2002

Design of a prototype personal static var compensator

Scott Alan Zemerick
West Virginia University

DESIGN OF A PROTOTYPE PERSONAL STATIC VAR COMPENSATOR

by

Scott Alan Zemerick

Thesis submitted to the College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

Approved by

Dr. Bojan Cukic, Ph.D
Dr. Ali Feliachi, Ph.D
Dr. Powsiri Klinkhachorn, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia
2002

Keywords: Static Var Compensator, Fuzzy Control, Var Controller

ABSTRACT

DESIGN OF A PROTOTYPE PERSONAL STATIC VAR COMPENSATOR (PSVC)

by Scott Alan Zemerick

The focus of this thesis is the design and implementation of a *personal* static var compensator (PSVC) for distributed var control through load power factor correction. The PSVC demonstrates the two key benefits of power factor correction, which include decreased power costs and increased system capacity. The PSVC prototype consists of two types of branches – a TSC branch and a TCR branch. A microprocessor is responsible for calculating the load displacement power factor (PF_D) and for executing the fuzzy logic control scheme for the two branches. The PSVC was found to reduce the RMS current drawn by a 55-watt AC motor by 25% while raising its PF_D by 40% to 0.99 lagging. The expected quick rate of return of installation costs is attributed to the PSVC's low initial cost and its ability to reduce tariffs for reactive power consumption.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to Dr. Klinkhachorn for his extremely valuable guidance and suggestions throughout this project. His technical assistance and personal advice have been invaluable. A special thanks to Dr. Cukic for serving on my examining committee and for providing me with valuable career advice. Also, special thanks to Dr. Feliachi for serving on my examining committee and for allowing me to be a part of the research group.

This thesis is dedicated to my parents because if it were not for their support and encouragement, this work would not exist. Special gratitude also goes to my brother and girlfriend for their perpetual encouragement.

Funding for this project was provided by the National Science Foundation under grant ECS-9870041 and a DOE/EPSCoR WV state Implementation Award.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1 The Static Var Compensator (SVC)

The focus of this research has been on a particular FACTS (Flexible AC Transmission System) device – the Static Var Compensator (SVC). The SVC is a proven technology for power factor correction and reactive power compensation. Traditionally the SVC has been used as a shunt-connected device that offers voltage stability and load compensation to the power system at particular points such as transmission line midpoints or near varying loads. Since EPRI's (Electric Power Research Institute) release of the FACTs strategies in 1987, SVC's have grown in popularity and are well regarded in the power industry [1].

1.2 Effects Associated with a Poor Power Factor

Power utilities and their customers often concern themselves with three poor power factor effects that include *increased power costs*, *reduced system capacity*, and *diminished power quality*. Each effect is summarized below.

- **Increased Power Costs**
  - Higher kVA demand – The reactive power that is needed by inductive loads must be generated by the power utility. This increase in overall power will result in a higher power bill for the end consumer.
  - Low PF penalty – For reasons discussed below, it is in the power utility's best interest to offer power factor correction incentives to consumers when their power factor becomes low. These incentives take the form of tariffs.
- **Reduced System Capacity**
  - Less kW per kVA – Total power generated (kVA) is composed of two components:
    1. Kilowatts (kW): The useable power
    2. Kilovars (kVar): The unusable power

The desired power factor of 1.0 (unity) is obtained by reducing the costly generation of kVars, thus maximizing the kW. Maximizing kW results in more efficient power generation and delivery.

  - Higher Current Draw – Power factor and load current have an inverse relationship. If the power factor is low, then the current drawn will be high. Minimizing consumer load current will lessen the demand on the distribution feeder.
  - Less Room for Expansion – If the generation of kVars is significant then the pathway for real power transfer will become smaller making power delivery more difficult.

- **Diminished Power Quality**
  - Increased Voltage Drop – A poor power factor will yield an increased voltage drop because lagging current in inductive loads causes load voltages to fall.
  - Higher Distribution System Losses – Increasing the power factor to near unity can reduce distribution system losses. This is discussed in more detail in section 4.3.4.

1.3 Personal SVC (PSVC) Prototype and Its Applications

The topic of this thesis describes a "personal" static var compensator (PSVC) prototype that can be installed adjacent to a reactive load so that its displacement power factor can be optimized in real time. The displacement power factor (PF_D) is the ratio of real power to apparent power, only considering the fundamental frequency. Raising the PF_D will allow for a lower RMS current draw, decreased power costs, and increased system capacity.

The PSVC can be installed on a wide-variety of reactive loads including air conditioning units, refrigerators, escalators, elevators, or any application that relies on AC induction motors. For example, "efficient" induction motors that are found in residential refrigerators and air conditioning units have a typical power factor of 0.8 to 0.82. On any given day a billion induction motors are operating in elevators, escalators, machine tools, intake and exhaust fans, conveyors, pumps, and compressors [2]. Increasing the displacement power factor, thus the efficiency, will result in significant power savings of 15% - 20% when using these motors. The PSVC is targeted for three audiences: The *power utility company*, the *light-industrial or residential* *consumer*, and the *manufacturers of applications that rely on induction motors*. Each targeted audience is described in more detail.

- **The Power Utility Company:**

As previously discussed, the power utility company has an incentive to encourage consumers to maintain a near unity power factor. The power company could purchase PSVC's and install them at particular distribution locations – very near the customers' reactive loads. As a result, a lower reactive power demand will allow real power to be delivered more efficiently.

- **The Light-Industrial (Commercial) or Residential Consumer:**

The light-industrial or commercial consumer is concerned with minimizing costs due to tariffs imposed by the power utility. A customer who operates several AC induction motors (such as those in escalators and elevators) would be able to increase their efficiency by 15% - 20% by installing low-cost PSVC's. Air conditioning units and refrigerators, which are abundant in residential homes, would also be ideal PSVC applications. Nearly 25% of residential energy consumption is a result of refrigerators and air conditioners with 99.9% of American residences having at least one refrigerator and 72% of residences having an air conditioner [3]. Increasing the energy efficiency of residential and commercial air conditioners and refrigerators would offer significant energy savings to the consumer. Since standard air conditioners and refrigerators typically maintain power factors from 0.5 – 0.7, an approximate 30-50% increase in efficiency is possible [4]. Currently, residential customers are not charged based on their reactive power usage, but current technology does exist to implement this type of billing.

- **Manufacturers of Applications that Rely on Induction Motors:**

Manufacturers are constantly developing new techniques to increase the efficiency and lower the power consumption of their applications. Lower power consumption and more efficient operation are needed to attract consumers as well as meet government regulations.

