Wrocław University of Science and Technology

Design and Simulation of a PI Controller for Speed Control of a Separately Excited DC Motor

---

Author:

David Ugochukwu Asogwa

January 2021

FACULTY OF ELECTRONICS, PHOTONICS AND MICROSYSTEMS

Class: INTERMEDIATE PROJECT

Lecturer: DR. ENGR. WITOLD PALUSZYNSKI

Abstract

The aim of this project is to design, and simulate using Multisim, a PI (Proportional Integral) controller for controlling the speed of a separately excited DC motor. A separately excited DC motor is one that has separate DC sources for the field and armature circuits. The PI controller takes in two variables, the set point (fixed) and the process variable (motor speed). In the simulation, the speed measurement of the DC motor is simulated with a variable DC source, referred to as process variable (PV), and the reference signal is a fixed DC voltage. The circuit parameters were calculated and standard values chosen. The working of the PI controller was tested with a triangular wave generator to observe it's characteristics when operating with a separately excited DC motor. This was carried out under three cases; motor running at constant load, motor running under reduced load and motor running under increased load.

INTRODUCTION

The proportional integral (PI) controller is a circuitry used in the speed control of DC (Direct Current) motors in industrial and laboratory applications. The device, in analog form, is designed using operational amplifiers and passive elements (resistor and capacitor). The controller (PI) consists of the error amplifier that takes in the error signal from the monitored parameter (speed), a set point signal which is the desired output, integral and proportional controller's and an inverting summer and an inverting amplifier at the output. The speed of the motor is monitored using a speed sensor that outputs a DC voltage which is sent to the error amplifier input and compared with the set point and the output is sent to the switching signals of the motor supply to either increase or decrease the voltage to the motor.

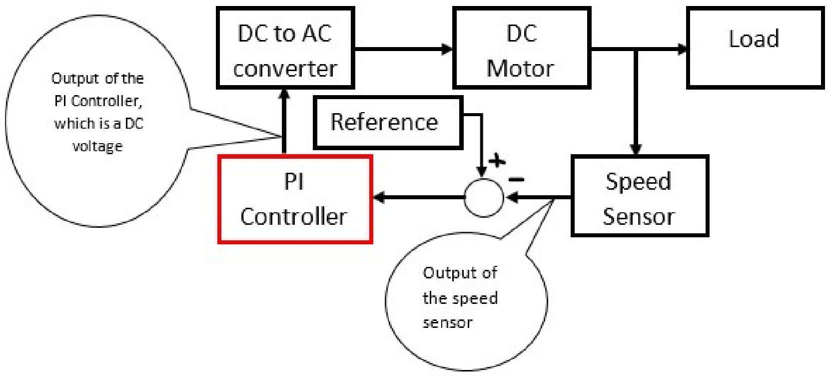

Fig. 1: PI Controller Function

INSTALLING NATIONAL INSTRUMENT (NI) MULTISIM

Multisim is an industry standard SPICE simulation and circuit design software for analog, digital, and power electronics in education and research. It integrates industry standard SPICE simulation with an interactive schematic environment to instantly visualize and analyse electronic circuit behaviour. Multisim has an intuitive interface that helps educators reinforce circuit theory and improve retention of theory throughout engineering curriculum. Researchers and designers use Multisim to reduce PCB prototype iterations and save development costs by adding powerful circuit simulation and analyses to the design flow.

The latest version Multisim 14.2 software can be downloaded from **https://www.ni.com**. The software have professional and educational versions, of which the licenses are to be purchased before activation. There is also an evaluation version of the NI Circuit Design Suite Education edition for students. This includes features from the Student Edition.

NI does not sell the Multisim Student Edition directly to students. The student must purchase the software online through one of our distributors like **Studica** (https://www.studica.com) at the price of **$54.00** and **$39.99** at **Digilent** (https://www.digilent.com), which is way cheaper.

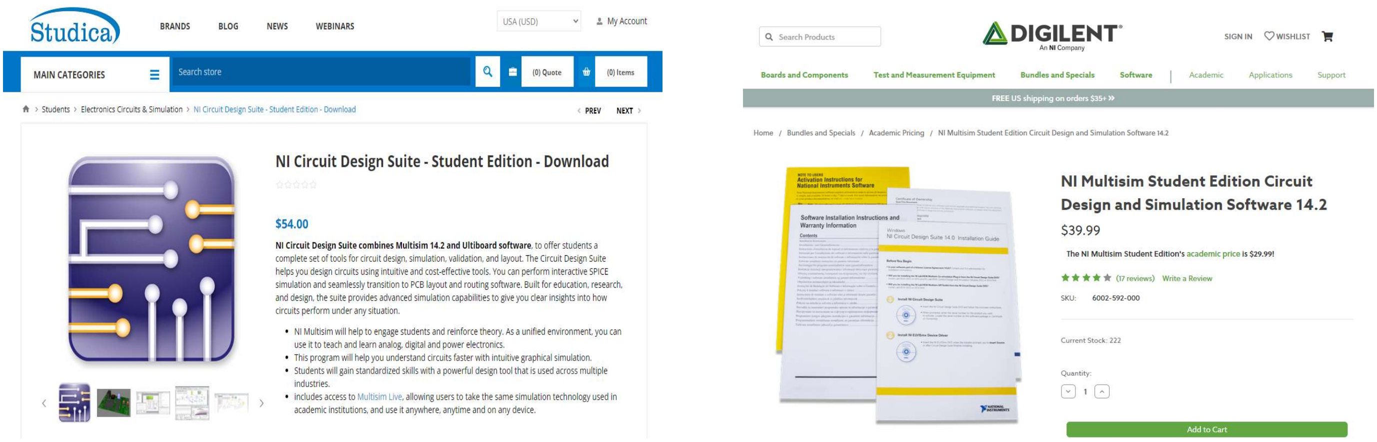

Fig. 2: Studica and Digilent price charts

Activation of the Student Edition is done like any other NI software, you may use the **NI Activation Wizard** to activate over the web when you launch the program.

