conclusions

A closed-loop PID algorithm, implemented on FPGA was proposed, designed and simulated. The performance is verified and tested. The results demonstrate that FPGAs are well suited for implementation of complex motor control and estimation algorithms due to their parallel and high speed execution characteristics. The VHDL software developed, contains a set of building blocks, each geared towards a specific algorithm. The test results also showed that with PID algorithm control, the steady-state error is eliminated and the desired output speed is obtained. The FPGA implementation reduced the total hardware complexity and cost. Simulation results show that when the speed is changed, the motor speed locks to the set speed, when the current error en, previous error en-1 and previous to the previous error en-2 becomes zero.

References

1. Henao H., Capolino G. A., Methodologie et application du diagnostic pour les systemes electriques, Article invit dans *Revue de l'Electricit et de l'Electronique (REE)*, No. 6, (in French), Jun. 2002, p. 79 86.

2. Raghavan S., *Digital Control for Speed and Position of a DC Motor*, MS Thesis, Texas A&M University, Kingsville, Aug. 2005.

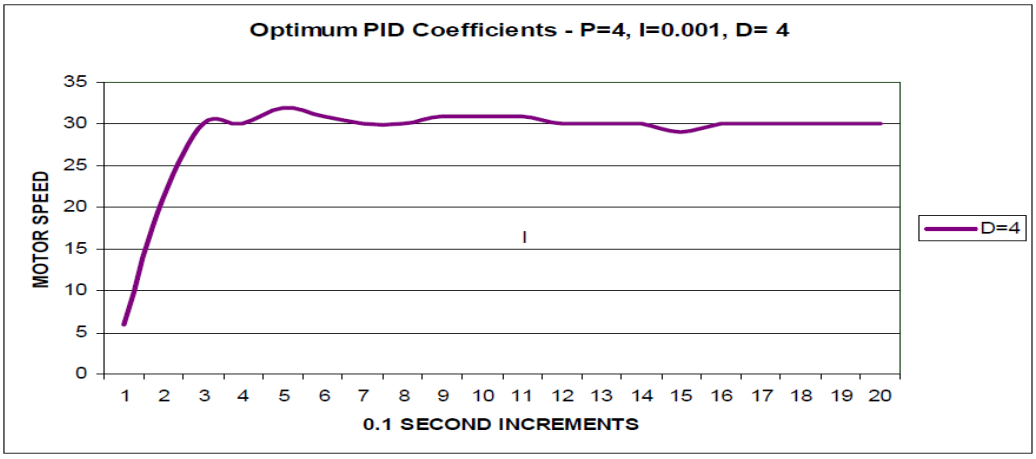

Figure 20: Experimental result of conventional PID control (motor speed scale = X 10 RPM).

3. M.S. Jr. Tomlinson, D.J. Walker, M.A. Sivilott. A digital neural network architecture for VLSI. *Proc. International Joint Conference on Neural Networks*, pp. 545-550, 1990.

4. Brown, S.D., Francis, R.J., Rose, J., and Vranesic, Z.G. *Field-Programmable Gate Arrays*. Kluwer Academic Publishers,1996.

5. Dinavahi, V.R.; Reza Iravani, M.; Bonert, R. Real-time digital simulation of power electronic apparatus interfaced with digital controllers. *, IEEE Trans. Power Delivery*, Volume: 16 Issue: 4, Oct. 2001 Page(s): 775-781.

6. A. de Castro, P. Zumel, O. Garcia, T. Riesgo, J. Uceda. Concurrent and simple digital controller of an AC/DC converter with power factor correction based on an FPGA. *IEEE Trans. Power Electronics*. Vol. 18 pp. 334 343, 2003.

7. Shih-Liang Jung, Meng-Yueh Chang, Jin-Yi Jyang, Li-Chia Yeh, Ying-Yu Tzou. Design and implementation of an FPGA-based control IC for AC-voltage regulation. *IEEE Trans. Power Electronics*. Vol. 14 pp. 522 532, 1999.

8. http://www.ijetch.org/papers/90New.pdf

9. S. Nerto, S.Bolognani, M. Ceschia, A. Paccagnella, M. Zigliotto. FPGA-based random PWM with real-time dead time compensation. *Proc. IEEE PESC03*, Vol 2, pp.15-19, (2003).