One such government guideline is the *Energy Star* rating for applications that meet strict energy usage criteria. Due to its small size, the PSVC could easily be included into an existing energy-inefficient application design. Traditionally, fixed capacitors have been utilized in motor applications to provide a fixed power factor correction. The PSVC overcomes this lack of power factor correction flexibility by providing dynamic correction and a power factor closer to unity.

1.4 Research Goals

The goal of this research is to design a low cost distributed var controller that is capable of power factor displacement (PF_D) correction. The var controller design will be based on the static var compensator (SVC), which is a proven FACT's device for power factor correction and voltage stability. Verification simulations will be performed to help validate hardware control methods. The microprocessor – controlled prototype PSVC will be economical, robust, and "personal". The PSVC will contain two branches – a TSC and a TCR branch that are controlled by a fuzzy controller. It will be a completely self-contained device capable of being installed on a variety of reactive loads.

1.5 Thesis Organization

Chapter 2 of this thesis is a literature survey describing FACTS, and specifically, SVC operation. Some reviewed papers also describe microprocessor – controlled SVC prototype designs and implementations. Chapter 3 describes MATLAB/SIMULINK simulations and PSVC hardware design as well as PSVC software and fuzzy controller design. Chapter 4 describes an application of the PSVC and presents experimental results. In Chapter 5, conclusions, future work, and lessons learned are provided.

CHAPTER 2 - LITERATURE SURVEY

In this chapter a literature survey of topics related to the SVC operation and microprocessor control will be given. Improvements over existing work are presented at the conclusion of the chapter.

2.1 Fundamentals of SVC's

The static var compensator (SVC) is regarded as the first FACTS (Flexible AC Transmission System) devices [5]. Basin Electric Power Cooperative installed the first SVC in Nebraska in 1977 [6]. Since then, the SVC has become known for its ability to offer voltage stability to the power system and to compensate reactive loads. The SVC, however, does not provide a means for controlling real power flow directly.

In 1978, Gyugyi, Otto, and Putman presented the fundamentals of static load compensation in [7]. This early work described the theoretical background for thyristor – switched inductors (reactors) and capacitors. The paper describes the two appropriate circumstances when reactive load compensation should be considered: 1) the AC system cannot maintain the desired voltage and 2) it is too costly to have the AC system provide for the reactive power demand of the load.

The static compensator switching and control fundamentals are also described in [7]. It is important to ensure that the static compensators are switched at the appropriate times so that transients do not occur. For both capacitors and inductors, the appropriate switching times occur at the zero crossing of the current. Capacitors have a maximum theoretical switching-in delay of 1 cycle, assuming that the capacitor is precharged to the "incorrect" polarity. The capacitor can be switched out in a half-cycle. Inductors, however, are able to be both switched in and out within a maximum delay of a half-cycle. Capacitors are switched completely in or out of the system as needed; however, the current through the inductor can be controlled continuously providing a variable susceptance. Susceptance is defined as the ease with which

AC current will pass through capacitors or inductors. Delaying the closure of the inductor thyristor switch every half cycle can control this susceptance. This delay is commonly referred to as the firing angle, α. Although the inductor's thyristors can be fired at anytime after the zero crossing of the current, they can only be turned off through natural commutation, when the current reaches zero. Equation 2.1 from [7] illustrates the relationship between the firing angle α and the current through the inductor I_L. α = 0 is defined as the zero crossing of the current.

$$I_L = (V / \omega L)(1 - (2/\pi)\alpha - (1/\pi)\sin 2\alpha) \tag{2.1}$$

Two control strategies for SVC's are discussed in [7]. The first strategy is feedforward control, which repeatedly solves modeled equations to determine the optimum inductor firing angle and the number of capacitor banks needed. Feedforward control is a good choice when performing load compensation because at any time the load characteristics can be measured and the optimum compensating susceptance can then be calculated. The second strategy, feedback control, is useful for closed-loop control when minimization of error signals is the primary goal. Feedback control is useful when terminal voltages are being maintained because the error signal can represent the difference between a desired voltage and an actual voltage. Any change in the error signal will result in a change in the susceptance of the compensator.

The controllable susceptance of the inductor (B_L) can be defined as a function of the conduction angle and the reactance of the inductor (X_L). The conduction angle σ, opposite from the firing angle α, is defined as the time that the thyristors are turned on. Equation 2.2 from [8], shows this relationship.

$$B_L(\sigma) = (\sigma - \sin\sigma) / \pi X_L \tag{2.2}$$

Using the relationship $\sigma = 2(\pi - \alpha)$ also from [8], the controllable susceptance of the inductor in terms of α is shown in Equation 2.3.

$$B_L(\alpha) = [2(\pi - \alpha) - \sin 2(\pi - \alpha)] / \pi X_L \tag{2.3}$$

Furthermore, it is often useful to express the controllable inductor susceptance in terms of its counterpart, reactance. Reactance is defined as the opposition of AC current flow in capacitors and inductors. As shown in Equation 2.4, the inductor reactance can be found by inverting Equation 2.3. The reactance equation is related to Equation 2.1 in that the chosen firing angle will determine how much current is allowed to flow through the inductor.

$$X_L(\alpha) = [[2(\pi - \alpha) - \sin 2(\pi - \alpha)] / \pi X_L]^{-1} \tag{2.4}$$

2.2 The Adaptive Power Factor Controller (APFC)

An adaptive power factor controller (APFC) utilizing fixed and switched capacitors is described in [9] and [10]. Even though the described APFC does not make use of an inductor branch as in SVC's, the contributions to effective capacitor switching techniques are notable. The described APFC was first built and tested in the late 1980's. It was targeted for applications that consist of a dynamic reactive power demand such as industrial induction motors, transmission lines, and distribution feeders. In [10], the application of APFC's to wind farms is justified by the large amount of reactive power demanded by the wind turbine from the utility.

The APFC consisted of a certain amount of fixed capacitance along with a determined amount of switched capacitance that would be controlled dynamically as needed. As outlined in [9], capacitors should be installed in binary ratios so that the maximum amount of VARS can be supplied with the least number of capacitors. "For example, three switchable capacitors per phase lead to seven distinct steps of reactive power compensation (in addition to zero) [9]." The APFC utilized SCR's (silicon controlled rectifiers) with a back-to-back connected diode as the capacitor switching devices. The diode was used to insure that the SCR would never become reversed biased. The APFC control circuitry consisted of a current sensor circuit, decision logic circuit, timing circuit, and a switching circuit. The circuits acting together were responsible for intelligently controlling the capacitor switching.

