

## INTRODUCTION

In a circuit with reactive (inductive or capacitive) loads, the voltage and current are about 90 degrees out of phase. Inductive loads are mainly found in industries that use heavy equipment driven by motors. These inductive loads cause the current to lag the voltage which results in the consumption of reactive power that does no useful work and a low power factor. The traditional way of power factor correction has been by connecting capacitors in parallel to the inductive load. These capacitors generate reactive power for the inductive load to consume thereby improving the power factor as reactive power would not be drawn from the supply. The major issue with this method is that the exact value of capacitance to be added is unknown because the inductive load is unknown hence the capacitors are switched into the circuit until the power factor is almost unity.

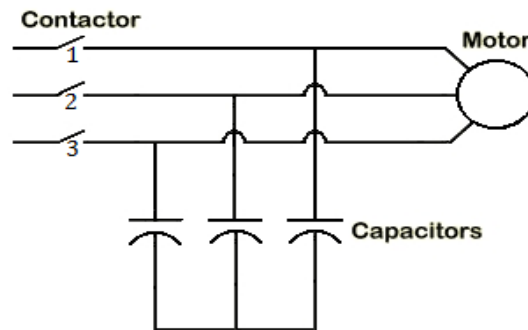


Figure 1: Capacitor bank connected in parallel to inductive load.

## PROPOSED SOLUTION

The aim of this project is to design and implement a circuit with a controller that supplies the necessary reactive power to an unknown inductive circuit. This circuit consists of a buck converter connected to a capacitor and connected in parallel to an inductive load. By controlling the duty factor of the buck converter, the apparent capacitance seen by the main load can be adjusted thus improve the power factor.

## STEPS TAKEN

1. MATLAB Simulation
2. Spice Simulations
3. Hardware Implementation

## MATLAB Simulation

Given any sinusoidal voltage and current, a MATLAB script was created in order to determine the phase difference between voltage and current and the peak magnitudes of the voltage and current. From the results obtained, the value of capacitance required can be added.

If  $V = A \sin \omega t$  and  $I = B \sin(\omega t - \theta)$  then  $C = \frac{1}{\omega} \sin(\theta) * \frac{B}{A}$  where  $\omega = 2\pi f$