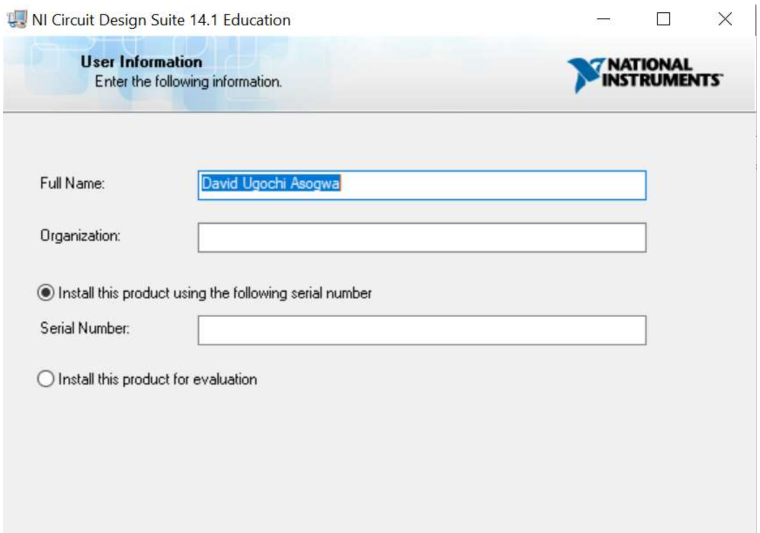

Fig. 3: Installation and Activation Interface

It should be noted that the Student Edition serial number can only be used to activate the Student Edition. An error message will be received when the Student Edition serial number is used to activate the Professional version.

The installations process is quite straight forward after accepting the terms and conditions. The interface for simulations is as shown below after the installation is completed.

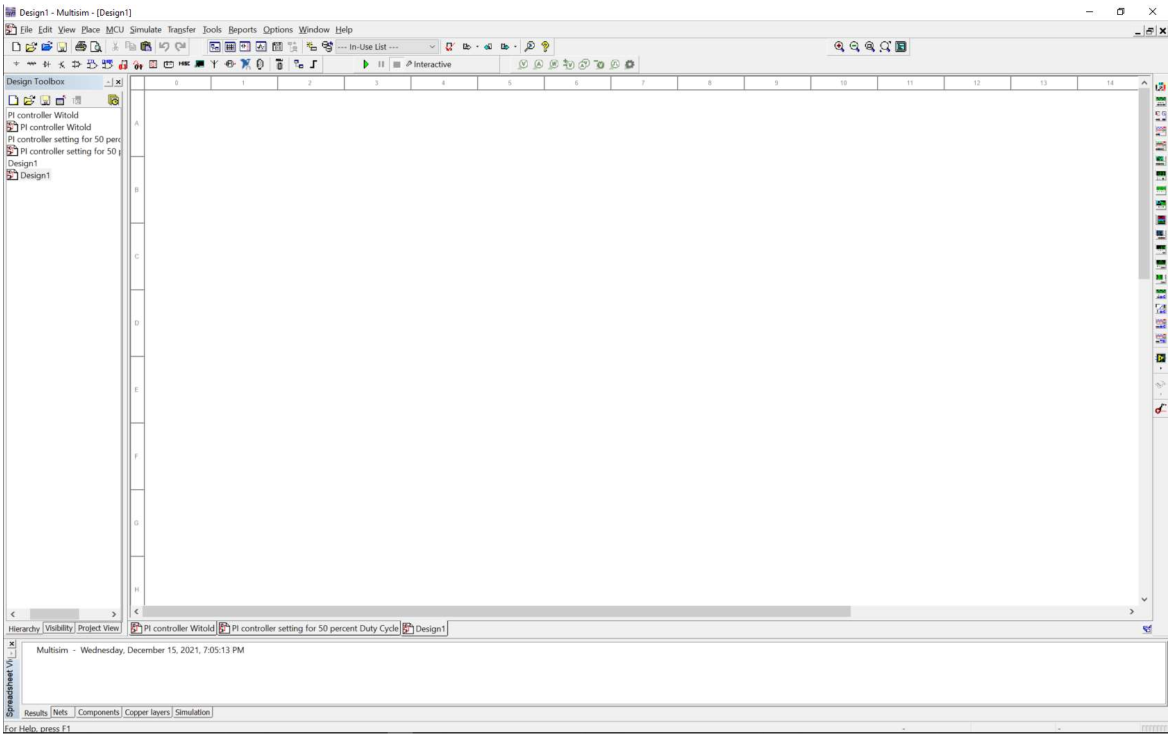

Fig. 4: Multisim Design User Interface

PI CONTROLLER DESIGN

All controllers must begin by generating an error signal which is the difference between the **set point** and the **actual value**.

$$\boldsymbol{Error = Set\ point - Actual\ value}$$

The PI controller is made of different operational amplifier (Op-Amp) configurations. These different configurations work together to achieve the desired aim. The different Op-Amp configurations include: **Buffer, Difference amplifier, Proportional Controller, Integrator, summer and inverter**. These different configurations are explained later in the page.

