

Santa Clara University Scholar Commons

Electrical Engineering Senior Theses

Student Scholarship

6-8-2016

Reactive Power Compensation

Kimberly Meyers
Santa Clara University

Martin Prado
Santa Clara University

Follow this and additional works at: http://scholarcommons.scu.edu/elec_senior

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Meyers, Kimberly and Prado, Martin, "Reactive Power Compensation" (2016). *Electrical Engineering Senior Theses*. Paper 28.

This Thesis is brought to you for free and open access by the Student Scholarship at Scholar Commons. It has been accepted for inclusion in Electrical Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact rscroggin@scu.edu.

SANTA CLARA UNIVERSITY

Department of Electrical Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Kimberly Meyers, Martin Prado

ENTITLED

Reactive Power Compensation

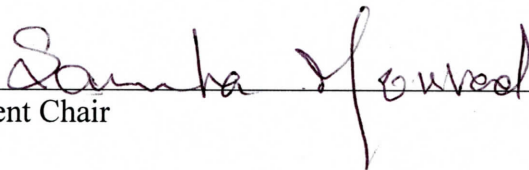
BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE
IN
ELECTRICAL ENGINEERING



Thesis Advisor

6/8/2016
date



Department Chair

6/8/2016
date

REACTIVE POWER COMPENSATION

By

Kimberly Meyers, Martin Prado

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical Engineering

of

SANTA CLARA UNIVERSITY

In Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical Engineering

Santa Clara, California

June 8th 2016

ABSTRACT

Reactive power compensation serves to increase the power factor of large industrial loads and increase the efficiency of power transmission. In doing so, reactive power compensation decreases overall energy consumption and has lasting economic and environmental impacts. This paper discusses a simulation to model power consumption in a multitude of different loads, which gives industrial consumers a solution to increase the power factor of their systems to a more desirable level. The MATLAB simulation that we built successfully calculates and displays how much energy can be saved and it shows how that energy can be saved. If physical implementations of our simulation are adopted on a large scale, the economic and environmental impact will be substantial.

TABLE OF CONTENTS

	Page
1. Introduction.....	1
1.1. Objectives.....	1
1.2. Requirements.....	1
1.3. Background.....	2
1.3.1. Need.....	2
1.3.2. Reactive Power Compensation.....	3
1.3.3. Prior Skills.....	4
2. Design.....	5
2.1. Design Alternatives.....	5
2.1.1. Constraints.....	5
2.1.2. Physical Model.....	5
2.1.3. MATLAB Simulink.....	6
2.2. Final Design.....	6
2.2.1. Transmission Line Losses.....	7
2.2.2. Original Power Calculations.....	7
2.2.3. Power Factor Correction.....	8
2.2.4. Building the Load.....	8
2.2.5. Functions.....	9
2.2.6. Testing.....	10
3. Results.....	11
3.1. Transmission Line Losses.....	11
3.2. The Grid Load.....	11
3.3. Reactive Power Compensated.....	12
4. Ethical Analysis.....	16
4.1. Ethical Justification.....	16
4.2. Ethics, Our Project and a Good Engineer.....	17
4.3. Risks.....	18
5. Impacts.....	20
5.1. Economic Impact.....	20
5.2. Environmental Impact.....	21
5.3. Grid Stability.....	22
6. Conclusion.....	24
7. References.....	25
8. Appendices.....	26

LIST OF FIGURES

	Page
1 Power Factor Calculation.....	3
2 24 Hour Test Load.....	10
3 24 Hour Day Load.....	12
4 Change in Reactive Power.....	13
5 Change in Apparent Power.....	13
6 Increase in Power Factor.....	14
7 Phase Angle Correction.....	14
8 The Duck Curve.....	22

LIST OF TABLES

	Page
1 MATLAB Functions.....	9
2 Reactive Power Savings Example.....	20

1 Introduction

This report gives an in depth outline of a MATLAB simulation built to increase the efficiency of power transmission. The introduction covers the purpose of the project and the background information on reactive power compensation. The paper then details the design of the simulation, followed by the results seen upon completion of the project. An ethical analysis and a discussion of the impact of this project completes the paper.

1.1 Objectives

This project provides a tool for use by industrial consumers or power companies to increase the efficiency of power transmission along every step of the transmission process, from generation to consumption, leading to lasting environmental and economic benefits all the while maintaining a stable grid.

1.2 Requirements

The simulation is designed to take in a wide range of different loads seen by short transmission lines. The simulation is built to have inputs of the voltage, load and desired power factor of a system. Using the impedance characteristics of short line transmission lines, realistic transmission losses had to be taken into account to provide a more accurate representation of the actual voltages and currents seen by the load. The simulation is designed to output the improved power factor of the inputted system along with the improvements of reactive and apparent power. To further increase the accuracy of the simulation, the capacitor bank added in parallel to the system is modeled after high voltage rated capacitor values.

1.3 Background

1.3.1 Need

Current high voltage commercial and industrial loads are cause for much wasted power. In commercial and industrial inductive loads, the power factor, a measure of efficiency, remains lower than ideal (typically anywhere from 0.7-0.95), meaning that much of the power transmitted to the load is wasted. Renewable energy production techniques are great solutions to decreasing the harmful effects of coal and natural gas energy production, however, reduction in energy consumption can have just as large of an impact. Reactive power compensation can complement renewable energy by increasing the power factor of a system, thereby decreasing the amount of energy that needs to be produced. Along with an environmental benefits, power companies and consumers will see economic benefits if reactive power compensation is implemented.

Another problem that reactive power compensation addresses is the instability of the grid that could be seen due to the increase in home solar systems. Due to the lessened power draw throughout the day, followed by a peak draw at night from residential areas, the grid could become less stable, requiring many generators to be turned on in a short amount of time. Reactive power compensation causes the peak to decrease and the apparent power to change only based on how the real power changes, maintaining stability throughout the entire day.

1.3.2 Reactive Power Compensation

The power factor of a system is a measure of the efficiency of the power seen by the load. As shown in Figure 1, the apparent power transmitted is split into two parts: the reactive power and real power. Reactive power is wasted power caused by inductive loads. The real power is the power actually used by the load. The power factor is a ratio of the real power to the apparent power. In order to increase the power factor, the amount of reactive power transmitted must be decreased, causing the apparent power to be mostly comprised of real power.

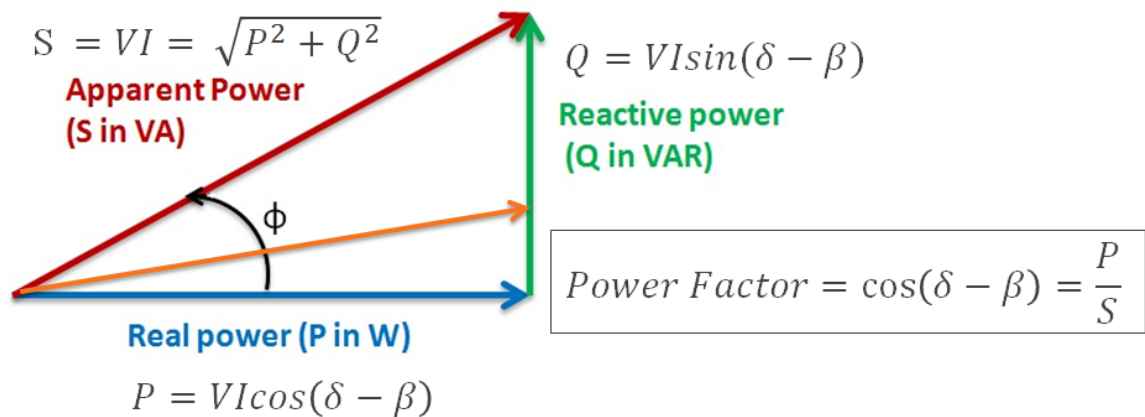


Figure 1: Power Factor Calculation

In order to increase the power factor of an inductive load, capacitors can be added in parallel to the load. The value of the capacitance depends on the desired reduced amount of reactive power. By switching a different number of capacitors on or off, a variable capacitor bank added in parallel to a load allows for the capacitance to vary depending on what value of capacitance is needed.