2.3 Microprocessor – Based SVC's

Two papers, which were both published in 1992, describe a microprocessor-based SVC: [11] discusses an SVC for power systems laboratory experiments, while [12] addresses the open-loop control strategy of SVC's. The contributions of each paper are presented.

In [11], a hardware SVC model was developed for laboratory experiments. The model consisted of a fixed capacitor branch and a controlled inductor branch. The inductor branch consisted of thyristors that received a gating signal from an 8253 programmable interval timer. An Intel 8085 microprocessor on a single-board computer was responsible for determining the correct firing angle α to be provided to the inductor. Two software control strategies were developed to control the inductor's firing angle: proportional plus derivative (PD) control and proportional plus integral plus derivative (PID) control. These control strategies were used to maintain the load voltage at a desired value. Opto-couplers were used to electrically separate the lower voltage of the integrated circuits from the higher voltage of the thyristors. The thyristor gating signals would be supplied to the opto-couplers, which in turn would activate the thyristors.

A second microprocessor-based SVC was developed in [12]. Unlike [11] which focused primarily on hardware design, [12] focused on a specific inductor control strategy that was verified through a developed prototype. An open-loop inductor control strategy was described that could accommodate load changes within the next cycle. The open-loop control is based upon the simple idea that "improving the power factor, no matter its definition, is equivalent to reducing the current drawn from the feeder [12]." The total current is measured and an inductor firing angle α is computed that would minimize the RMS current drawn. Fixed capacitors were chosen to provide the leading VARS to the load. The prototype developed utilized an 8088 10MHZ personal computer with traditional plug in cards such as an ADC (analog to digital conversion) card and a digital I/O card. Current measurements were sampled 128 times each cycle and because integer arithmetic was used, the optimum α could be calculated in 1ms.

2.4 A Fuzzy Logic Control Strategy

In [13] a fuzzy logic control scheme was designed and simulated for a power system SVC. This particular fuzzy logic control strategy was simulated on a fixed capacitor – controlled inductor type SVC, however, the fuzzy control strategy is completely independent of the SVC type used. The goal of this fuzzy controller was to provide maximum damping and stability to the power system. This particular fuzzy controller only used the real power flow as a raw input, and the output is the firing angle α. The power flow signal is conditioned and filtered so that the measures of acceleration, speed deviation, and phase deviation can be extracted. These measurements combine to determine the current "state" of the system. This system state is then used, as a fuzzy input, to a rule set whose defuzzified output will determine whether the inductor susceptance should be capacitive (to increase real power flow at the SVC) or inductive (to decrease real power flow at the SVC).

2.5 Thesis Objective

The objective of this thesis is to develop a prototype *personal* static var compensator (PSVC) that incorporates **both** the switched capacitor banks as well as a controlled inductor. MATLAB/SIMULINK simulations will be performed to verify prototype hardware and embedded software measurement techniques. A low-cost standalone microprocessor will be utilized to control both the capacitor banks and the inductor for load displacement power factor correction, which will ultimately reduce the RMS current drawn. An open-loop, simpler fuzzy logic control strategy will be described to control the firing of the inductor that is responsible for fine-turning the displacement power factor to near unity. The PSVC will be a completely self-contained, robust device capable of being installed on a variety of reactive loads. The PSVC will be evaluated in terms of applicability, efficiency, and cost. The PSVC is designed to target distributive reactive loads, which demand costly reactive power from the AC system. A printed circuit board will be created for the PSVC.

CHAPTER 3 - PSVC PROTOTYPE DESIGN

3.1 PSVC Design Description

Typical types of SVC's include the TCR-FC (Thyristor Controlled Reactor – Fixed Capacitor) and the TCR-TSC (Thyristor Controlled Reactor – Thyristor Switched Capacitor). The TCR-FC type utilizes a thyristor-controlled reactor (inductor) to absorb reactive power while the fixed capacitors generate reactive power. The more flexible TCR-TSC type utilizes capacitor banks that can be switched in and out as needed. Fig. 3.1.1 illustrates a typical SVC that includes a TSC branch as well as a TCR branch that can be controlled continuously to absorb reactive power. Although only one capacitor is shown for the TSC branch, often banks of capacitors are utilized to provide discrete steps of reactive power.

The proposed PSVC prototype will consist of two branches – one TCR branch and one TSC branch. The TSC branch is composed of a bank of thyristor - switched capacitors that provide a dynamic range of leading vars. A firing angle is supplied to the TCR branch so that the susceptance of the reactor can be controlled continuously, thus a finely tuned power factor is achieved. Figure 3.1.2 shows the PSVC prototype configuration. An open-loop fuzzy logic control scheme was chosen for the TCR firing angle calculation, and a PICmicro 16F877 8-bit microcontroller, manufactured by Microchip, was chosen to implement this control strategy.