What is an Op-Amp: An operational amplifier (Op-Amp) is a DC-coupled high gain electronic voltage amplifier with a differential input and, usually, a single-ended output. In this configuration, an op amp produces an output potential (relative to circuit ground) that is typically 100,000 times larger than the potential difference between its input terminals. Operational amplifiers had their origins in analog computers, where they were used to perform mathematical operations in linear, non-linear, and frequency-dependent circuits, with the following characteristics:

i. No current flows into in or out of the op-amp inputs because of the high input resistance.

ii. The op-amp tries to keep the voltage at the input terminals at the same voltage.

iii. The op-amp gain is relatively huge and cannot be used without any external parameter to reduce the voltage gain.

V_{s+}: The DC positive power supply

V_{s-}:The DC negative power supply

V_{out}: Output Terminal

V_{+}: Non-inverting input

V_{-}: Inverting input

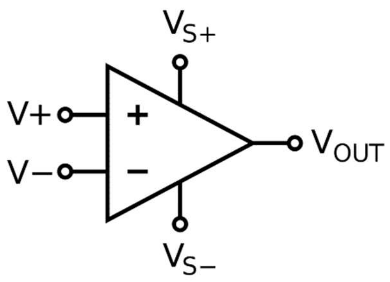

Fig. 5: Op-Amp Schematics

Difference Amplifier: The difference amplifier generates and amplifies the difference between the inverting and the non-inverting input.

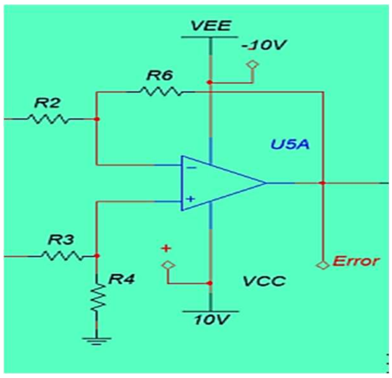

Fig. 6: Difference Amplifier with Parameters unknown

$$Error = V_{sp} - V_{pv}$$

$$V_o = V_{error} = \frac{R_6}{R_2}\left(V_{sp} - V_{pv}\right)$$

For this amplifier, the following condition ensues:

$$R = R_2 = R_6$$

All four resistances (R_2, R_3, R_4, R_6) must be carefully matched to prevent bias problems and offset problems, and enhance Common Mode Rejection Ratio when $V_{error} = 0$.

Buffer: This is a unity gain op-amp circuit added before the input terminals for the **set point value** and the **actual value**. This is to prevent overloading the feedback source and the set point voltage source. The buffer transfers voltage from a circuit with high impedance level to one with low impedance level and prevents the first circuit from interfering with the operation of the second signal.

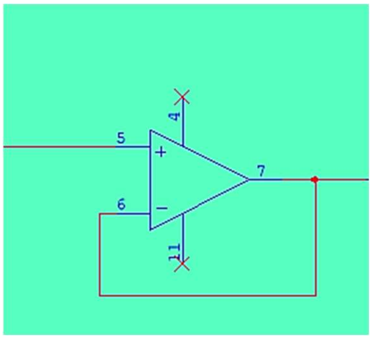

Fig. 7: Buffer Circuit

Proportional Controller: The proportional controller provides linear control in the circuit. Large negative error forces the proportional controller to go fully OFF and large positive error forces the controller to go fully ON. The controller is described by its proportionality band, that is, change in error that will cause the output to go from OFF to fully ON.

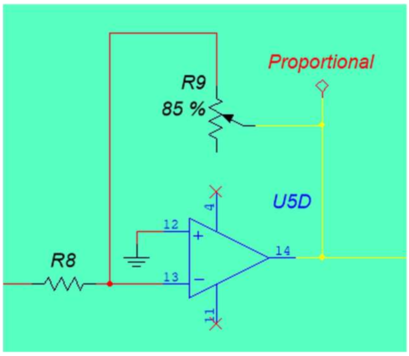

Fig. 8: Proportional Controller

$$V_o = \frac{R9}{R8} V_{error} + \frac{R9}{R_{os}} V_{os}$$

Usually, $R9 = R_{os}$, then we have: $V_o = \frac{R9}{R8} V_{error} + V_{os}$

With zero error, $V_{error} = 0$, and: $V_o = V_{os}$

V_{os} is often set the controller fill scale output. To reduce residual error, the gain of $U5$ is increased but care should be taken to avoid oscillations. In Laplace transform, the gain of the controller is given as:

$$K_p = \frac{V_o}{V_{error}} = \frac{R_9}{R_8}$$ K_p is the proportional gain of the controller.