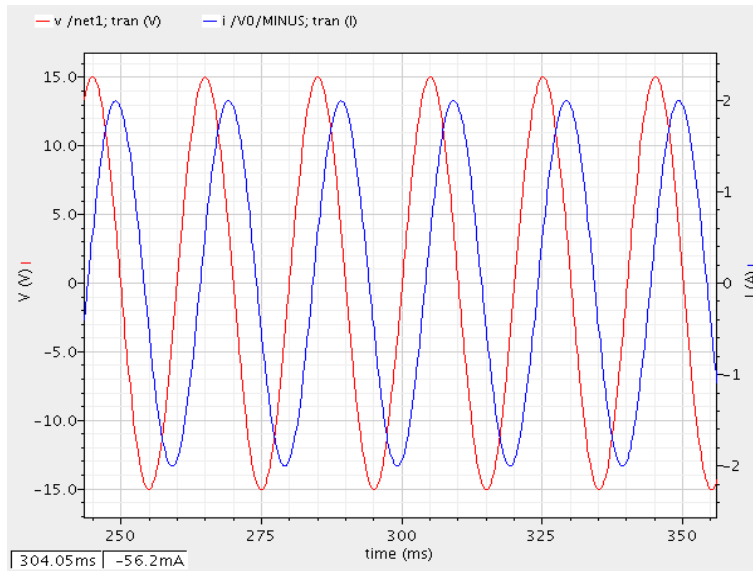


Figure 2: Waveform before adding a Capacitor

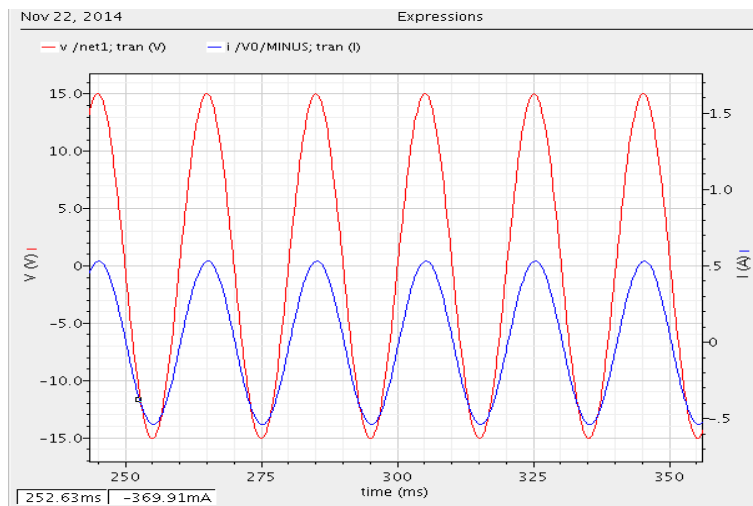


Figure 3: Waveform after adding a capacitor

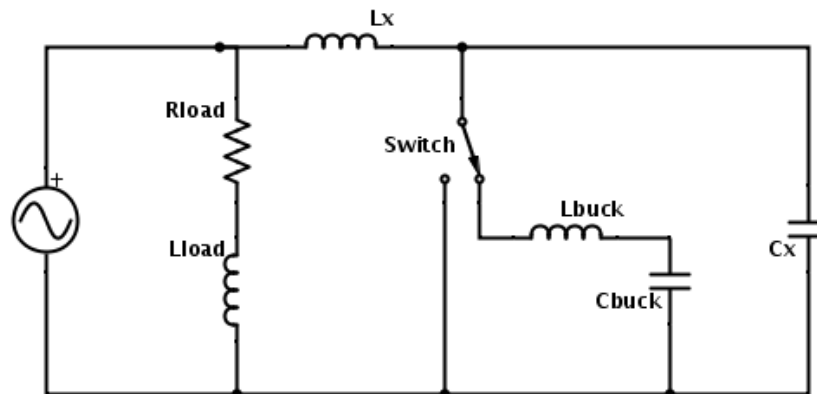


Figure 4: Schematic of circuit used

With this achieved, a controller implemented in open loop was designed using MATLAB as follows:

1. Determine whether the current is leading or lagging the voltage
2. Increase or decrease the duty factor accordingly with a given duty cycle step.
3. If the duty factor changes direction, cut the duty cycle step in half.

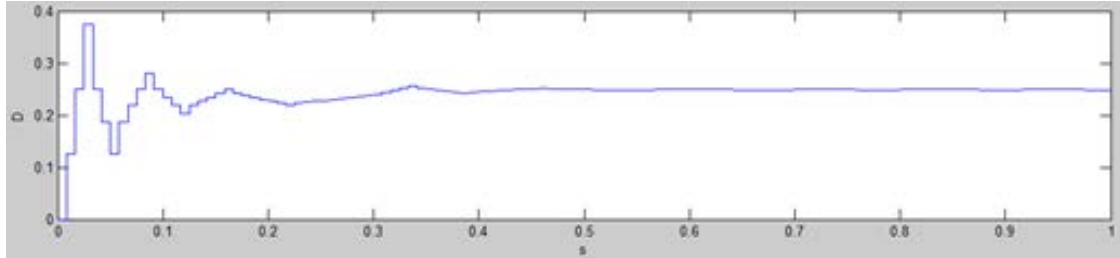


Figure 5: Duty factor converging to a value necessary to attain a unity power factor