Many power companies already have variable capacitor banks implemented in the transmission lines or they have the resources and knowledge to do so. On the other hand, many commercial and industrial consumers do not have these setups implemented, explaining why the project focuses on these types of consumers.

1.3.3 Prior Skills

Our understanding of power and power transmission obtained in our Power Systems Analysis class provided us with the theory and equations behind reactive power compensation. Further research had to be done in order to understand what realistic values might be seen for the loads, voltages and power factors. We had enough familiarity with MATLAB to build the entire code, only having to do minor research when we were uncertain of how a MATLAB function might work.

2 Design

2.1 Design Alternatives

2.1.1 Constraints

The largest constraint faced was that physically working with high voltage transmission is not allowed by Santa Clara University's Engineering Department due to safety concerns. The team from the previous year that built a dynamic capacitor bank ran into this same issue, but decided to implement their design on a low voltage scale. While successful, the previous project is not scalable to high power systems, one of our prominent requirements. For this reason, the project focused on simulation instead of working with any physical reactive power systems.

The requirement that the simulation be as accurate to a realistic system was cause for a second constraint. In simulation, any capacitor value could be added in parallel to the load in order to reach a power factor of one. However, that would not be a model of a realistic system. The simulation was constrained by capacitor values that are actually present in the industry, leading to power factors hovering slightly below the desired values with a percent error of around two percent.

2.1.2 Physical Model

Originally accompanying the simulation was a physical model that would be attached to a load in order to determine the actual power factor and show what could be done to the load in order to increase the efficiency to a desired level. As the simulation developed, the need for a physical model greatly decreased as we realized that it would serve little

purpose due to its inaccuracies. Unable to work with high voltage loads, the physical model would not be scalable and would not actually correct the power factor as the simulation does. It was deemed an unnecessary use of time, allowing the focus to remain on improving the simulation.

2.1.3 MATLAB Simulink

A MATLAB Simulink model was also proposed to increase the user friendliness of the MATLAB code. In using Simulink, the code would be easier to follow, and anyone unfamiliar with the code would be able to use the simulation. As the simulation developed, the decision was made to not transfer over to Simulink in order to focus on the functionality of the code instead of the readability. Since the Simulink model would have the same exact function as the original code, it was considered a luxury but not a necessity. Not having to transfer over maintained the focus on improving the function of the already developed code.

2.2 Final Design

Our final simulation was built using MATLAB scripts and functions. First we calculate the transmission line losses according to the inputted length and impedance characteristics of the line. The apparent, real and reactive powers are then calculated at the load side along with the power factor of the power seen by the load. The necessary parallel capacitance is calculated based on the desired decrease in reactive power and the corresponding number of capacitors is added to the load, increasing the power factor.

This process continues for the set amount of time that the load is changing. A more in-depth analysis of every step and function follows.

2.2.1 Transmission Line Losses

The first main step in the code calculates the transmission line losses based on the specifications of the transmission line that the user inputs. For our testing, we used a transmission line length of 70km with a per length (km) line impedance of $0.0679+j0.289$ ohms. These are typical values seen in short transmission lines. The resistive part of the line impedance is given a random variance between 0.001 and 0.01 to simulate the noise that a realistic transmission line would see. The inductive part is given a random variance between 0.01 and 0.1. The transmission line characteristics can be easily changed depending on the load for which the simulation is being run.

2.2.2 Original Power Calculations

After calculating the current through the transmission line and load, the voltage drop through the line is subtracted from the sending voltage. Using the new voltage and the previously calculated current, the real, reactive and apparent powers are calculated along with the power factor. The equations for these calculations are shown below.

$$\text{Real Power: } P = V * I * \cos(\theta_v - \theta_i)$$

$$\text{Reactive Power: } Q = V * I * \sin(\theta_v - \theta_i)$$

$$\text{Apparent Power: } S = V * I$$

$$\text{Power Factor: } PF = \cos(\theta_v - \theta_i)$$

2.2.3 Power Factor Correction

Depending on the desired power factor, the amount of reactive power that must be reduced is calculated. The amount of parallel capacitance needed to reduce this reactive power is calculated and then translated into the correct number of capacitors that need to be turned on in the capacitor bank. The calculations are shown below.

$$\text{New Apparent Power: } S_{desired} = P/PF_{desired}$$

$$\text{New Reactive Power: } Q_{desired} = \sqrt{S_{desired}^2 - P^2}$$

$$\text{Reduced Reactive Power: } Q_C = Q_{initial} - Q_{desired}$$

$$\text{Necessary Impedance: } X_C = V_{Amp}^2/Q_C$$

$$\text{Necessary Capacitance: } C = 1/(2\pi f * X_C)$$

Once these values are calculated, the new capacitance is added in parallel to the load. The power calculations are all redone with the new load and the improved power factor is calculated and displayed. This process continues until the load stops changing.

2.2.4 Building the Load

The main load that we focused on modeling was a 24-hour load representing the grid power draw seen in Silicon Valley. Using average hourly power values from Silicon Valley Power, we separated the day into 12 two-hour increments. Each two-hour increment was given a range of resistance values and a range of inductance values based on typical grid loads running at power factors between 0.85-0.9 seen within each two hour period.

2.2.5 Functions

All of the functions that were created in MATLAB along with the role of the functions are listed in Table 1.

Table 1: MATLAB Functions

Function	Role	Inputs	Outputs
power_factor	Calculates power factor	thetav, thetai	pf
trans_build	Calculates transmission line impedance	length, perlen_imp	line_impedance
load_calc	Calculates initial current phase angle and impedance of load	R, X, thetav	impedance,
load_build	Builds load over a certain time frame	R, L, time, num_load	R_output, L_output
day_load	Builds specific day load	R_bounds, L_bounds, rate, total_time	Rout, Lout
cap_change	Calculates capacitance value needed and number of capacitors to turn on	capacitance_needed	caps_on, cap_value
curr_calc	Calculates current	Vmag, impedance, thetai, f, t	curmag, current
load_calc_parallel	Calculates load after added capacitance	R, L, C_par, f	Res, Xreac

2.2.6 Testing

We tested each function outside of the main code in order to verify that each function worked as expected. Functions were tested using typical input values that might be seen. The outputs were compared to already known or hand calculated results. After successful testing, a function was easily integrated into the main code.

To verify that the main code as a whole functioned properly, we implemented a test load with randomized resistances and inductances within a certain range over a simulated 24-hour period. A generation of the test load is shown in Figure 2.

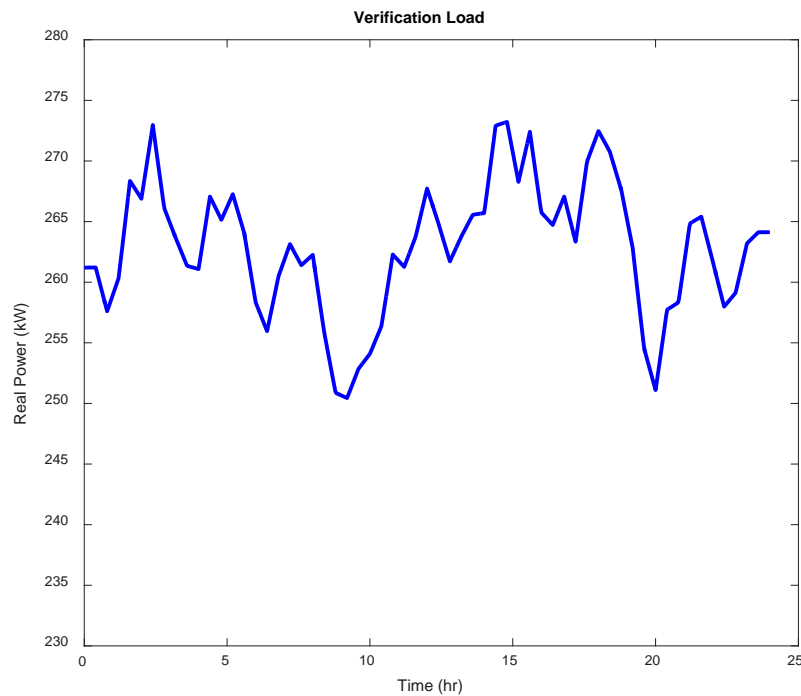


Figure 2: 24 Hour Test Load

3 Results

The simulation successfully outputs the number of capacitors needed to increase the power factor to the desired level. It also shows the decrease in reactive and apparent power supplied to the load, as well as the increase in the power factor throughout a 24-hour period. All of this is accomplished after the transmission line losses are calculated and factored into the voltage that the load sees.