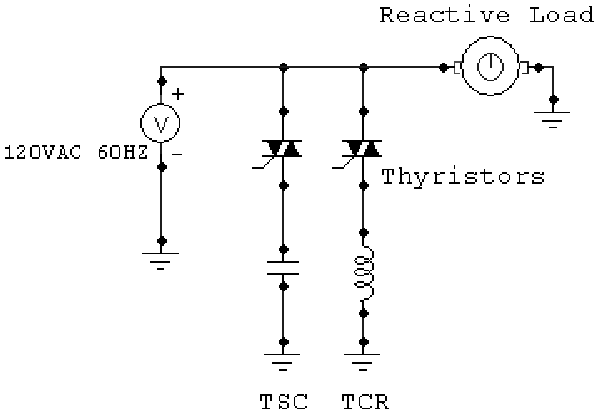


Fig. 3.1.1. Basic SVC Configuration

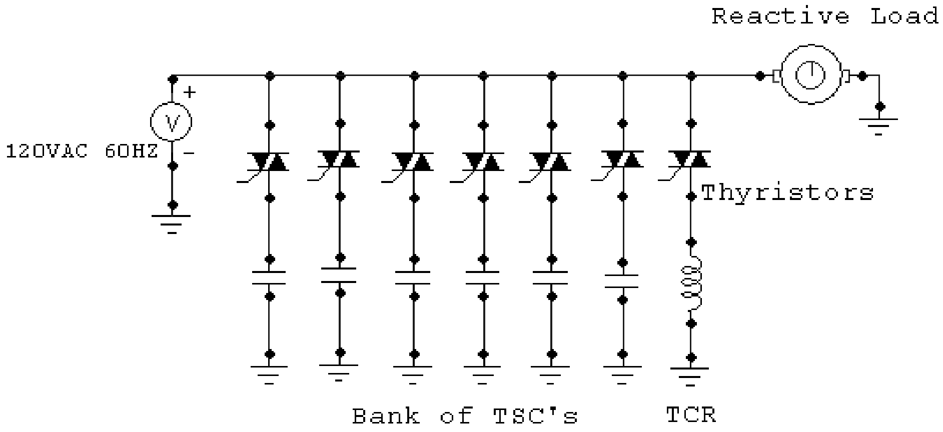


Fig. 3.1.2. Prototype PSVC Configuration

3.2 PSVC MATLAB/SIMULINK Simulations

3.2.1 PSVC Control Verification Through Simulation

SIMULINK is a simulation environment based upon the scientific computing platform, MATLAB. Both MATLAB and SIMULINK are products and registered trademarks of the Mathworks Company. The goal of the simulations was to verify the applicability of the control strategies for the two branches, especially the fuzzy logic control for the TCR. These simulations served as an excellent method for verifying measurement techniques that would later be applied when programming the microprocessor.

Several SIMULINK blocks were created to encapsulate particular functionality needed for the PSVC model. These blocks include: *Capacitor Switching Controller*, *Power Factor Displacement Calculator (PF_D)*, *Load Phase Angle Calculator*, and *TCR Firing Angle Controller*. Each block developed consisted of several hidden sub levels, each with their own defined functionality. These blocks will be described individually, and then a PSVC model will be presented along with its simulation results.

- **Capacitor Switching Controller**

As shown in Fig. 3.2.1, the *Capacitor Switching Controller* block accepts one input: The total power (VA). The total power is further broken down into its two components – real power (watts) and reactive power (vars). Once the reactive power falls below a certain preset value, the *Capacitor Switching Controller* will activate the TSC capacitors with a gating signal outputted from the block's D latch. The "C" block represents a constant preset VARS value that is used to determine when the capacitors should be activated.

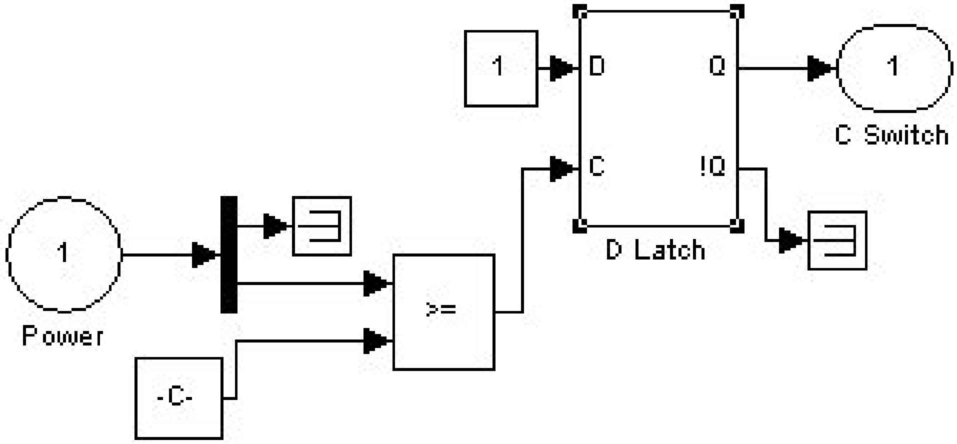


Fig. 3.2.1. Capacitor Switching Controller

- **Power Factor Displacement (PF_D) Calculator**

Fig. 3.2.2 shows the *Power Factor Displacement (PF_D) Calculator* block, which accepts the total power (VA) and outputs the displacement power factor for the particular load. As described previously, the displacement power factor (PF_D) is the ratio of real power to total power, when only considering the 60HZ fundamental frequency. PF_D, referenced in [14], is defined by Equation 3.1. This block is used to verify the output from the *Load Phase Angle Calculator* block, which is described next.

$$PF_D = (kW / kVA) = \cos\theta \qquad (3.1)$$

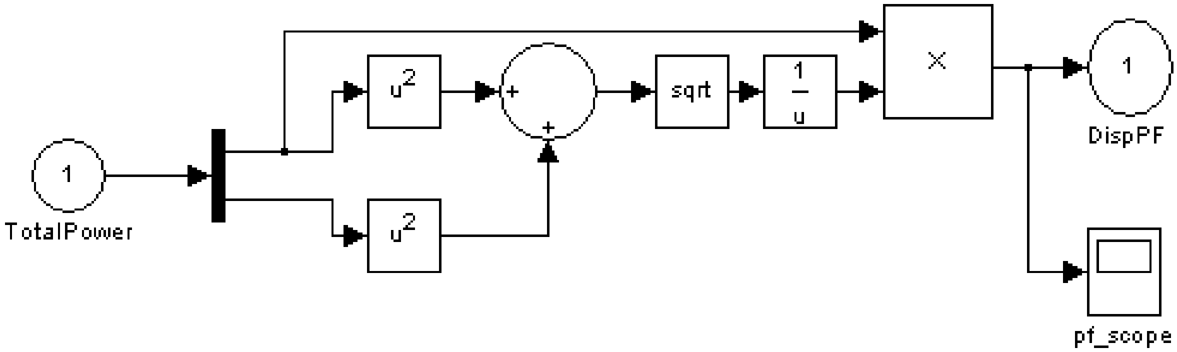


Fig. 3.2.2. Power Factor Displacement Calculator

- **Load Phase Angle Calculator**

The *Load Phase Angle Calculator* block, Fig. 3.2.3, was designed to calculate the load phase angle using nearly the same method as the microprocessor. These simulations served as good indicators for measurement techniques that would later be used when programming the microprocessor.