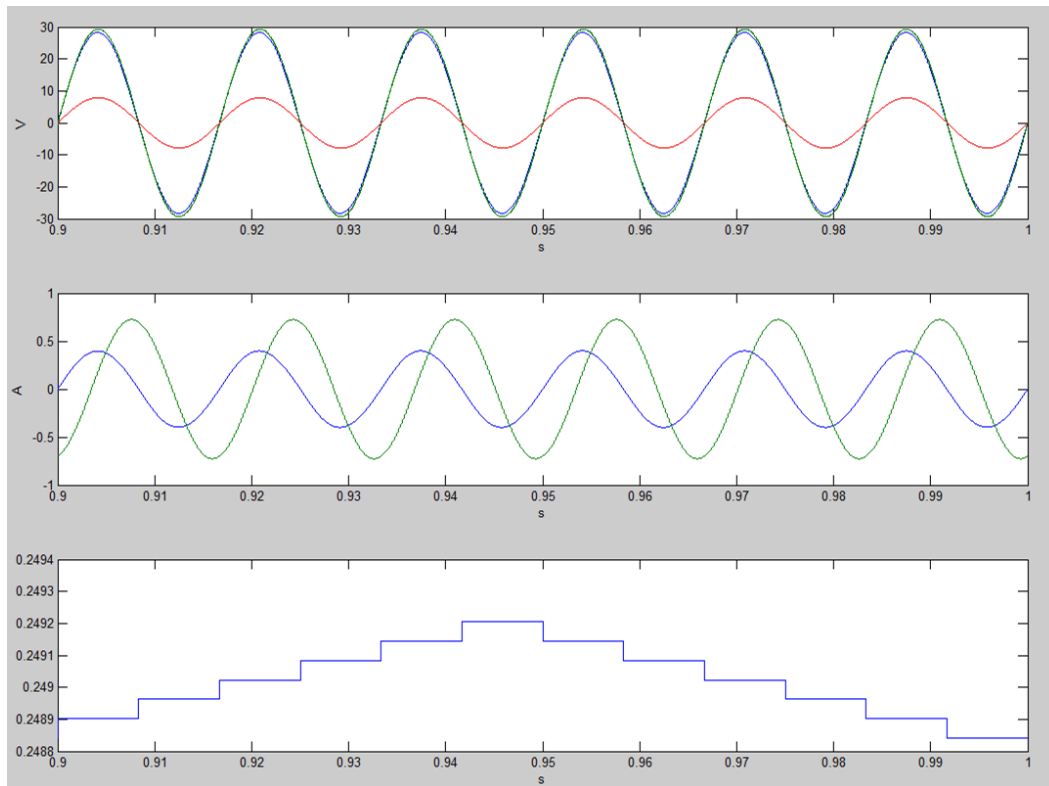


Figure 6: Expanded view showing voltage and current waveform in phase (Red plot in first subplot is Voltage while the Blue plot in the second subplot is current)

## Spice Simulation

Simulating the circuit using LTSpice yielded perfect results as expected. Ideal switches with an on-resistance were used initially for the buck circuit.