3.1 Transmission Line Losses

Depending on the transmission line specifications input by the user, the simulation subtracts the transmission line losses from the sending end voltage. Transmission line specifications are determined from the standard values of different types of transmission lines and vary depending on the material, length and width of the line. These specifications must be known in order for the simulation to accurately represent the entire power system.

3.2 The Grid Load

The main load built was a model of a 24 hour day based off of Silicon Valley Power's average loads seen in March. While this is a very specific load, it gave us the ability to verify that our simulation was working properly and it gave a wide range of values for our simulation to compute. Not to mention, most loads inputted into our system will be specific and will likely be inputted in 12 to 24 hour increments. The final load built is shown below in Figure 3.

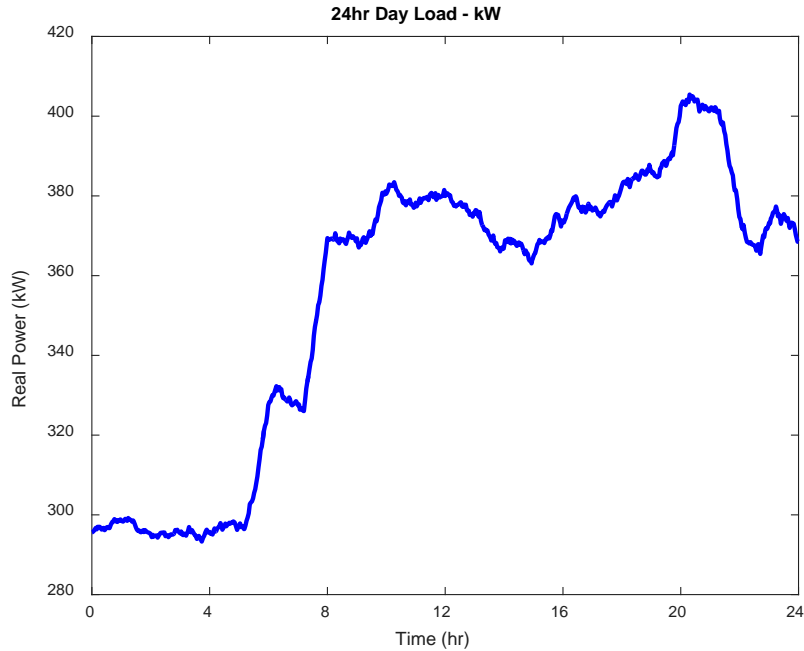


Figure 3: 24 Hour Day Load

3.3 Reactive Power Compensated

The most telling results of our simulation can be seen in Figure 4 and Figure 5. The first plot shows the decrease in reactive power seen when implementing a variable capacitor bank. Without reactive power compensation, the amount of added reactive power hovers around 175 kVAR. The next plot details the decrease in apparent power due to reactive power compensation throughout a day load. As the apparent power decreases and the real power remains the same, we see an increase of power factor to around 0.987 as shown in Figure 6. Our desired power factor of 0.99 cannot be reached due to the rounding error that occurs when switching on or off a specific number of capacitors. While less accurate of a simulation, it portrays a more realistic power factor correction system.

