AC Motor Drive With Power Factor Correction Using Arduino

C. Chakiris, R. Brudnicki, R. Troy, J. Nelson, M. Dittmar, A. Brapoh Jr., M. Andrade, S. Lugo, A. Szabo, K. Dudek

The University of Scranton

Abstract

By using various electrical and computer engineering concepts, this project incorporates different sectors explored through current curriculum. By implementing these concepts, a fully functioning AC motor controller will be designed. The project is split into 5 groups: AC to DC power conversion, DC to AC power control, power factor correction, capacitor bank control, and Arduino interfacing, all working on separate critical components for the motor controller. As this is currently a work in progress, actual conclusions cannot be made, but speculation based on calculations is available.

Introduction

This project, designed to span the duration of the Spring 2023 semester, demonstrates many different principles of electronic circuit design. The desired outcome of a controllable AC motor relies on the integration of many different subsystems assigned to different team members. Each subsystem calls for the application of concepts relevant to electrical engineering, all with a seamless final product in mind.

Figure 1. AC Motor Drive with Power Factor Correction Project

Figure 1, shown above, outlines the breakdown of the components and how they relate to the final product. Each component plays a vital role in allowing the AC motor to be controlled for the desired outcome. When the entire system is assembled, the motor will be controlled to properly blend ingredients to produce dough. The apparatus created to link this project to the dough mixing stage is pictured below in figure 2.

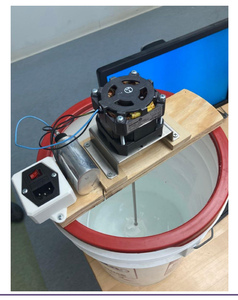

Figure 2. AC motor mounted on bucket with stirring element attached to motor shaft

Method/Experiment

- This project is broken up into 5 distinct groups, each with individual data to explain.
- AC to DC power conversion is required to provide the entire system with power, as the benchtop power supplies are not adequate for this application.
- Using a step-down transformer from the wall outlet and a diode rectifier, the incoming voltage will be fixed at 24v and 3A for this specific motor.
- DC to AC power control will take the incoming DC power and convert it back to AC, however it will be controlled by the pulse-width-modulation from an Arduino to allow user input.
- Power Factor Correction is mandatory as the motor will experience differing levels of mechanical resistance, thus requiring constant waveform correction to maintain efficient power factor.
- Using a Hall Effect sensor along with an AC voltmeter, a script can be used to compare the waveforms, and make corrections to the power factor in real time to be sent to the AC power supply.
- The Adjustable Capacitor Bank is critical in the operation of the AC motor to provide reactive power to compensate for the changing inductive load generated by the motor.
- Multiple tests were run on the motor, including artificial load, to generate data on the size of capacitors needed for a variable array capable of adjusting to 16 different combinations for a variety of inductance changes.
- Arduino is the brain of this entire project and will be used extensively to control the calculations and allow for user operation of the entire system.
- By taking advantage of the extensive shared libraries, the Arduino is controlled by a Python program through Pyfirmata, and will display values of interest, along with allow for user control from an integrated GUI developed using PyQt5.

Data and Analysis

- The DC to AC component of is integral to the control of the motor. By simulating a variable resistor through Python code, the width of DC square wave pulses could be controlled, and their direction inverted. Figure 3 shows how square waves relate to sinusoidal waves. By manipulating the period of these sin waves, control of the motor's speed can be achieved.

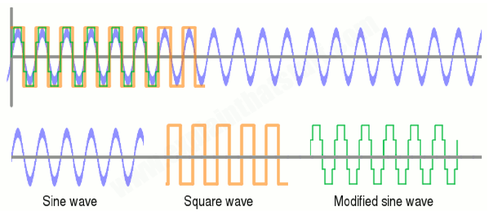

Figure 3. Conversion of square DC waves to sinusoidal AC waves

- The power factor correction circuit can be viewed on Figure 4, where certain elements such as the LCD screen and the lightbulb will be replaced with integrated GUI elements and the motor, respectively. Testing for the power factor can begin pending the arrival of both the Hall effect Sensor and the AC voltmeter. The equation used by the code is simple and will compare 2 instances of the waveform to check if corrections are needed.