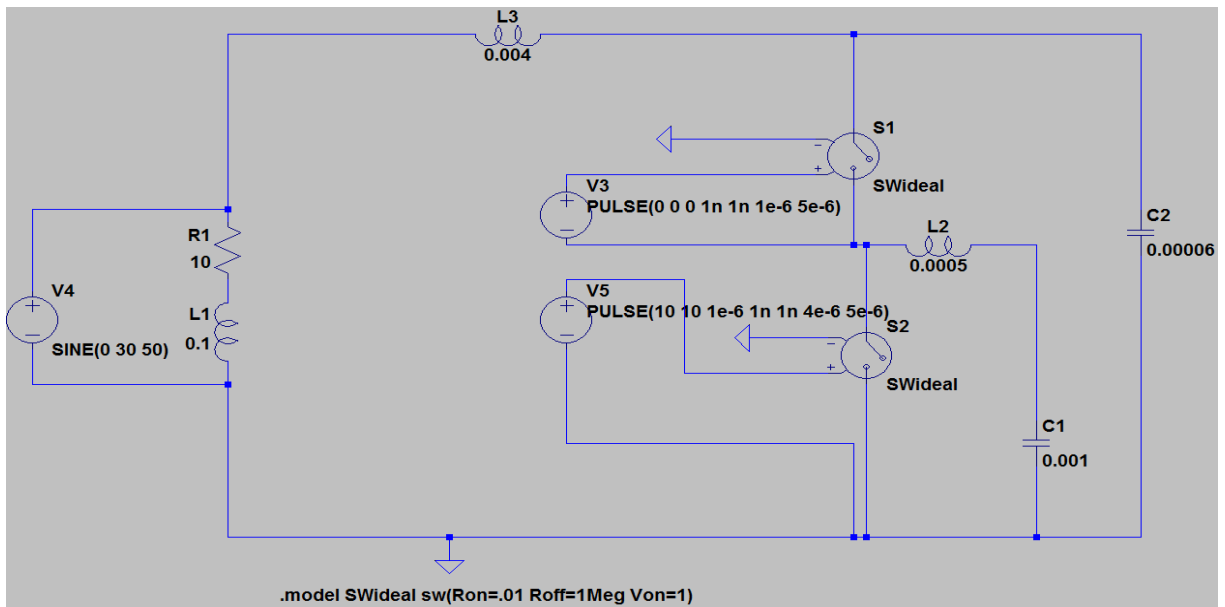


Figure 7: Circuit Simulation in LTSpice using ideal switches

The circuit was simulated for conditions with a duty cycle of 0%, 20% and 40%. From the MATLAB simulation, the duty cycle necessary to attain a unity power factor is about 20%.

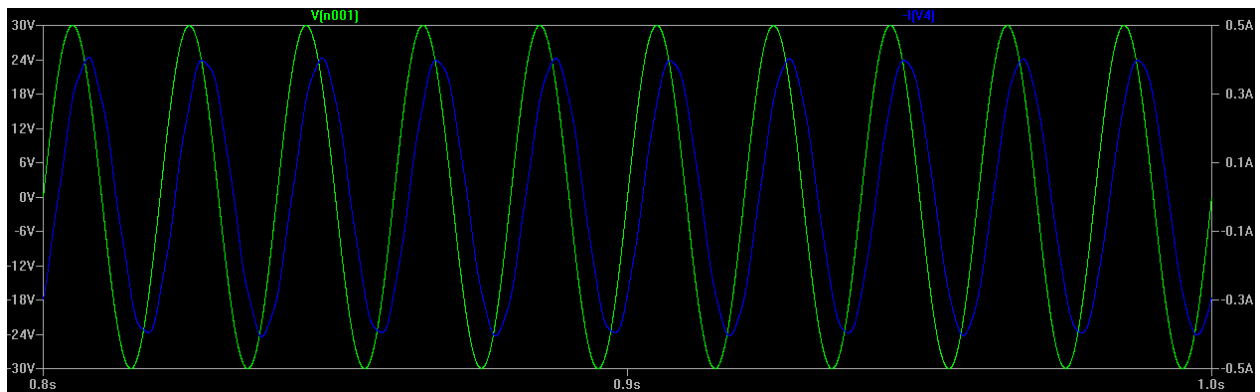


Figure 8: Waveform of voltage leading current at 0% duty factor.