This block accepts the voltage and current waveforms as inputs and calculates the phase angle (difference) of the two waveforms. The voltage and current waveforms are polled for their zero crossings independently, and their status is saved as control flags. Once the zero crossings occur, their individual times are stored. To determine whether the current is leading or lagging, the control flag variable is set depending on which zero crossing is first detected. After both zero crossings are detected, the times are subtracted from one another and converted into a phase angle θ. The time difference (ΔT) can be converted into the phase angle θ by using Equation 3.2. The PF_D can be found by taking the cosine of θ. The *PF_D Calculator* block was used to verify the calculated θ.

$$\theta = 360[\Delta T/(1/60)] \tag{3.2}$$

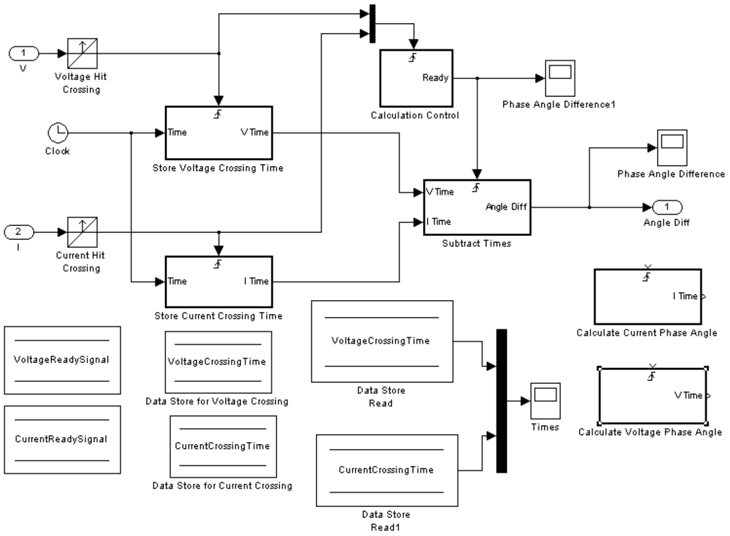


Fig. 3.2.3. Load Phase Angle Calculator

- **TCR Firing Angle Controller**

The TCR firing angle controller, shown in Fig. 3.2.4, accepts the current waveform (I), total power (VA), and the phase angle difference (θ) as inputs. Note that the current waveform and power inputs are used only internally for inspection, and is not actively needed by the controller. However, the phase angle difference input is needed as the primary input to the fuzzy logic controller. The output from the fuzzy logic controller is the TCR firing angle in degrees, which is translated into timed firing pulses and sent to the TCR thyristors. The fuzzy logic control scheme was developed using the MATLAB *Fuzzy Logic Toolbox* and the membership functions are shown in Fig. 3.2.5 and 3.2.6. The input membership functions from left to right are: *LargestNeg*, *LargeNeg*, *MediumNeg*, *SmallNeg*, *SmallPos*, *MediumPos*, *LargePos*, and *LargestPos*. The output membership functions from left to right are: *Smallest*, *Small*, *Medium*, *Large*, *Largest*, *Huge*, and *Off*.