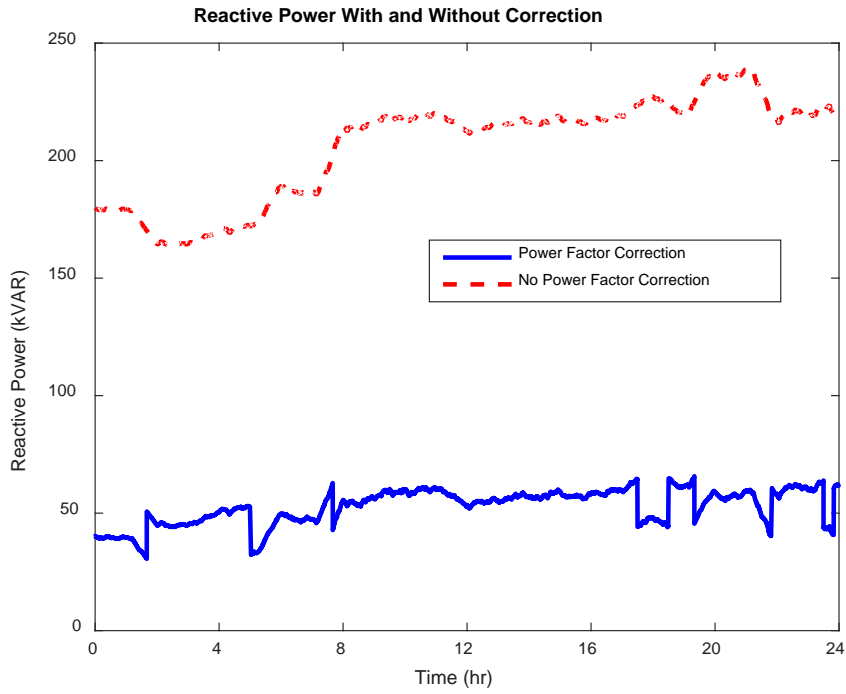


Figure 4: Change in Reactive Power

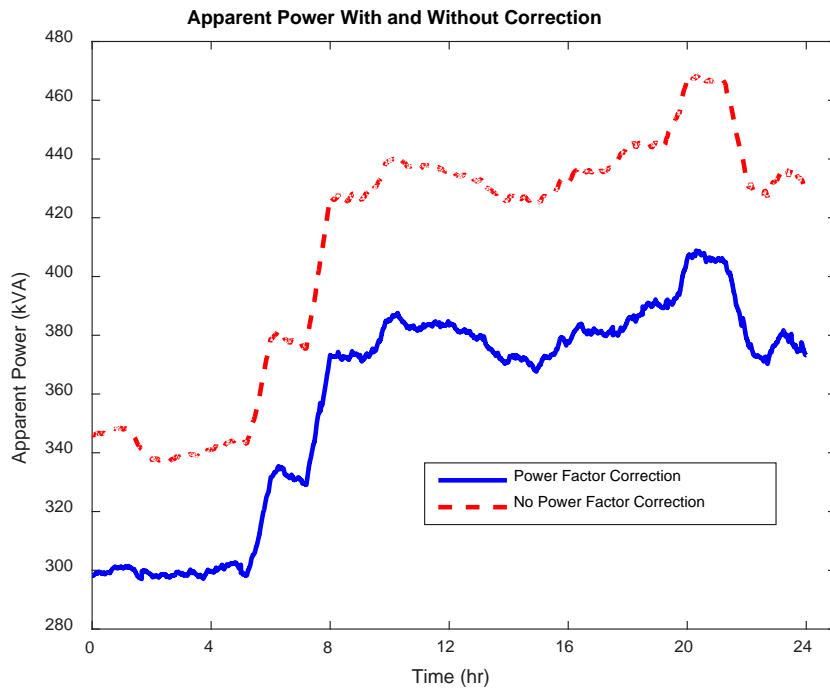


Figure 5: Change in Apparent Power

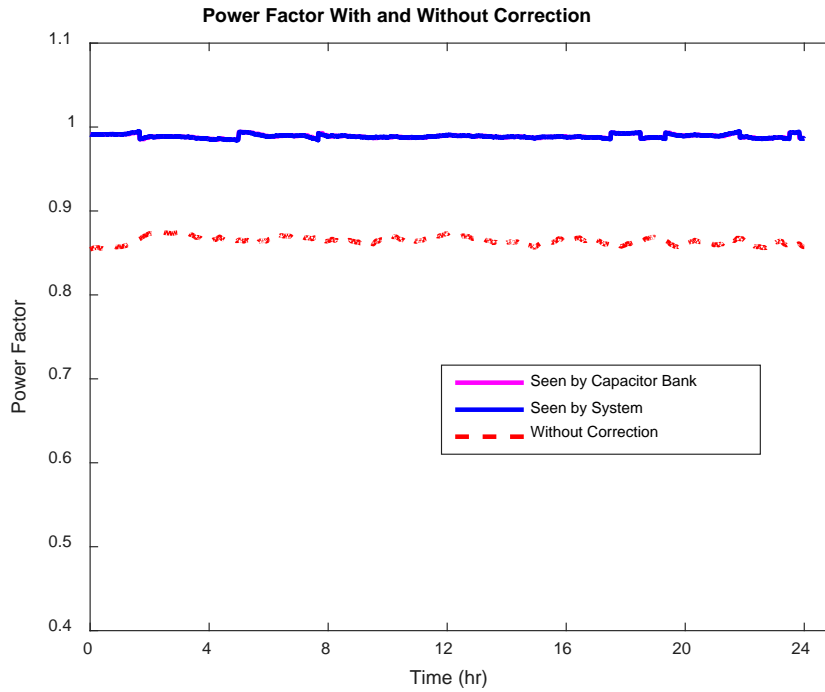


Figure 6: Increase in Power Factor

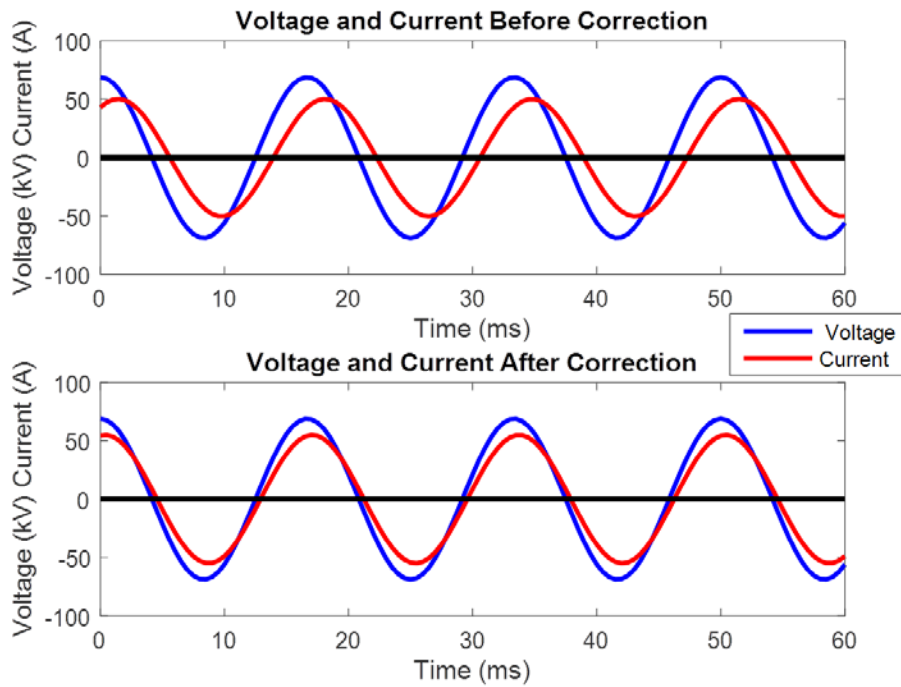


Figure 7: Phase Angle Correction

The final plot, Figure 7, represents the phase shift that the current undergoes which is the cause of the decrease in reactive power. With a current closer in phase with the voltage, the reactive power consumption decreases.

4 Ethical Analysis

When approaching an engineering challenge, an engineer must take into account three different ethical considerations. First, an engineer must understand the motivation behind a project. Why is the project important? For us, we must understand why we built a simulation for reactive power compensation and the impact that it could have. Secondly, an engineer must realize the values behind an ethical engineer and an ethical project. What thinking and attitude must be maintained in order for success? Lastly, an engineer must understand and make known any safety concerns or other risks towards all individuals involved.

4.1 Ethical Justification

Our simulation of reactive power compensation provides both economic and environmental benefits while maintaining a stable grid. All three of these impacts are beneficial not only for the parties directly involved, but also for the rest of the population affected by reactive power compensation. The main economic benefit is seen by the industrial consumer who saves money by receiving fewer reactive power penalties and by paying for less total power. Power companies as well can see savings due to the decrease in cost of operation caused by lower demand, especially since the power companies pay for reactive power draw from residential areas. When the power companies save money, a decrease in electricity costs could potentially be seen by residential, commercial and industrial consumers.

The environmental impact of implementing reactive power compensation is a global benefit. Currently, roughly 66% of the electricity in the United States is produced using fossil fuels, utilizing processes that are very harmful for the environment. While renewable energy production is on the rise, it is equally beneficial to decrease the overall consumption of electricity, thereby decreasing the amount of fossil fuels burned. If implemented on a large scale, reactive power compensation could account for a significant decrease in CO₂ emissions, potentially having a global impact.

4.2 Ethics, Our Project and a Good Engineer

Every engineer has a responsibility to create the best solution that they can, all while doing so in an ethical manner. For our project, we realized the impact that could be seen if consumers used our tool to design power factor correcting capacitor banks. For this reason, our simulation was made to be as accurate and adaptable as possible. We had to build a tool that is reliable and that will not pose a threat along any part of the process, especially during a physical implementation. We created an adaptable solution in order to cater to all different types of loads, whether large or small. In doing so, we can reach a larger audience and have a broader impact.

Throughout this project, we have realized the importance of teamwork and communication. In dividing up work, we were able to cover more ground and play to our strengths, all the while building trust in the reliability of the other. In having such a small team, each member was held accountable for the work that was done and the impact that they made on the project. The key to successful division of labor was to maintain

constant communication. In doing so, we were able to work efficiently and effectively, picking right up where the other left off. Communication was also key in the relationship with our advisor. Weekly update meetings allowed time for questions and guidance, making sure that we remained on schedule and on point.

4.3 Risks

The main risk with this project is seen through a physical implementation of a variable capacitor bank. If an industrial consumer uses our tool to create a capacitor bank and attach it to the high voltage load, there is danger involved. Working with high voltages requires significant knowledge of power transmission and an understanding of the safety requirements of that sort of work. Even with proper training, accidents can still happen, therefore caution must be used when installing a physical representation of our simulation results.

Since our project is a simulation, there is no risk in running the simulation or fixing the code. However, there is the risk that the simulation produces an incorrect result. If this occurs and a physical implementation is modeled after an incorrect model, there is the possibility of danger and system failure which could eventually lead to negative economic impacts. We built our simulation to work for most loads and to recognize loads for which it will not function properly. Unfortunately, errors are still a possibility and it is our responsibility to recognize any issues with the simulation.

There is no risk to people who will not be implementing a dynamic capacitor bank. In fact, the risk of grid instability, which could affect a much larger population, has been mitigated. Grid instability could lead to power surges or other unexpected fluctuations in the power grid. With reactive power compensation, the grid will remain stable throughout all hours of the day, reducing any worry of unsafe or inconvenient conditions.

5 Impacts

As mentioned earlier, the potential impact of our project is threefold. The economic benefits of reactive power compensation remain within the power companies and the industrial consumers. On the other hand, the environmental benefits and the maintenance of grid stability can impact a massive population. In this section, we provide a more in depth analysis of these impacts along with relevant examples.

5.1 Economic Impact

The main monetary beneficiary will be the industrial consumer who saves money at two stages. First, with reactive power compensation, the consumer will have to pay for less total reactive power. Unlike residential or commercial loads, the industrial consumer must pay for its reactive power consumption. By reducing the reactive power, they reduce the cost. The second stage includes savings due to reactive power penalties. Industrial consumers must pay penalties for loads running at below a power factor of 0.95. If a consumer is able to increase and maintain the power factor above 0.95, they will avoid all penalties. The below example shows the economic impact for a 1000kW system with a power factor increased from 0.7 to 0.9. This correction yields a decrease in reactive power of 537kVAR.