10. A. Fratta, G.Griffero and S. Nieddu. Comparative Analysis among DSP and FPGA-based Control Capabilities in PWM Power Converters. *The 30th Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2004.

11. Monmasson E., Cirstea, M. FPGA Design Methodology for Industrial Control Systems – A review. *IEEE Trans. on Industrial Electronics*, Vol. 54, No. 4, August 2007.

12. www.iugaza.edu.ps/.../volume%2014-%20Issue%201%20-studies%20-7.pdf

13. R.-X. Chen, L.-G. Chen, and L. Chen. System Design Consideration for Digital Wheelchair Controller. In *IEEE Transactions on Industrial Electronics*, Vol.47, No.4, Aug. 2000, pp. 898-907.

14. http://www.gigiroy.com/index_ figure/Control_ 2004_ finalised_ paper_ 084.pdf

15. DORF, R. C.BISHOP, R. H. *Modern Control Systems*, Addison-Wesley, New York,

1990.
P. Kachroo and P. Mellodge, Mobile Robotic Car Design, McGraw-Hill, New York, 2005.

16. http://www.usmotors.com/Products/dc_ hsc.pdf

17. GRAS, P. W. VARGA, J. RATH, C. TM-SKZI, S. FODOR, D. SALG, A. BKS, F. Screening for Improved Water Absorption and Mixing Properties Using Four Grams of Flour: A New Small-Scale Farinograph Type Mixer. In: *Proceeding of 11th International Cereal and Bread Congress*, Broadbeach, Qld, Australia (2000).

18. Mohamed Abdelati. FPGA-Based PID ControllerImplementation. The Islamic University Of Gaza.

19. Y. F. Chan, M. Moallem, W. Wang. Efficient implementation of PID control algorithm using FPGA technology. *Proceedings of the 43ed IEEE Conference on Decision and Control*, V5, PP. 4885-4890, Bahamas 2004.

20. J. Tang. PID controller using the TMS320C31 DSK with on-line parameter adjustment for real-time DCmotor speed and position control. *IEEE International Symposium on Industrial Electronics*, V2, PP 786-791, Pusan 2001.

21. Ang K., Chong G., Li Y. PID control system analysis, design, and technology. *IEEE Trans. Control System Technology*, vol. 13, p. 559 576, Jul. 2005.

22. http://www.Wikipedia.org/wiki/PID_ controller

23. Popov A., Farag A., Werner H. Tuning of a PID controller Using a Multi-objective Optimization Technique Applied to A Neutralization Plant. *44th IEEE Conference on Decision and Control, and the European Control Conference* 2005.

24. R. Isermann. *Digital Control Systems*. Springer-Verlag, 1989.

25. Lee, Y., Lee, J., Park, S. PID Controller Tuning for Integrating and Unstable Processes with Time Delay. *Chem. Eng. Sci.*,vol. 55, pp. 3481-3493, 2000.

26. http://www.wikipedia.com/dough.htm

27. http://www.correllconcepts.com/Encyclo-pizza/05_ Dough-making/_ 05_ dough-making.htm# _ Toc533730361

28. http://www.wikipedia.com/Cookie_ dough.htm

29. Lindley, J.A. Mixing processes for agricultural and food materials: Part 1. Fundamentals of Mixing, *Journal of Agricultural Engineering Research* 48: 153 - 170, 1991

30. www.ece.rochester.edu/courses/ECE112/...-/DE2_ UserManual.pdf

31. http://www.engr.uky.edu/ aseeched/papers-/2001/a0322.pdf

32. Modelsim Xilinx Edition, Xilinx Inc. 2005.

33. Samet, L. Masmoudi; N. Kharrat, M.W. Kamoun, L. A digital PID Controller for Real Time and Multi Loop Control: a Comparative Study. *IEEE International Conference on Electronics, Circuits and Systems*, pp. 291 - 296, vol.1, 1998.

34. Klotchkov, I. V.; Pedersen, S. *A Codesign Case Study: Implementing Arithmetic Functions in FPGAC's. IEEE* 1996.

35. http://ses.library.usyd.edu.au/bitstream/21-23/2569/1/QWCRC%20Report%202.PDF

36. http://www.insipub.com/ajbas/2009/3575-3596.pdf