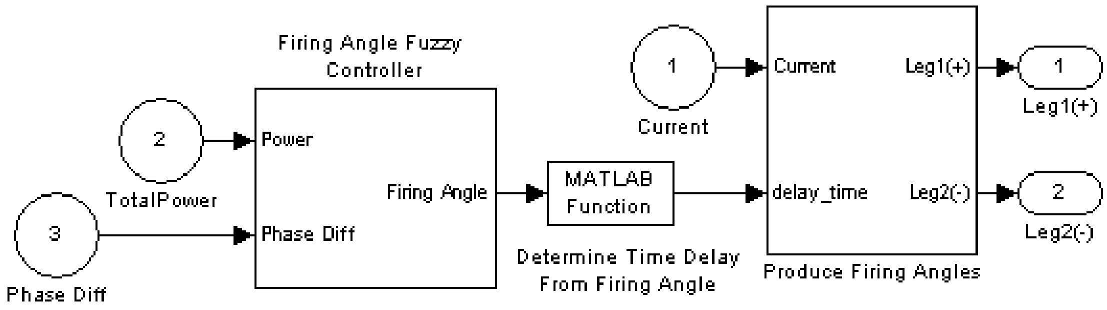


Fig. 3.2.4. TCR Firing Angle Controller

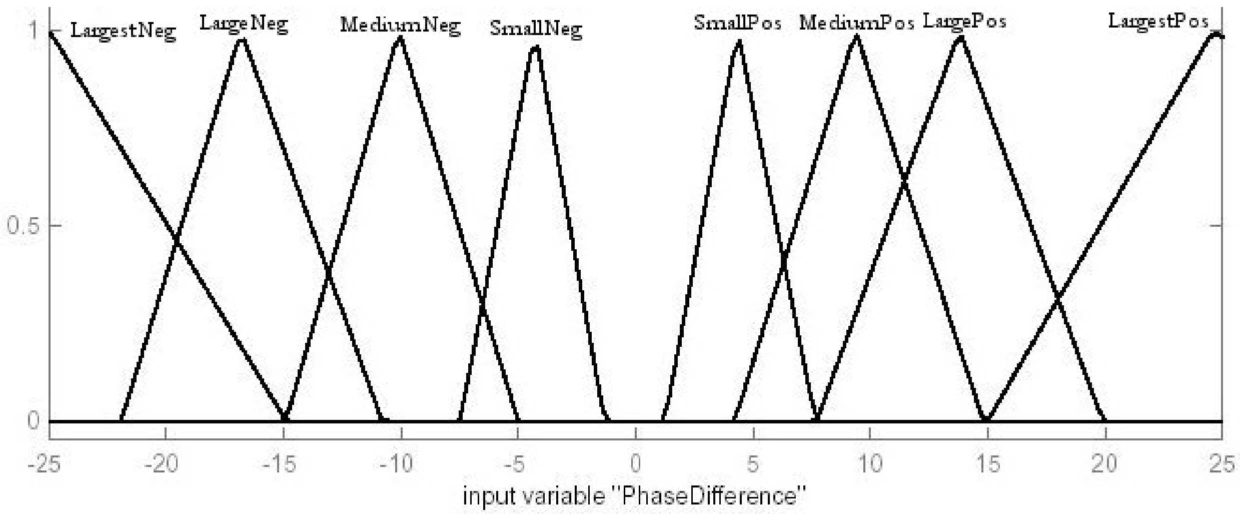


Fig. 3.2.5. Fuzzy Input Membership Functions for Phase Angle Difference

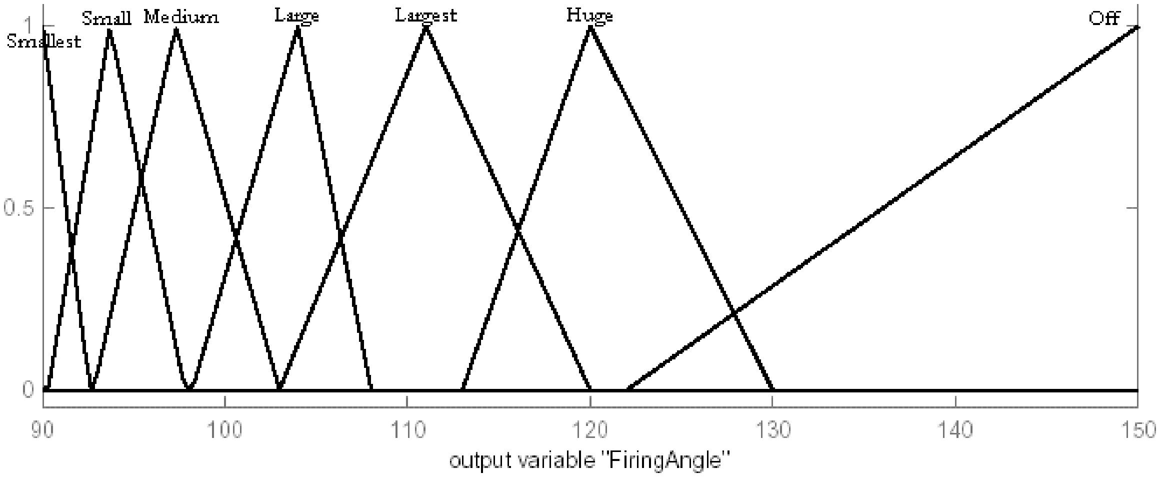


Fig. 3.2.6. Fuzzy Output Membership Functions for TCR Firing Angle

As Figures 3.2.5 and 3.2.6 illustrate, the controller consists of a one input – one output fuzzy logic controller. The *Phase Angle Difference* (θ) input spans from –25° to +25°. No membership functions are defined around zero because this area is defined as a "no-action" area. The *Firing Angle* output membership functions span from 90° (TCR on) to 150° (TCR effectively off). The fuzzy rules map one-to-one between the input and the output, and for the simulations, a centroid defuzzification was performed. Fig. 3.2.7 shows the "surface" covered by the fuzzy logic controller. The membership functions must be tuned so that the optimum firing angle for the particular load is centered in the output membership functions. This particular controller was configured so that a zero degree phase angle difference (θ = 0°) would settle to a 120° firing angle (α = 120°).

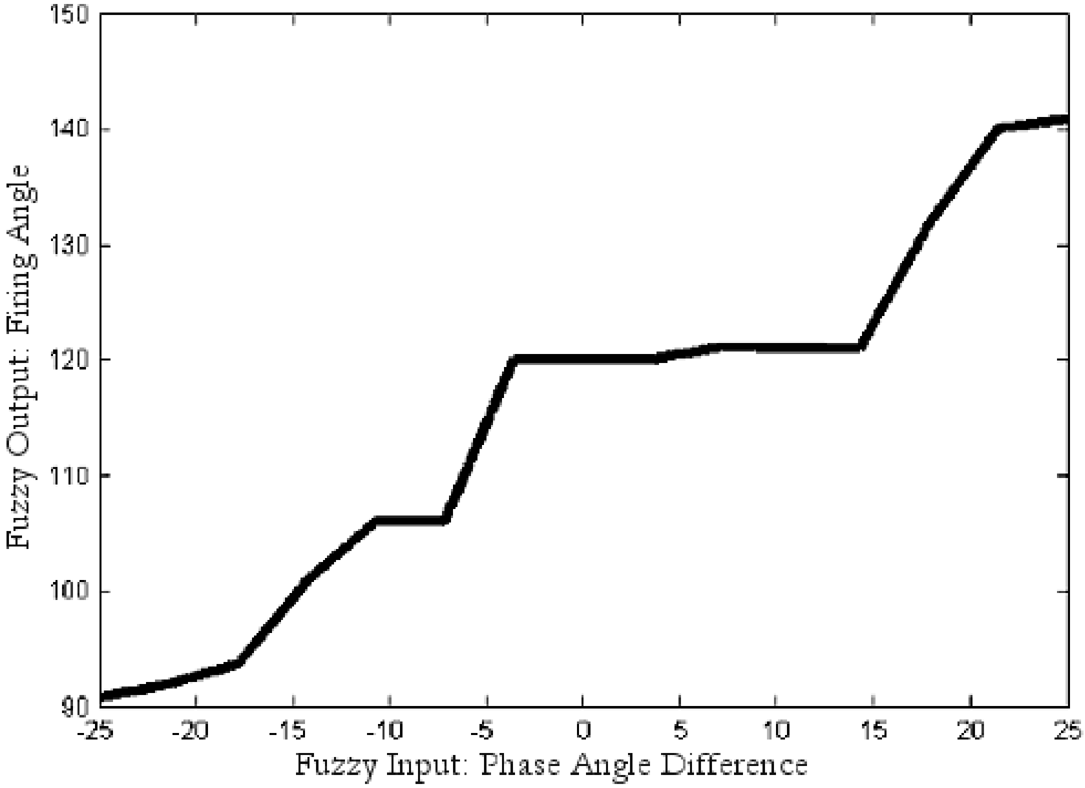


Fig. 3.2.7. Fuzzy Surface Area: Firing Angle vs. Phase Angle Difference

- **PSVC Simulation of TSC and TCR Branches**

Several SIMULINK simulations were created to verify the feasibility of the PSVC design and the TCR fuzzy controller. For example, a 5kW/8kVar reactive single-phase load was modeled with a slight load change (switch closing) occurring at time t = 1s. The reactive load is fed by a 1kV/60 HZ source. The TSC branch and a TCR branch were modeled with their respective controllers. The TSC branch consists of a single switched capacitor while the TCR branch consists of a reactor that is fed a firing angle determined by the fuzzy controller. The fuzzy controller accepts the phase angle difference of the load as an input and outputs the optimum firing angle. Back-to-back SCR's with snubber circuits were chosen to closely match the actual thyristors used in the prototype. This particular SIMULINK model is shown in Fig. 3.2.8. For readability purposes, several smaller SIMULINK blocks have been removed from the figure.