Table 2: Reactive Power Savings Example

Power Factor	0.7	0.9
Apparent Power	1429 kVA	1111 kVA
Reactive Power	1021 kVAR	484 kVAR

Reactive Power Penalty Savings:

$$537 \text{ kVAR} \left(\frac{\$1.10}{\text{kVAR}} \right) = \frac{\$590.70}{\text{month}} * 12 \text{ months} = \$7,088.40$$

Other Energy Cost Savings:

$$\begin{aligned} & \left(1 - \frac{PF_{old}^2}{PF_{new}^2} \right) * (P) * (\text{Energy Loss Before Correction}) * (\text{Price}) \\ & = \left(1 - \frac{0.7^2}{0.9^2} \right) * (1000\text{kW}) * (0.05) * \left(\frac{\$0.10}{\text{hour}} \right) * 8760 \text{ hours} = \\ & \qquad \qquad \qquad \$17,305.40 \text{ per year} \end{aligned}$$

Total Savings Per Year:

$$\$7,088.40 + \$17,305.40 = \$24,393.80$$

The savings will likely not be that high for every system since the jump in power factor is typically not as large as 0.2, but the example shows the potential for economic savings.

For most systems, the payback period lies between two to three years, which makes reactive power compensation an attractive option to industrial consumers.

5.2 Environmental Impact

As reactive power consumption, and therefore apparent power consumption, decreases, the power needed from power plants, many of which still rely on fossil fuels, is lessened.

By producing less energy, power plants produce less harmful CO₂ emissions. The following calculations show the CO₂ savings based off of the 537kVAR savings in the example above.

$$\left(1 - \frac{PF_{old}^2}{PF_{new}^2} \right) * (P) * (\text{Energy Loss Before Correction}) * (\text{Carbon Price})$$

$$= \left(1 - \frac{0.7^2}{0.9^2}\right) * (1000kW) * (0.05) * \left(\frac{1.222lbs\ CO_2}{kWh}\right) * 8760\ hours =$$

$$\frac{211106lbs\ CO_2}{2000\ lbs\ per\ ton} = 105.553\ tons\ of\ CO_2\ emmissions\ saved\ per\ year$$

The carbon savings of over 100 tons of CO2 per year translates to saving just under 4500 trees from being cut down each year. If reactive power compensation is effectively implemented on a large scale, the environmental benefits would be substantial.

5.3 Grid Stability

As the number of home solar systems increases, the load demand throughout the day slowly changes. As the “Duck Curve” shows in Figure 8, throughout the day, the load dips during peak sun hours then experiences a very sharp peak during the evening when the sun goes down and electricity consumption increases.

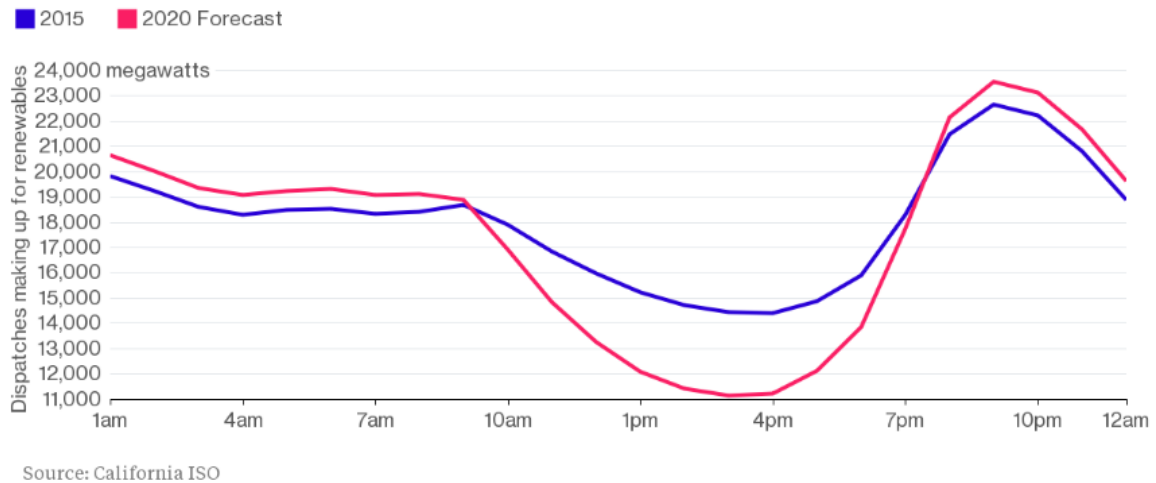


Figure 8: The duck Curve

This very quick spike in load demand is expensive and can lead to instability in the system. If reactive power compensation is applied on a large scale, the apparent power supplied will be reliant only on the changing real power. Therefore, the negative results of the quick spike will be depleted and the grid will remain stable.

6 Conclusion

Our project successfully demonstrates reactive power compensation techniques utilizing a MATLAB simulation. The simulation is to be used by industrial consumers to model their loads and to increase the power factor to a desired level, thereby decreasing the amount of reactive power of their systems. The simulation is able to take in a wide variety of loads and model any typical transmission line in order to meet customer need. If successfully implemented, there will be economic benefits to the customer and power companies as well as environmental benefits due to the decrease in carbon dioxide produced by burning fossil fuels.

7 References

- Celtekligil, U., "Capacitive power factor and power quality correction of a light rail transportation system," in *ELMAR, 2008. 50th International Symposium*, vol.2, no., pp.415-418, 10-12 Sept. 2008
- Cavallini, Andrea, Giovanni Mazzanti, and Gian Carlo Montanari. "Design Of Shunt Capacitor Circuits For Power Factor Compensation In Electrical Systems Supplying Nonlinear Loads: A Probabilistic Approach." *IEEE Transactions On Industry Applications* 34.4 (1998): 675-681. *Applied Science & Technology Source*. Web. 21 Nov. 2015.
- Choudhry, M.A.; Zada, W., "A simplistic technique for power factor compensation capacity calculation in medium size industry," in *Power Engineering Conference, 2005. IPEC 2005. The 7th International* , vol., no., pp.1-220, Nov. 29 2005-Dec. 2 2005
- Cole, Bernard. "Smartening up Power Monitoring." *EETimes* 2015. Web. 21 Nov. 2015.
- Mork, B.A., "Parameters for modeling transmission lines and transformers in transient simulations," *Power Engineering Society Winter Meeting (2001): 716-717. IEEE*. Web. 15 Nov. 2015.
- Pires, Dulce F., Vitor F. Pires, Carlos H. Antunes, and Antonio G. Martins, "Passive and Active Anti-Resonance Capacitor Systems for Power Factor Correction," *Power Electronics and Motion Control Conference, 2006. EPE-PEMC 2006. 12th International*, (2016): 1460-1465. *IEEE*. Web. 17 Nov. 2015
- Saied, Mohamed Mostafa. "Optimal Power Factor Correction." *IEEE Transactions On Power Systems* 3.(1988): 844-851. *Applied Science & Technology Source*. Web. 21 Nov. 2015.
- Silver, H. Ward, "Power Factor And Phase Angle." *Ost* 99.11 (2015): 72-73. *Applied Science & Technology Source*. Web. 19 Nov. 2015.
- Zohaa, Ahmed .F., "Maintaining a Good Power Factor and Saving Money for Industrial Loads." *Industrial Electronic* 53.2 (2006): 710-711. *IEEE Transactions*. Web. 17 Nov. 2015