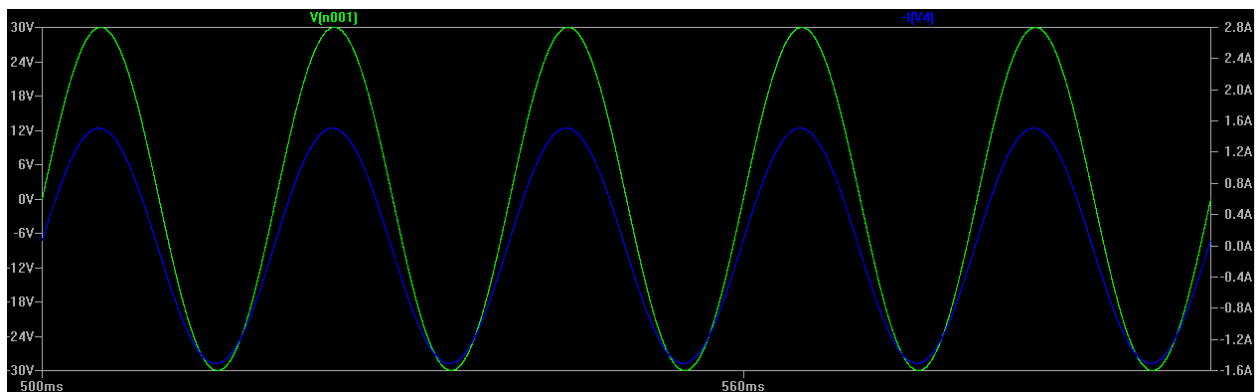


Figure 9: Waveform of voltage in phase with current at 20% duty factor

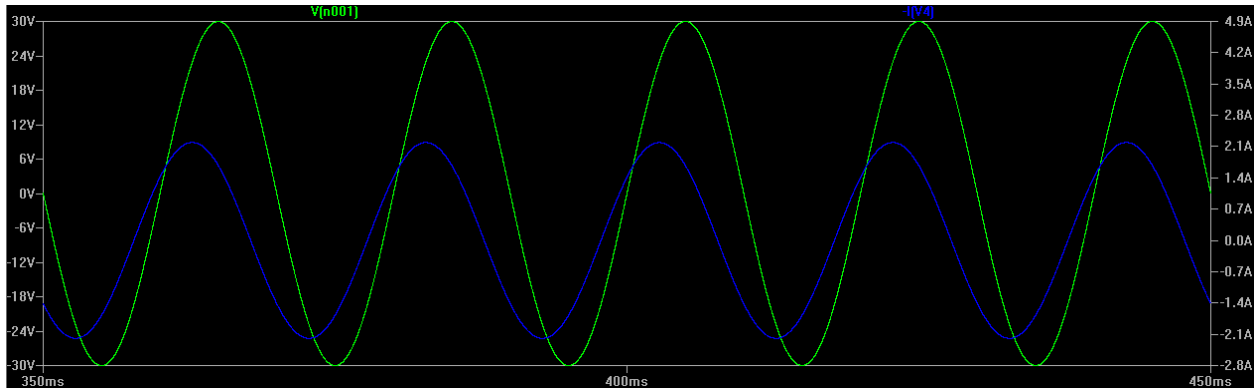


Figure 10: Waveform of voltage lagging current at 40% duty factor

Proceeding further to simulate with FETs, we realized that we had made a fundamental design error by thinking FETs will switch AC signals. To remedy this issue, we tried using an unfolding bridge to rectify the AC signal before passing it through the FET but we still weren't getting the desired result as there were unexplainable current spikes up to kilo amps values despite including dead and rise times. We tried different modifications to no avail.

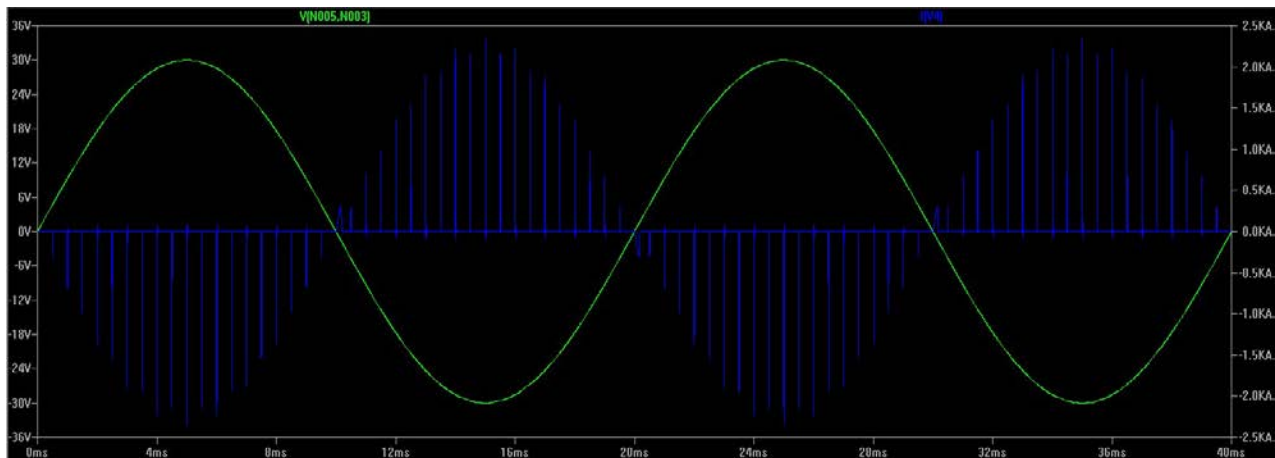


Figure 11: Waveform showing current spikes when implemented with FET and unfolding bridge

## Hardware Implementation

We intended building a low scale circuit using a function generator as an 8V AC source, the half-bridge module used as switches and a 100mH inductor as an inductive load. The first issue faced was programming the FPGA to determine if the current is leading or lagging. The first method used was to determine the time difference between the zero crossing of the voltage and current. The difficulty faced with this method was getting the accurate timing at zero crossing as the voltage and current were quite noisy. We resorted to another simpler method which determines the sign of current at the zero crossing of the voltage. If the sign current is negative, then the voltage is leading and vice versa. This method implements an open loop controller as the actual time difference cannot be obtained. The next issue faced with hardware implementation was getting the buck circuit to switch an AC signal. We haven't been able to solve this issue and are currently trying to figure this out using SPICE.