Integrator: Also known as the **integral controller** has an output whose rate of change is proportional to the error. As long as there is any error at all, the output will continue to change until the error is driven to zero.

$$\frac{dV_o}{dt} = K_i V_{erro} \ ,$$ K_i is the integral constant

The controller under a large error changes rapidly to correct the error, and changes slowly under a small or minimal error.

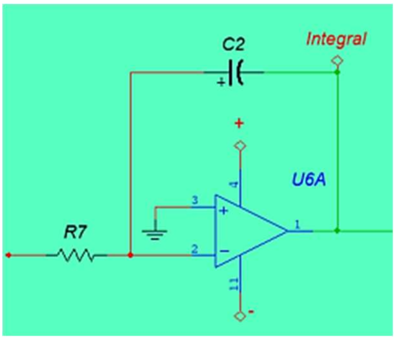

Fig. 9: Integral Controller (Integrator)

In integral form, we have the formula:

$$\int dV_{out} = \int K_i V_{error} \ dt$$

$$V_{out} = K_i \int V_{error} \, dt + V_x$$ ---*

V_x is the initial controller offset.

Applying Laplace transform, and taking the transfer function, we have:

$$\frac{V_{out}}{V_{error}} = \frac{K_i}{S}$$

The capacitor voltage, which is the output voltage is given as:

$$V_c = \frac{1}{R_7 C_2} \int V_{error} \, dt + V_x$$ ---**

Equating (*) and (**), we have:

$$\boldsymbol{K_i = -\frac{1}{R_7 C_2}}$$

$\boldsymbol{R_7 C_2}$ is the integration time.

Summer: This is an op-amp configuration that performs the mathematical operation of adding two values and producing a single output.

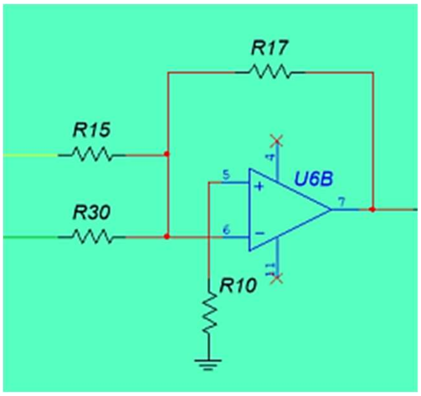

Fig. 10: Summer

For proper operation of the summer, the negative of the actual set point desired is used, which is then inverted (since the summer is configured in an inverting mode, i.e., inputs going in through the inverting input). **The resistor $R10$ compensates for the offset of bias current and should equal the parallel combination of resistances to the inverting terminal.**

Setting $R_{15} = R_{30} = R_{17} = R$, we have that:

$$V_o = V_{error} = V_{sp} - V_{pv}$$

The output of the summer now becomes:

$$V_{out} = -\left(-\frac{R_{17}}{R_{30}} V_{sp} + \frac{R_{17}}{R_{15}} V_{pv}\right)$$

$$Gain, K_s = \frac{-R_{17}}{R}$$

Inverter (Inverting Amplifier): The inverter (also known as inverting amplifier) is a unity gain amplifier that changes the phase of signals passed through it by 360°. Since the output of the summer is inverted as observed from the description above, the inverter is used to return the output to its normal phase.

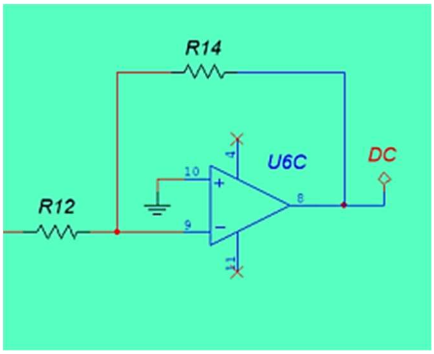

Fig. 11: Inverter

Below is the PI controller circuit parts as one with unknown parameters, to be calculated.