8 Appendices

Appendix A: Main Code

```
clear; clc; clf; cla; close all;

% For loads, have a certian Kwhr usage and have the powerfactor within
a
% desired range (average powerfactor fro residencial area 0.65-0.75)
%% fun for X number of hours

total_time = (24); % in hours
time_min = total_time*60; % min
time_sec = total_time*3600; % seconds
time_check = 600; % seconds
load_changes = 600;
num_cap_check = time_sec/time_check + 1; % number of times we check and
change cap

ccl_index = 1; % stands for 'cap check loop index'

rate = time_sec / load_changes;
%% creating load vaiance(??)
line_variance_r = (rand(time_sec,1)/100);
line_variance_l = (rand(time_sec,1)/10);

C_parallel = zeros(1,num_cap_check);
Cinitial = 0;

% for load build
Rload_bounds =[45000, 70500];
Lload_bounds = [58, 75];

%for day load
R_bounds = [11600,11900; ... % midnight -2am
    12000, 12600;... & 2-4a
    11650, 12500;... % 4-6a
    10500, 11350;... % 6-8a
    09250, 10100;... % 8-10a
    09100, 09650;... % 10a-12p
    09300, 09800;... % 12-2p
    09350, 09850;... % 2-4p
    09200, 09725;... % 4-6p
    09000, 09500;... % 6-8p
    08500, 09000;... & 8-10p
    09000, 10000];...& 10-12p

L_bounds = [17.50 20.30; ... % midnight -2am
    17.00, 20.00;... & 2-4a
    16.50, 20.20;... % 4-6a
    14.50, 18.40;... % 6-8a
    13.30, 16.80;... % 8-10a
```

```

13.00, 16.20;... % 10a-12p
12.85, 16.40;... % 12-2p
13.35, 16.55;... % 2-4p
13.15, 16.30;... % 4-6p
12.85, 16.00;... % 6-8p
12.15, 15.30;... & 8-10p
12.90, 16.75];...& 10-12p

%[Rload, Lload] = load_build(time_sec, Rload_bounds, Lload_bounds,
load_changes);
[Rload, Lload] = day_load(total_time, R_bounds, L_bounds, rate);

t = 0:.0005:.06;
PF_des = .99;

%% Define transmission Line (3/9/16) (wont change with loop)

length_tm = 70; % in kilometers, total length of transmission line
Vamp = 69000;
conv = (pi)/180;
f = 60;
w = 2*pi*f;
thetav= 0;
%% Defining arrays such that they dont changes size everyloop

% current and voltage
%vin = (num_cap_check;:);
%current = (num_cap_check; 600);
Iamp = (num_cap_check);
%new_current = (num_cap_check);
thetai = (num_cap_check);

% power factor
pf_new = (num_cap_check);
pf_orig = (num_cap_check);
pf_new_nc = (time_sec);
pf_new_total = (time_sec);

%Impedances
Imped_new = (time_sec);
X_nc = (time_sec);
R_nc = (time_sec);
R = (time_sec);
X = (time_sec);
Rload_new = (time_sec);
Xload_new = (time_sec);
Rtran = (time_sec); Ltran = (time_sec);

% variables used for keeping track of power
S_total_nc = (time_sec);
Q_total_nc = (time_sec);
P_total_nc = (time_sec);
S_total_cap = (time_sec);
P_total_cap = (time_sec);

```

```

Q_total_cap = (time_sec);
P = (ccl_index);

%C Bank info
number_of_caps = 0;
caps_on = (time_sec +1);
caps_on(1) = 0;
%% Actual function:

for time_run = 1:time_sec; % total run time in seconds
    % check capacitance ever 10 minutes? we can change that
    %% Define transmission Line (will change with loop)
    per_length_line_imp = (0.0679+line_variance_r(time_run)) +...
        (0.289+line_variance_l(time_run))*1i;

    [line_imp]= trans_build(length_tm, per_length_line_imp); % Ohm/Km

    lin_impedence = sqrt(((real(line_imp))^2) + ((imag(line_imp))^2));
    %% define variables
    vin(ccl_index,:) = Vamp*cos(2*pi*f*t + thetav*conv);

    Rtran(time_run) = real(line_imp);
    Ltran(time_run) = (imag(line_imp)/w);

    if (mod(time_run-1,time_check) == 0)

        % total load from the inital stage, starting with zero
        capacitance

        if ccl_index == 1
            Cinitial = 0;
            R(time_run) = Rload(time_run) + Rtran(time_run);
            X(time_run) = (Lload(time_run) + Ltran(time_run))*w;
        else
            Cinitial = C_parallel(ccl_index-1);
            % for R and L we need to first find the to
            [Rload_new(time_run),Xload_new(time_run)] =
load_calc_parallel(Rload(time_run),...
                    Lload(time_run),Cinitial,f);

            R(time_run) = Rload_new(time_run) + Rtran(time_run);
            X(time_run) = Xload_new(time_run) + (Ltran(time_run)*w);

        end

        %% Calculate current

    [impedence,thetai(ccl_index)]=load_calc(R(time_run),X(time_run),thetav)
;

```

```

[Iamp(ccl_index),current(ccl_index,:)]=curr_calc(Vamp,impedence,...
        thetai(ccl_index),f,t);

pf= power_factor(thetav,thetai(ccl_index));
pf_orig(ccl_index) = pf;
%disp(sprintf('Initial power factor = %f \n', pf));

%% Calculate power factor

%Q and P calculations to be used with desired P.F. to find cap
values
P(ccl_index) = Vamp*Iamp(ccl_index)*cosd(thetav-
thetai(ccl_index));

%theta_S = acosd(PF_des);
%Qnew = P(ccl_index)*tand(theta_S)

S_old = P(ccl_index)/pf;
Q_old = sqrt(((S_old)^2) - (P(ccl_index)^2));

% from example problem
S_desired = P(ccl_index)/PF_des;

Q_desired = sqrt(((S_desired)^2) - (P(ccl_index)^2));

Qc = Q_old - Q_desired; % reactive power required from the
capacitors

%%

Xc = (Vamp^2)/Qc;
Cval=1/(2*pi*f*Xc);

%disp(sprintf('Value of capacitance needed (in Farad): %d \n'
,Cval));

% Cal function to calculate the number of caps needed to turn
on and the
% change in capactiance value we will see
[number_of_caps,C_added] = cap_change(Cval);
%disp(sprintf('Number of capacitors needed: %d
\n',number_of_caps));
caps_on (time_run +1) = caps_on(time_run)+ number_of_caps;

C_parallel(ccl_index) = Cinitial + C_added;

%since we calculated the necessary capacitance in parallel
[R_after_cap, X_after_cap]=
load_calc_parallel(Rload(time_run)...

```

```

        ,Lload(time_run),C_parallel(ccl_index),f);

R_final = R_after_cap + Rtran(time_run);
X_final = X_after_cap + (Ltran(time_run)*w);

[Imped_new(time_run), thetai_new] = load_calc(R_final,X_final,
thetav);

[new_Iamp,new_current(ccl_index, :)] = curr_calc(Vamp,
Imped_new(time_run), thetai_new, f, t);

pf_new(ccl_index) = power_factor(thetav, thetai_new);

pf_new_total(time_run) = power_factor(thetav, thetai_new);

%disp(sprintf('New power factor = %f \n\n\n', pf_new(ccl_index-
1)));
ccl_index = ccl_index + 1;

else % not chnaging the parrellel capacitance in this section
C = C_parallel(ccl_index-1);
caps_on (time_run +1) = caps_on(time_run);
% for R and L we need to first find the to
[Rload_new,Xload_new] =
load_calc_parallel(Rload(time_run),Lload(time_run),C,f);

R(time_run) = Rload_new + Rtran(time_run);
X(time_run) = Xload_new + (Ltran(time_run)*w);

[Imped_new(time_run), thetai_new] =
load_calc(R(time_run),X(time_run), thetav);

[new_Iamp,new_current(ccl_index, :)] = curr_calc(Vamp,
Imped_new(time_run), thetai_new, f, t);

pf_new_total(time_run) = power_factor(thetav, thetai_new);

end

%% calculate powerfactor over whole 'time' without correction

% '_nc' = no correction values

% for R and L we need to first find the to
% [Rload_nc,Xload_nc] =
load_calc_parallel(Rload(time_run),Lload(time_run),0,f);

R_nc(time_run) = Rload(time_run);% + Rtran(time_run);
X_nc(time_run) = (Lload(time_run)*w);% + Ltran(time_run))*w;