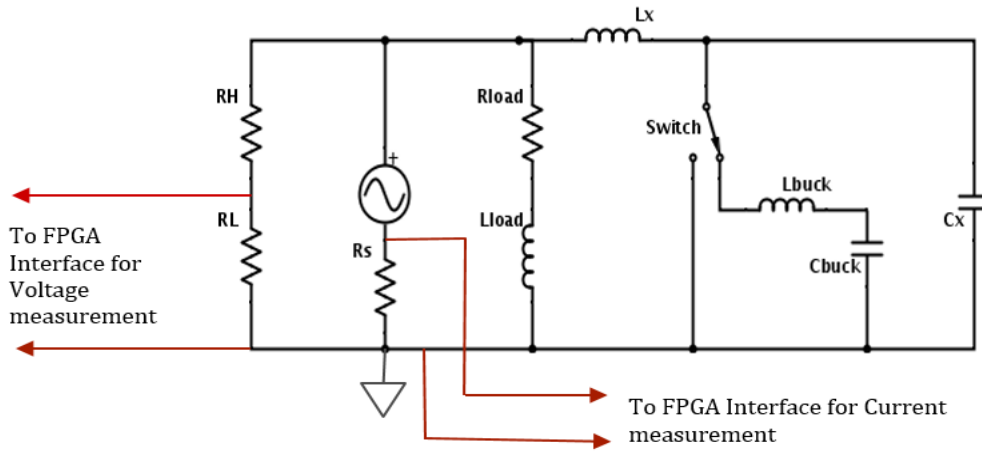


Figure 12: Circuit schematics used in hardware implementation

### Knowledge Gained

1. Proper understanding of design before delving into implementation.
2. Getting a controller to work on the FPGA.
3. In-depth understanding and practical applications of FETs.
4. Circuit debugging in a systematic manner.

### Future Implementations

Going forward, we will like to get our spice simulation functioning using FET. We have already gone through different iterations and we hope to do some more. Below is a circuit we would like to try out

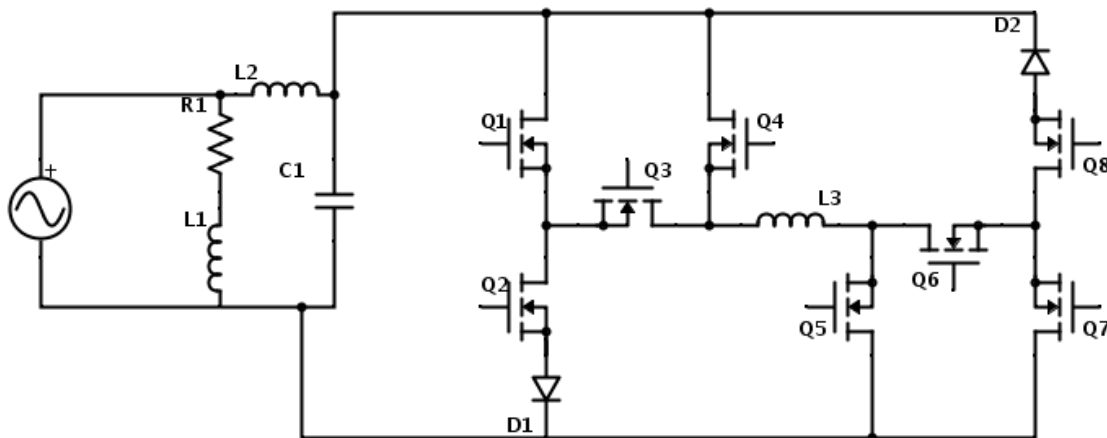


Figure 13: Schematic of circuit to try out next

Furthermore, we also want to have a working circuit by finishing the hardware implementation and scaling it up to handle 120V AC.