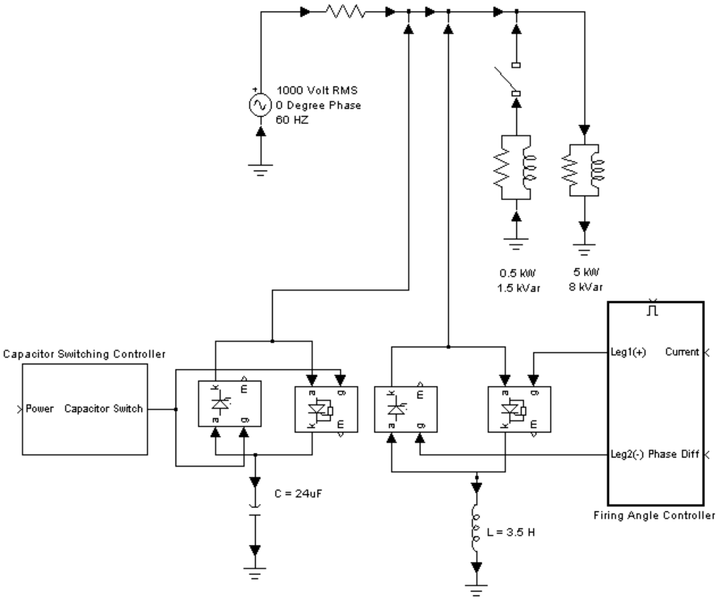


Fig. 3.2.8. Simulink Model of PSVC

Fig. 3.2.9 shows the displacement power factor (PF_D) of the load without the branches compensating. At t = 1s the switch closes and the PF_D drops slightly. Fig. 3.2.10 illustrates the improved PF_D because of the compensating branches. Fig. 3.2.11 illustrates the changing current through the TCR branch at t = 1s, which is a result of the updated firing angle.

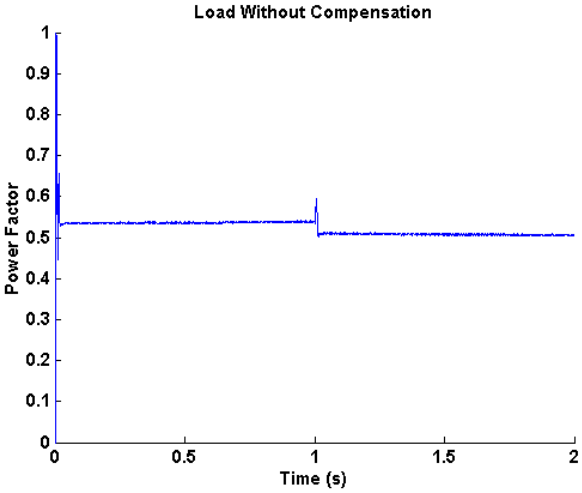


Fig. 3.2.9. Simulated reactive load without PSVC compensation

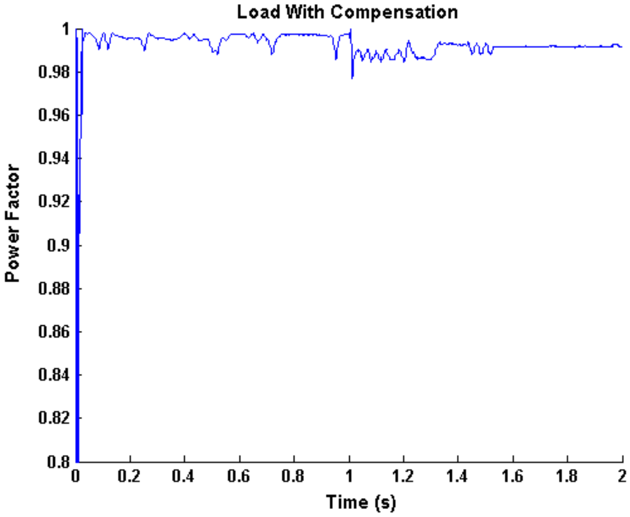


Fig. 3.2.10. Simulated Reactive Load with PSVC Compensation

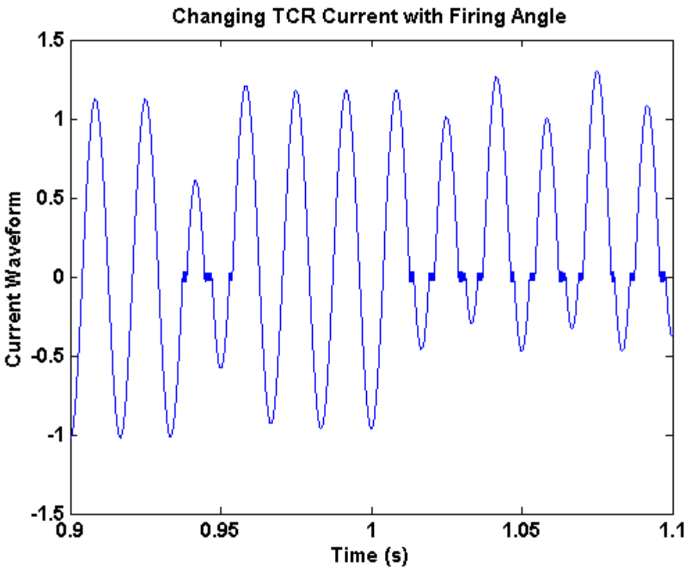


Fig. 3.2.11. Changing TCR Current at Load Change

The values for the capacitor and inductor were selected by considering the reactive power demanded by the load. The TSC branch is calculated to provide a slight overcompensation (approximately 10% - 20%) of reactive power, and the TCR size is chosen to offset that overcompensation. The coordination of the two branches provide for a finely tuned displacement power factor. Equation 3.3 shows how the capacitor value can be found. Equation 3.4 illustrates how the inductor size can be found. For the equations, a 20% TSC overcompensation is used. Designers will vary this value considerably depending on load characteristics and available capacitor bank sizes.

$$\begin{aligned} Q_{LOAD} + (20\% * Q_{LOAD}) &= V^2{}_{RMS} / X_C \\ X_C &= (2\pi f C)^{-1} \end{aligned} \qquad (3.3)$$

$$\begin{aligned} (20\% * Q_{LOAD}) &= V^2{}_{RMS} / X_L \\ X_L &= (2\pi f L) \end{aligned} \qquad (3.4)$$

3.2.2 Simulation of PSVC Targeted Application – AC Motor

Before hardware design began, the targeted PSVC application was simulated to verify the static components' sizes and effectiveness. Specific mathematical calculations will be presented in the upcoming sections where hardware design is the main focus. The purpose of this section is to introduce the PSVC test application and illustrate the effectiveness of the PSVC through simulations.

The test application is a small 55 Watt AC motor that demands 78 vars of reactive power, and has a displacement power factor (PF_D) of 0.58. Table 3.2.1 summarizes the characteristics of the AC motor with 120 V_{AC} RMS source.