```

```

    [Imp_new_nc(time_run), thetai_new_nc] =
load_calc(R_nc(time_run),X_nc(time_run), thetav);
    [new_Iamp_nc ,new_current(ccl_index,:)] = curr_calc(Vamp,
Imp_new_nc(time_run),thetai_new_nc,f,t);

    pf_new_nc(time_run) = power_factor(thetav,thetai_new_nc);

    %% lets calculate some statistics about our loads!!

    % three types of power after corrected by the capacitor bank
    P_total_cap(time_run) = Vamp*(new_Iamp) * cosd(thetav -
thetai_new);
    Q_total_cap(time_run) = Vamp*(new_Iamp) * sind(thetav -
thetai_new);
    S_total_cap(time_run) = Vamp*(new_Iamp) ;

    % what happens when not corrected by capacitor bank
    P_total_nc(time_run) = Vamp*(new_Iamp_nc) * cosd(thetav -
thetai_new_nc);
    Q_total_nc(time_run) = Vamp*(new_Iamp_nc) * sind(thetav -
thetai_new_nc);
    S_total_nc(time_run) = Vamp*(new_Iamp_nc) ;

end

%% Plots and such
for i = 1:num_cap_check-1
    pf_seen_by_cap_change((i-1)*time_check +1 : i*time_check) =
pf_new(i); % loop

end

ind = time_run;
plot_index = (ind);
plot_hour = (ind);
for index = 1:ind
    plot_index(index) = index;
    plot_hour(index) = index/3600;
end

[PE_total, average_total] = average_calc (PF_des, time_run,
pf_new_total);
[PE_cc, average_cc] = average_calc (PF_des, time_run,
pf_seen_by_cap_change);
[PE_nc, average_nc] = average_calc (PF_des, time_run, pf_new_nc);

%% figure 1 - voltage and current and the beginning

figure(1)
vin_s = vin(1,:)/1000;
vin_s2 = vin(num_cap_check,:)/1000;

```

```

x = zeros(length(vin(1,:)));

p1 = (vin(1,:).*current(1,:)) / 1000;
p2 = (vin(num_cap_check,:).*current(num_cap_check-1,:)) / 1000;
t2 = t*1000;

subplot(2,1,1)
plot(t2,vin_s(1,:), 'b', t2, 10*current(1,:), 'r',
t2,x,'k','LineWidth',2);
title('Voltage and Current Before Correction');
xlabel('Time (ms)');
ylabel('Voltage (kV) Current (A)');
legend(' Voltage', 'Current')
%xlim ([0 0.05])

subplot(2,1,2)
plot(t2,vin_s(1,:), 'b', t2, 10*current(num_cap_check -1,:),
'r', t2,x, 'k', 'LineWidth', 2);
title('Voltage and Current After Correction');
xlabel('Time (ms)');
ylabel('Voltage (kV) Current (A)');
legend(' Voltage', 'Current')
xlim ([0 60])

%% figure 3 - power factor graph
figure(3)
plot(plot_hour, pf_seen_by_cap_change, 'm', plot_hour, pf_new_total,
'b', ...
plot_hour, pf_new_nc, '--r', 'LineWidth', 2)
title('Power Factor With and Without Correction')
xlabel('Time (hr)');
ylabel('Power Factor');
ylim([0.4 1.1])
legend('Seen by Capacitor Bank', 'Seen by System', 'Without
Correction')
ax = gca;
ax.XTick = [0 4 8 12 16 20 24];

%% figure 5 impedance graph
figure(5)

ind= length(Imped_new);
scaled_imp = (ind);
for i =1:length(Imped_new)
scaled_imp(i) = Imped_new(i)/1000;
end

plot(plot_hour, scaled_imp)
title('Day Load Total Impedance over Time');
ylabel('Total Load (kOhms)');
xlabel('Time (hr)');
%ylim([0 80])

```

```

ax = gca;
ax.XTick = [0 4 8 12 16 20 24];

%% figure 6 - individual inductance and resistance ?
% figure(6)
% subplot(2,1,1)
% plot(L)
% title('Inductance over time in Henry')
%
% subplot(2,1,2)
% plot(R)
% title('Resistance over time in Ohms')

%% figure 7/8/9 - Reactive and apparent power difference
figure(7)

differencesS = (time_run);
differenceP = (time_run);
differenceQ = (time_run);

for index = 1:time_run
    differencesS(index) = (S_total_nc(index) - S_total_cap(index));
    differenceP(index) = (P_total_nc(index) - P_total_cap(index));
    differenceQ(index) = (Q_total_nc(index) - Q_total_cap(index));
end

%scale Apparent Powers
ind = length(S_total_nc);
S_total_nc_s = (ind);
S_total_cap_s = (ind);
for i = 1 : ind
    S_total_nc_s(i) = S_total_nc(i)/1000;
    S_total_cap_s(i) = S_total_cap(i)/1000;
end

plot(plot_hour, S_total_cap_s, 'bl',plot_hour, S_total_nc_s, '--
r','LineWidth',2)
title('Apparent Power With and Without Correction');
xlabel ('Time (hr)');
ylabel ('Apparent Power (kVA)');
%ylim([0 200]);
legend('Power Factor Correction', 'No Power Factor Correction')
ax = gca;
ax.XTick = [0 4 8 12 16 20 24];

% figure 8 - Reactive Power

figure(8)

%scale Reactive Powers
ind = length(Q_total_nc);
Q_total_cap_s =(ind);
Q_total_nc_s = (ind);
for i = 1 :ind
    Q_total_nc_s(i) = Q_total_nc(i)/1000;

```



```

    Q_total_cap_s(i) = Q_total_cap(i)/1000;
end

plot(plot_hour, Q_total_cap_s, 'b',plot_hour, Q_total_nc_s, '--
r','LineWidth',2)
title('Reactive Power With and Without Correction');
xlabel ('Time (hr)');
ylabel ('Reactive Power (kVAR)');
%ylim([0 180]);
xlim([0 24]);
legend('Power Factor Correction', 'No Power Factor Correction')
ax = gca;
ax.XTick = [0 4 8 12 16 20 24];

% figure 9 - real power

figure(9)
%scale real Powers
P_total_cap_s = (length(P_total_nc));
for i = 1 :length(P_total_nc)
    P_total_cap_s(i) = P_total_nc(i)/1000;
end

plot(plot_hour, P_total_cap_s, 'b','LineWidth',2)
title('24hr Day Load - kW');
xlabel ('Time (hr)');
ylabel ('Real Power (kW)');
%ylim([230 280]);
xlim([0 24]);
ax = gca;
ax.XTick = [0 4 8 12 16 20 24];

%% figure 10 - number to caps on

ind = length(caps_on);
caps_on_new = (ind-1);
for i = 2:ind
    caps_on_new(i-1) = caps_on(i);
end

figure (10)
plot(plot_hour, caps_on_new,'LineWidth',2)
title('Number of Capacitors On');
xlabel('Time (hr)');
ylabel('Number of Capacitors');
ylim([0 10]);
xlim([0 24]);
ax = gca;
ax.XTick = [0 4 8 12 16 20 24];

```

Appendix B: Functions

```
function [line_impedence] = trans_build(length,perlen_imp)
```

```
%inputs: will inputs of function  
%voltage = 69000; % volts  
%length = 70; % kilometers  
%perlen_imp = 0.4 + 1i*0.3; % Ohms/Km
```

```
line_impedence = length*perlen_imp;  
%real_imp = real(line_impedence);  
%imag_imp = imag(line_impedence);
```

```
function [pf] = power_factor(thetav, thetai)
```

```
pf=cosd(thetav-thetai);
```

```
end
```

```
function [Res,Xreac] = load_calc_parallel(R, L , C_par,f)
```

```
w = 2*pi*f;  
if C_par==0  
    Res = R;  
    Xreac = w*L;  
else  
    load = ((R + i*w*L)*(1/(i*w*C_par)))/( R + i*w*L +  
    (1/(i*w*C_par))); % complex nummber  
    Res = real(load);  
    Xreac = imag(load);
```

```
end
```

```
end
```

```
function [impedance, thetai] = load_calc(R,X, thetav)
```

```
% fixed function to output theta in degrees
```

```
if X > 0  
    impedance = sqrt(R^2 + X^2);  
    thetaZ = atand(X/R);  
    thetai=thetav-thetaZ;
```

```
else
```

```

        if X <= 0 % if reactance from the capacitor is bigger than
            % the reactance from the inductor, we would need to add
            negative capacitacne
            %disp ('Can NOT be Fixed with Parallel CapacitanceX was
            negative')
            impedance = sqrt(R^2 + (X)^2);
            thetaZ = atand(X/R);
            thetai=thetav-thetaZ;
        end
    end
end
end