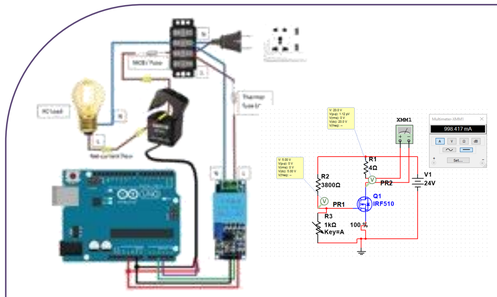

Figure 4, Power factor controller [1]

Figure 5, PyQt5 GUI

$$C = \frac{Q}{2\pi f V^2}$$

$$C = \frac{31.5}{2\pi(6.66)(24)^2} = 65.34\mu f$$

$$C = \frac{31.5}{2\pi(133.33)(24)^2} = 1.31\text{mf}$$

$I_C = \omega c V \quad \omega = 2\pi f$

I_C of $10\mu f = 2\pi(133.33)(10 \times 10 - 6)(24) = 0.2$ A rate for the triac

Power: 17.7W
Voltage: 121.8 V
current: 0.294 A
Frequency: 60 Hz

w/ Pen Pressure
18.4 W
121.8 V
0.299 A
60 Hz

Without Load

$$P_f = \frac{P}{VI} = \frac{17.7}{(121.8)(0.294)} = 0.494$$

$$Q = vI\sqrt{1 - Pf^2} = (121.8)(0.294)\sqrt{1 - (0.494)^2} = 31.135 \text{ V/A}$$

$$C = \frac{Q}{2\pi f V^2} = \frac{31.135}{2\pi(60)(121.3)^2} = 5.57HF$$

With Load

$$P_f = \frac{P}{VI} = \frac{18.4}{(121.8)(0.299)} = 0.505$$

$$Q = VI\sqrt{1 - Pf^2} = (121.8)(0.299)\sqrt{1 - (0.505)^2} = 31.433 \text{ V/A}$$

$$C = \frac{Q}{2\pi f V^2} = \frac{31.433}{2\pi(60)(121.8)^2} = 5.62\text{HF}$$

Values of Capacitors:
Required. $200 - 4000$ (RPM) RPM $\rightarrow$ Hz

$$Hz = \frac{RPM \times Poles}{120} \quad SO = \frac{RPM \times 4}{120}$$

$$Hz \min = \frac{200 \times 4}{120} = 6.66 \text{ Hz}$$

$$Hz \max = \frac{4,000 \times 4}{120} = 133.33 \text{ Hz}$$

$$C = \frac{Q}{2\pi f V^2}$$

$$C = \frac{31.5}{2\pi(6.66)(120)^2} = 52.34f$$

$$C = \frac{31.5}{2\pi(133.33)(120)^2} = 26.14f$$

Figure 4, Calculations

$$Power\ Factor\ (pf) = \cos\theta_z$$

- The capacitor bank will utilize Triac semiconductors as switches to control the 16 possible combinations of needed capacitors. The values selected for the 4 capacitors are 100 µF, 50 µF, 25 µF, and 12 µF for the 120V. The Figure 6 calculations are used for the 24V side to provide upwards of a 30% improvement of power factor.
- The Python GUI using PyQt5 and Pyfirmata (Figure 5), works very well with controlling a variety of Arduino functions including reading/writing to digital pins, analyzing analog inputs, and controlling the PWM function all from the Figure 4 test GUI.

Conclusion

- As this project is currently still in progress, there are many unknowns that need to be addressed in the coming weeks.
- Currently the major hangups are awaiting parts to test the current theoretical implementations.
- Using Python allows for much more freedom without sacrificing the ability to accurately control the Arduino

References

[1] Solarduino. "How to measure power factor and phase angle with Arduino?" Solarduino.com https://solarduino.com/how-to-measure-power-factor-and-phase-angle-with-arduino/ (accessed Feb. 15, 2023)

[2] C. Woodford. "Inverters." Explainthatstuff.com. https://www.explainthatstuff.com/how-inverters-work.html (accessed Mar. 8, 2023)

Acknowledgements

Professor Kenneth Dudeck outlined the basis for the entire project and assists students when necessary. Laboratory equipment manager Mr. Majid Mokhtari also provides continued support through the process. Mr. Mokhtari is responsible for acquiring components and constructing any necessary physical components.