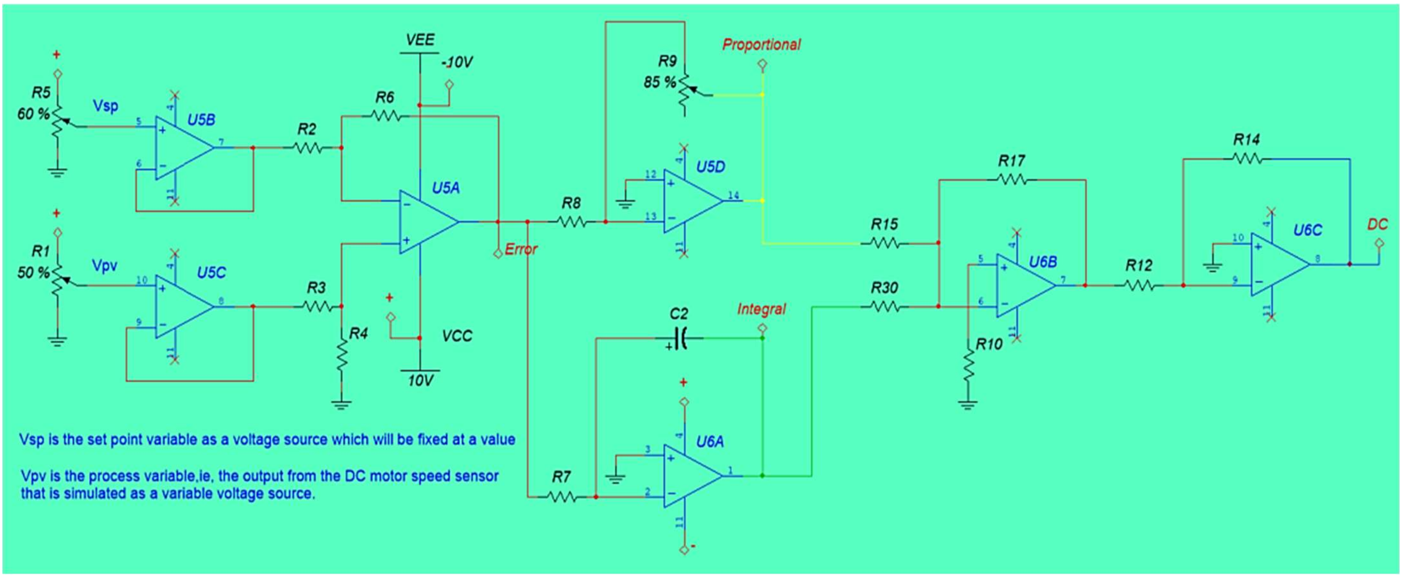

Fig. 12: PI Controller Circuit (With parameters unknown)

DESIGN CALCULATIONS

Buffer

The buffer added at the input of the difference amplifier inputs is to prevent overloading of the feedback source and the set point voltage source.

Difference Amplifier Parameters

$$V_o = V_{error} = \frac{R_6}{R_2}\left(V_{sp} - V_{pv}\right)$$

The amplifier gain is given as:

$$gain = \frac{R_6}{R_2}$$

Since we require a unity gain, the values of R_6 and R_2 must be equal and precisely matched. The value randomly chosen is $\mathbf{22K\Omega}$.

Proportional Controller

The proportional gain is given by the formula:

$$\boldsymbol{K_p = \frac{V_o}{V_{error}} = \frac{R_9}{R_8}}$$

For a variable gain of $0\ to\ 5$, (this is to avoid op-amp saturation), lets set $\boldsymbol{R_8 = 33K\Omega.}$ The value of $\boldsymbol{R_9}$ becomes:

$$\boldsymbol{R_9 = 5 * 33K\Omega = 165K\Omega}$$

The gain of **5** was used in order to get the maximum value of resistor that will handle the maximum gain of the amplifier.

Integral Controller

$$\boldsymbol{K_i = -\frac{1}{R_7 C_2} = \frac{1}{T_i}}$$

Setting $\boldsymbol{K_i = 3V/sec}$ and $\boldsymbol{R_7 = 33K\Omega,}$ we have:

$$\boldsymbol{C_2 = \frac{1}{3 * 33K\Omega} = 10.10\mu F}$$

A fixed value of $\mathbf{10\mu F}$ will be used.

Inverting Summer

$$Gain, K_s = \frac{-R_{17}}{R}$$

For unity gain: $R_{15} = R_{30} = R_{17} = R = 1K\Omega$

Low value resistance was chosen to avoid over-loading the integral and proportional controllers.

The output of the summer is inverted, to bring it back in phase, an inverting amplifier with unity gain is added to the output of the summer.

The completed circuit of the PI controller with calculated values is as shown below:

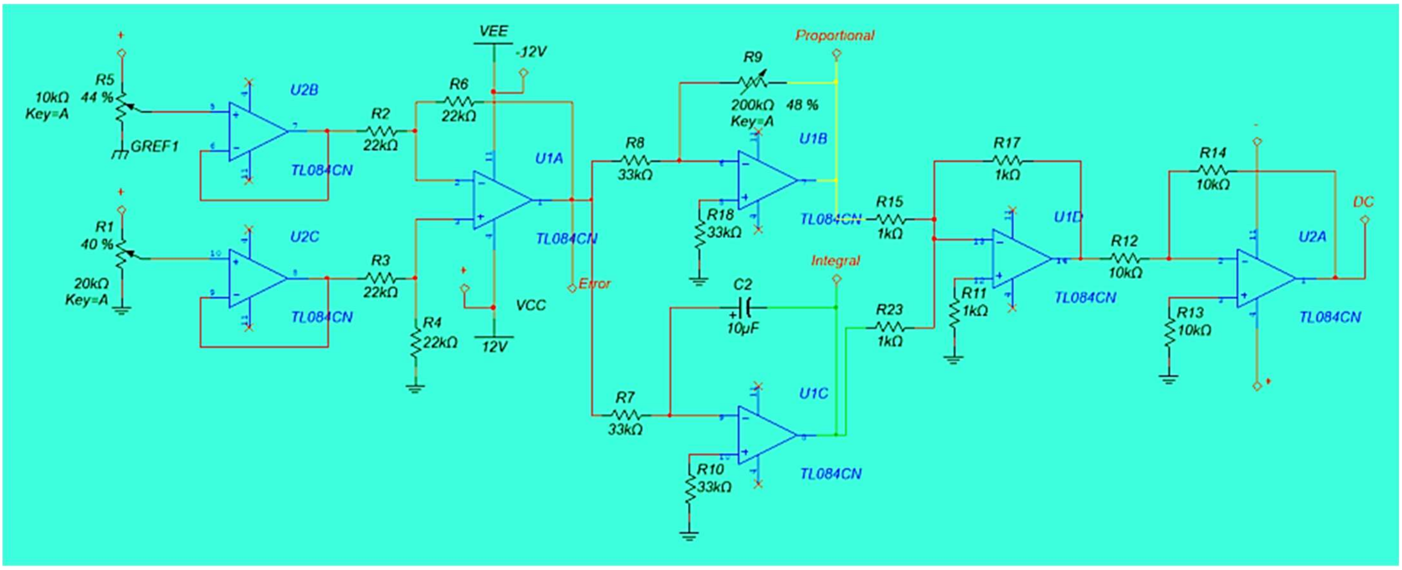

Fig. 13: PI Controller Circuit with calculated parameters

SIMULATION RESULTS

The simulation was carried out using the calculated parameters and the following results were obtained:

Fig. 14: DC Output of the PI Controller

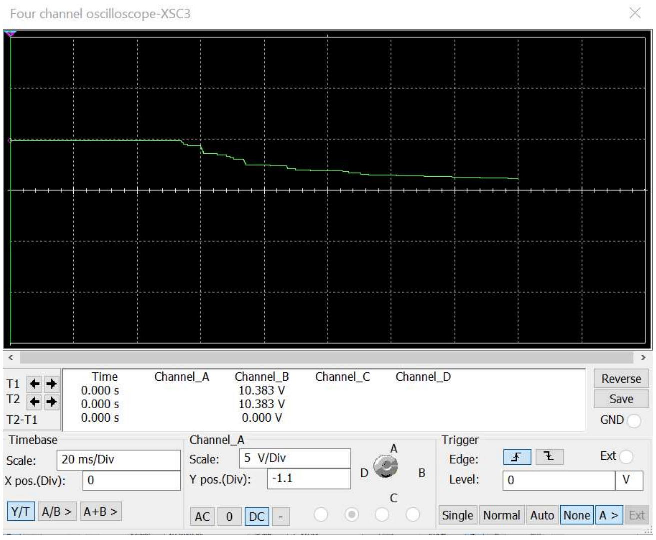

Fig. 15a: Controller Output when PV is increasing

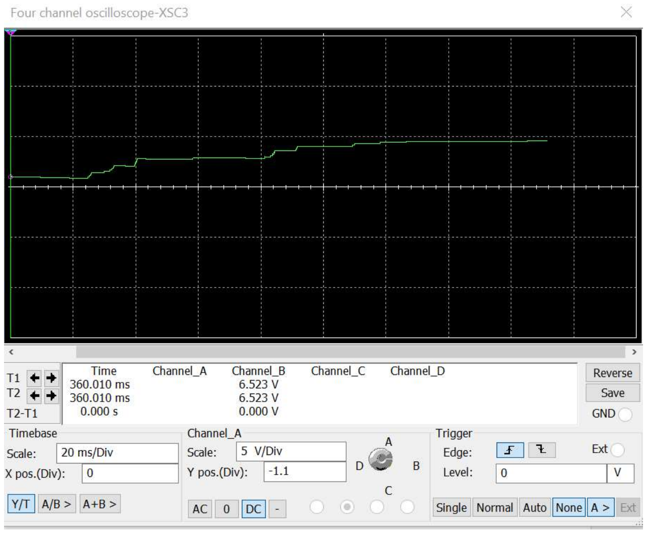

Fig. 15b: Controller Output when PV is decreasing

Explanation: In **Fig. 14**, the DC output is constant at 10.399V DC (cannot get to the op-amp supply voltage to avoid saturation) since there is no change in the process variable (PV) which is being monitored, in this case the DC motor speed. It remains at this point as there is no change in the PV (motor speed).

Fig. 15a shows the output of the controller when the process variable is increasing. As the PV increases, in this case the DC motor speed (this situation occurs when a load is removed from the motor's rotor), the PI controller output decreases in order to maintain the speed of the motor. The PI output decreased to a value of 6.523V DC in order to keep the speed constant.

Fig. 15b displays the controller output as the process variable decreases. The decrease in the process variable indicates a decrease in the DC motor speed, which occurs when more load is being added to the rotor of the motor. At this point, the PI controller output increases in order to maintain the motor speed. The controller output had different incremental levels, which indicates the points the loads were added and speed reduced but the controller maintained the speed at the required rpm.

SIMULATION TEST

A simulation test was carried out to verify the working of the PI controller using a triangular wave generator (which serves as DC supply to the DC motor) and a comparator to compare the triangular wave and the DC output of the PI controller.

In supplying a separately excited DC motor, a direct DC voltage source is not used, rather a Pulse-Width Modulated (PWM) voltage. This is to enable easy control of the motor parameters including speed and torque. In order to generate this Pulse Width Modulated voltage, the Pulse-Width Modulation technique (Triangular Pulse-Width Modulation) is used, which is the use of a comparator to compare a triangular wave and a DC voltage, in this case, the triangular wave generated with the circuit below and the DC output of the PI controller. For more information on Triangular PWM, visit https://microcontrollerslab.com/pwm-introduction-types/.