```

```

function [R_output, L_output] = load_build(time,R, L, num_load)
n = 16; % order of filter

```

```

R_out_1 = R(1) + (R(2)-R(1)).*rand(num_load + (n-1),1);
L_out_1 = L(1) + (L(2)-L(1)).*rand(num_load + (n-1),1);

```

```

% R_out(1) = R_out_1(1);
% R_out(2) = .5*(R_out_1(1) + R_out_1(2));
% R_out(3) = (1/3)*(R_out_1(1) + R_out_1(2) + R_out_1(3));
%
% L_out(1) = L_out_1(1);
% L_out(2) = .5*(L_out_1(1) + L_out_1(2));
% L_out(3) = (1/3)*(L_out_1(1) + L_out_1(2) + L_out_1(3));

```

```

for i = 1: (length(R_out_1) - (n-1))
    R_out(i) = (1/n)*(R_out_1(i) + R_out_1(i+1) + R_out_1(i+2) +
    R_out_1(i+3));
    L_out(i) = (1/n)*(L_out_1(i) + L_out_1(i+1) + L_out_1(i+2) +
    L_out_1(i+3));
end

```

```

sections = time/num_load;

```

```

n = length(R_out);

```

```

out=1;
for i = 2:n+1
    if i <= n
        slopeR = (R_out(i) - R_out(i-1))/ sections;
        slopeL= (L_out(i) - L_out(i-1))/ sections;

        for j = 1:sections
            R_output(out) = R_out(i-1)+slopeR*(j-1);
            L_output(out) = L_out(i-1)+slopeL*(j-1);
            out = out +1;
        end
    else

```

```

        for j = 1: sections
            R_output(out) = R_out(i-1);
            L_output(out) = L_out(i-1);
            out = out + 1;
        end
    end
end
figure(20)
subplot(2,1,1)
plot(R_output)
subplot(2,1,2)
plot(L_output)

function [figure_number] = graph_hours[data, figure_number]

for i = 1: length(data)
    time_sec(i) = i;
    time_hours(i) = i/3600;
end
figure_number = figure_number + 1;

plot(figure_number,

%clc; clear;
%% what function should do
% new function for the second load we want it to be
% lowest between the hours from 12am - 6am,
% slightly higher between the 6am - 4pm
% highest from 4pm to 9pm (people get off work, cok dinner, wash
clothes,
% turn on lights and tvs, ect.
% drops down again between 9pm-12am

%[min_low max_low min_mid max_mid min_hi max_hi]

%% Function
%(most of the inital things should ne passed into the fucnction for
Initial_code

function [Rout, Lout] = day_load(total_time,R_bounds, L_bounds, rate)

% R_bounds = [12, 14; 13, 15; 8, 9; 20, 22];
% L_bounds = [9, 10; 10, 12; 20, 22; 15, 17];
% rate = 60; % seconds in between load changes
% total_time = 2;

```

```

diff_loads = length(R_bounds);
if diff_loads ~= length(L_bounds)
    disp (' error in L/R bounds dimenstions')
end

time_sec = total_time*3600;
f = 60;
w = 2*pi*f;
percent_time_per_load = [(2/24)];

seconds_per_load = percent_time_per_load*diff_loads*time_sec;
num_load_change_per_section = seconds_per_load/rate;

% probably need to pass in the total number the hours the simulation is
% running.

aver = 24;

% assuming load change once per minute
q = diff_loads * num_load_change_per_section + (aver-1);
t = 1;
% number of load changes in previous load build, in this one, number of
% hours for each type of load, (low load twice)

for x = 1:diff_loads
if x < diff_loads
    R_in = R_bounds(x,1) + (R_bounds(x,2)-
R_bounds(x,1)).*rand(num_load_change_per_section,1);
    L_in = L_bounds(x,1) + (L_bounds(x,2)-
L_bounds(x,1)).*rand(num_load_change_per_section,1);
else if x == diff_loads
    R_in = R_bounds(x,1) + (R_bounds(x,2)-
R_bounds(x,1)).*rand(num_load_change_per_section+ (aver-1),1);
    L_in = L_bounds(x,1) + (L_bounds(x,2)-
L_bounds(x,1)).*rand(num_load_change_per_section+ (aver-1),1);
    end
end

for w = 1:length(R_in)
    real_impedence(t) = R_in(w);
    imag_impedence(t) = L_in(w);
    t= t+1;

end
end

%
% j=1;
% k=1;
% l=1;

```

```

% for i=1:(index +(aver -1))
%     if (i <= num_load_change_per_section(1))
%         real_impedance(i) = R_mor(i);
%         imag_impedance(i) = L_mor(i);
%     else if ((num_load_change_per_section(1) < i) && (i <=
(num_load_change_per_section(2) + num_load_change_per_section(1))))
%         real_impedance(i) = R_aft(j);
%         imag_impedance(i) = L_aft(j);
%         j=j+1;
%     else if
(((num_load_change_per_section(2)+num_load_change_per_section(1)) < i)
&& (i <= (num_load_change_per_section(2) +
num_load_change_per_section(1) + num_load_change_per_section(3))))
%         real_impedance(i) = R_eve(k);
%         imag_impedance(i) = L_eve(k);
%         k=k+1;
%     else if (((num_load_change_per_section(2) +
num_load_change_per_section(1) + num_load_change_per_section(3)) < i)
&& ...
%         (i <= (num_load_change_per_section(2) +
num_load_change_per_section(1) + num_load_change_per_section(3) +
num_load_change_per_section(4) + (aver-1))))
%         real_impedance(i) = R_nig(l);
%         imag_impedance(i) = L_nig(l);
%         l=l+1;
%     end
% end
% end
% end
%% implement moving average filter

for temp = 1: ((length(real_impedance)) - (aver -1))
    value_r = 0;
    value_l = 0;
    for inner = 0:aver-1
        value_r = value_r + (real_impedance(temp +inner));
        value_l = value_l + (imag_impedance(temp+inner));
    end
    real_impedance_1(temp) = (1/aver)* value_r;
    imag_impedance_1(temp) = (1/aver)* value_l;
end

%% interpolate values:
n = length(real_impedance_1);
sections = time_sec/n;
out=1;
for i = 2:n+1
    if i <= n
        slopeR = (real_impedance_1(i) - real_impedance_1(i-1))/
sections;
        slopeL= (imag_impedance_1(i) - imag_impedance_1(i-1))/
sections;

```

```

        for j = 1:sections
            Rout(out) = real_impedence_1(i-1)+slopeR*(j-1);
            Lout(out) = imag_impedence_1(i-1)+slopeL*(j-1);
            out = out +1;
        end
    else
        for j = 1: sections
            Rout(out) = real_impedence_1(i-1);
            Lout(out) = imag_impedence_1(i-1);
            out = out +1;
        end
    end
end
end

figure(19)
subplot(2,1,1)
plot(Rout)
subplot(2,1,2)
plot(Lout)

for temp=1:length(Rout)
    mag_impedence(temp) = sqrt((Rout(temp)^2) + (Lout(temp)^2));
    phase_impedence(temp) = (0 - atand(Lout(temp)/Rout(temp)));
end

figure(20)
subplot(2,1,1)
plot(mag_impedence)
subplot(2,1,2)
plot(phase_impedence)

end % function

function [curmag,current] = curr_calc(Vmag, impedance, thetai, f, t)
conv = (pi)/180;
curmag=Vmag/impedence;
current=curmag.*cos(2*pi*f*t + thetai*conv);

end

function [caps_on,cap_value] = cap_change(capacitance_needed)
f = 60;
w = 2*pi*f;
Qrated = 20000;
Vamp = 69000;
C = Qrated/((Vamp^2)*w);

```

```
size = C;  
%size = ((1*(10^-8)));  
caps_needed = round(capacitance_needed/size);  
  
caps_on = caps_needed;  
cap_value = (size*caps_on);  
  
end
```