Three cases will be considered:

1. When the motor is operating at constant speed under load
2. When load is removed from the motor and the speed increases, that is, the process variable increases
3. When load is added to motor and the speed decreases, that is, the process variable decreases

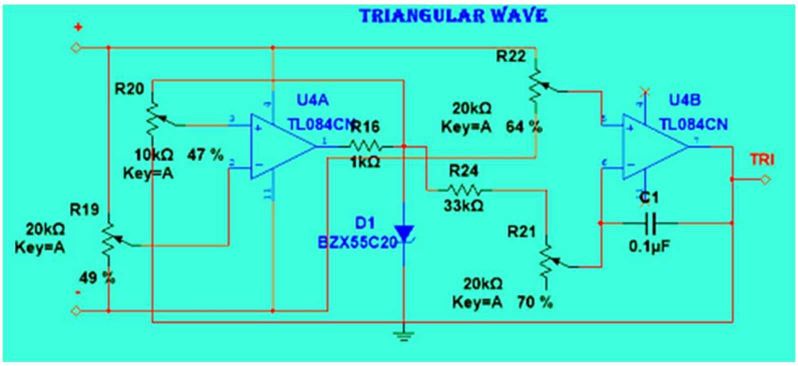

Fig. 16a: Triangular Wave Generator

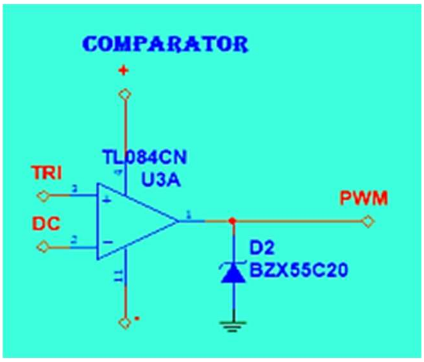

Fig. 16b: Comparator Circuit

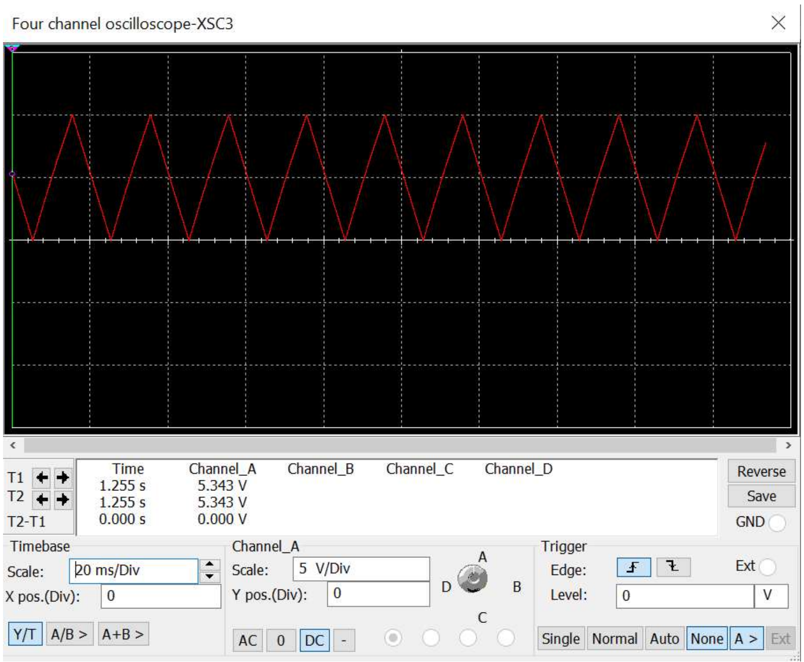

Fig. 17: Generated Triangular Wave

CASE 1: Motor running at constant speed under load

In this situation, the motor is assumed to have been running with a constant load at a constant speed. Since there are no load changes, there is no speed change and thus there is no change in the process variable (PV). The scope is as shown below.

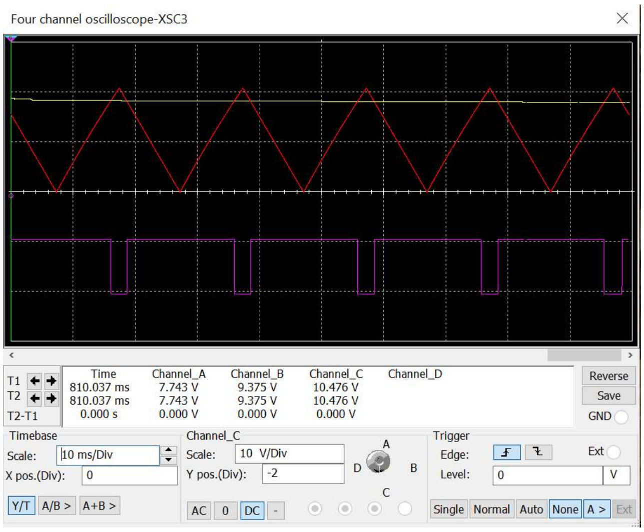

Fig. 18: Scope of Motor running under constant load

From the scope above, the PWM have a higher percentage ON, about 95% ON and 5% OFF, indicating that the motor is running under a high load, thus requires a high voltage for proper running.

CASE 2: When load is removed from the motor and the speed increases

When load is removed from the motor, the speed increases abruptly and may lead to break down. The process variable increases and the PI output decreases, thus reducing the DC supply to the motor and hence bringing the motor speed back to normal operation. The scope is as shown below.

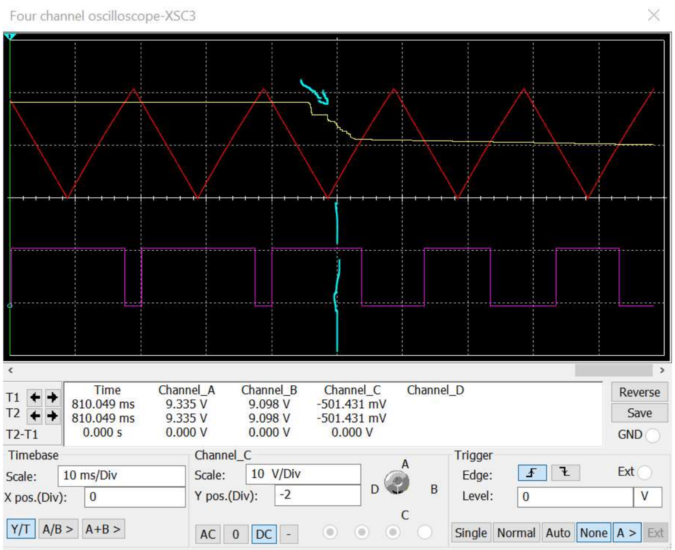

Fig. 19: Scope of Motor running under reduced load

The scope above shows the point of load reduction in the motor (marked with an arrow) and from the PWM output, it can be observed that the voltage input into the motor reduced to a high proportion. The process variable increased and the controller output reduced to stabilize the motor. This is the action of the PI controller to reduce the motor speed and prevent it from over running.

CASE 3: When load is added to the motor and the speed reduces

As load is added back to the motor, the speed of the motor will reduce, thus also reducing the value of the process variable (PV). At this point, the PI controller output increases and supply's more voltage to the motor to keep the speed constant.

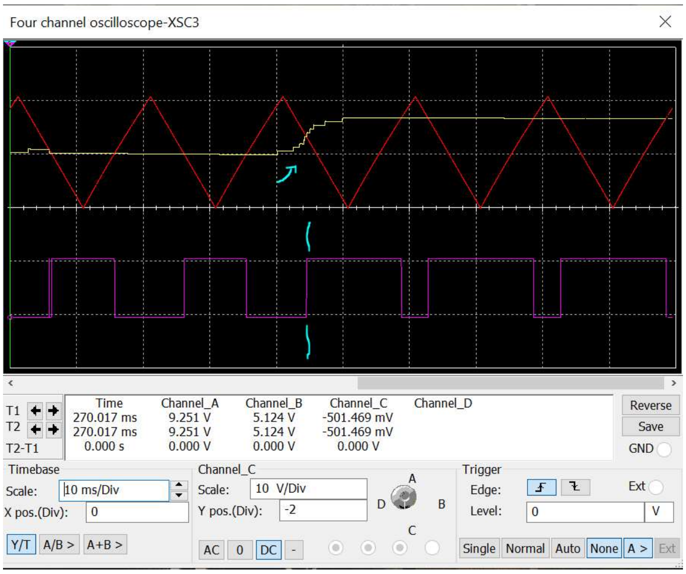

Fig. 20: Scope of Motor running under increased load

The scope shows the point of load addition to the motor, which resulted to an increased voltage supply to maintain the motor speed. This action resulted from a decrease in the process variable (motor speed) as the speed is added, hence the Pi controller output increased in order to supply more voltage to the motor to keep the speed constant.

CONCLUSION

The study, design and simulation of a Proportional Integral (PI) Controller for the speed control of a Separately Excited DC Motor (SEDCM) have been successfully carried out, with the calculated design parameters. The results obtained using National Instruments Multisim simulation software shows the performance of the PI controller. Three case tests were carried out using a triangular wave to generate a PWM signal, which is the supply voltage to the SEDCM. These tests were carried out to observe the controllers behaviour under constant load, reduced load and increased load and the response was as expected.

References

i. EEVblog #600 – Op-Amps Tutorial – What is an Operational Amplifier. Available at: https://www.youtube.com/watch?v=7FYHt5XviKc

ii. Analog PID Control using Op-Amps – Neel Mehta: available at: https://neelpmehta.wordpress.com/analog-pid-control-using-op-amps/

iii. EE462L Power Electronics PI Controller for DC-DC Boost Converter. Available at: https://fdocuments.in/reader/full/ee462l-power-electronics-pi-controller-for-dc-dc-boost-webecs-power-electronics

iv. PID: Using Op-Amp to implement a PI Controller. Available at: https://www.youtube.com/watch?v=OA2TWf8e_0Y

v. OPA01: Op-Amp Based Control Systems – Interdisciplinary Automatic Controls Laboratory. Available at: https://www.lehigh.edu/~inconsy/lab/css/ME389/guidelines/ME389_OPA01_TFSynthesis_Guideline.pdf

vi. Electronic Basics #21: Op-Amp (Operational Amplifiers). Available at: https://www.youtube.com/watch?v=kqCV-HGJc6A

vii. Electronic PI Controller. Available at: https://www.youtube.com/watch?v=4YAeU-XyYes