Motor Characteristic	**Value**
Total Power	95 VA
Real Power	55 Watts
Reactive Power	78 Vars
Current Draw	0.8 A
Power Factor (PF_D)	0.58

Table 3.2.1. AC Motor Characteristics

SIMULINK simulations were developed of the AC motor to verify the capacitor and inductor sizes needed to compensate the reactive load. In SIMULINK, the motor was modeled as an inductive load with the characteristics described in Table 3.2.1. The PSVC branches were modeled as capacitive and inductive elements. The TSC branch consisted of a single, fixed capacitor that provided a slight overcompensation of reactive power to the load. An 18uF capacitor was chosen to provide 98 vars of reactive power with an overcompensation of 21 vars. An inductor in series with two back-to-back thyristors modeled the TCR branch. Fuzzy control of the TCR branch was not utilized because the main purpose of these simulations was to verify the sizes of the static components. Instead of fuzzy control, pulse trains were utilized to simulate the TCR firing angle. A 125 mH inductor, present in the laboratory, was chosen to provide an excessive 300 vars of reactive power when fully turned on. Fig. 3.2.12 shows the model used to simulate the AC motor application.

Table 3.2.2 shows the effectiveness of the individual PSVC branches. First, the TCR branch is turned off and the TSC branch overcompensates providing a leading displacement power factor 0.95, shown in Fig. 3.2.13. Then, the TCR branch is activated and an optimum firing angle of 150° is found manually. It is obvious that the TCR inductor is overly large for the motor and TSC branch because it is nearly turned off with its 150° firing angle, as shown in Fig. 3.2.14. However, this indicates that the inductor could be used for a larger range of power factor fine-tuning if needed.

	Power Factor (PF_D)	**Firing Angle**
No Compensation	0.58	-
TSC Compensation	0.95 Leading	-
TSC + TCR Compensation	0.99 Lagging	150

Table 3.2.2. Simulated Effectiveness of PSVC

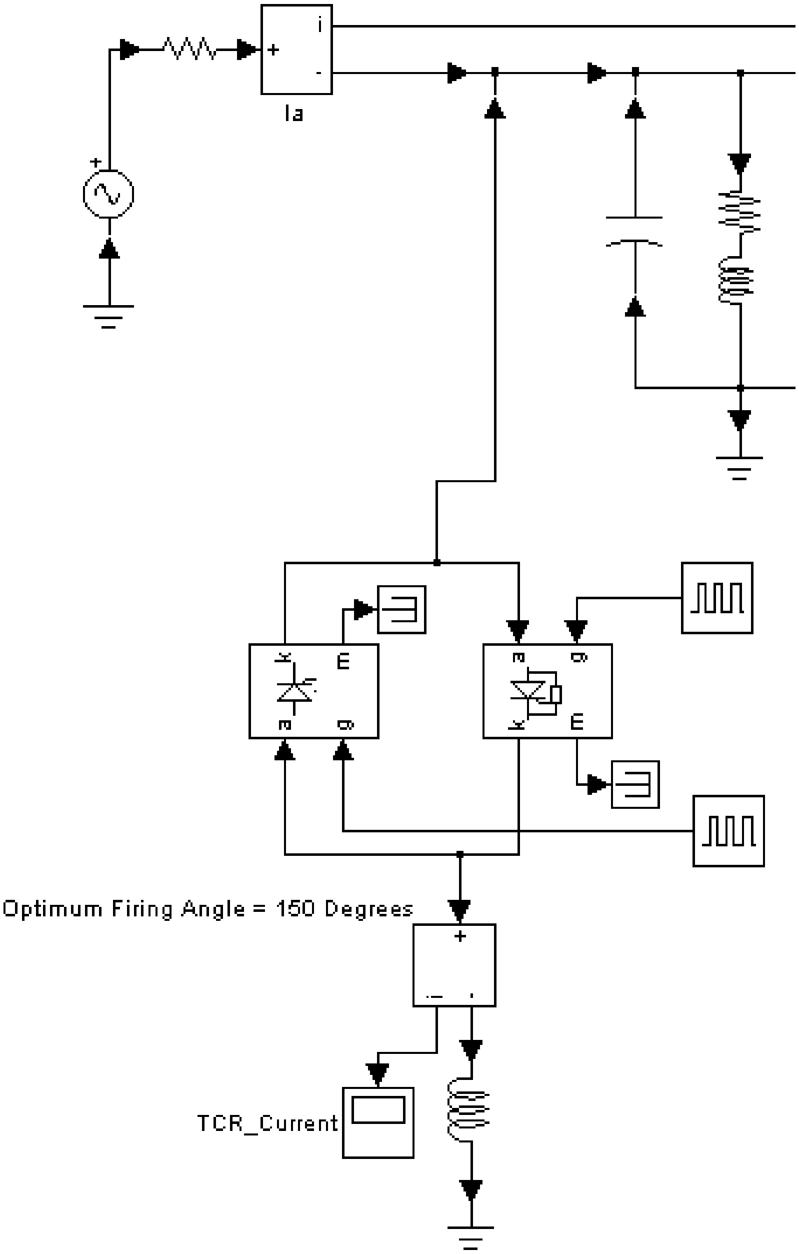


Fig. 3.2.12. Model of AC Motor with TSC and TCR Branches

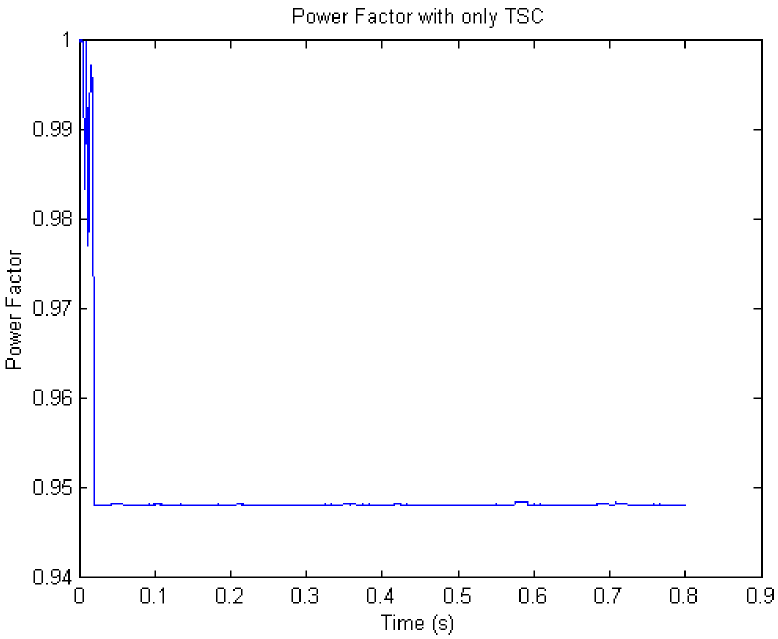


Fig. 3.2.13. Simulation of Only TSC Compensation – Load Displacement PF

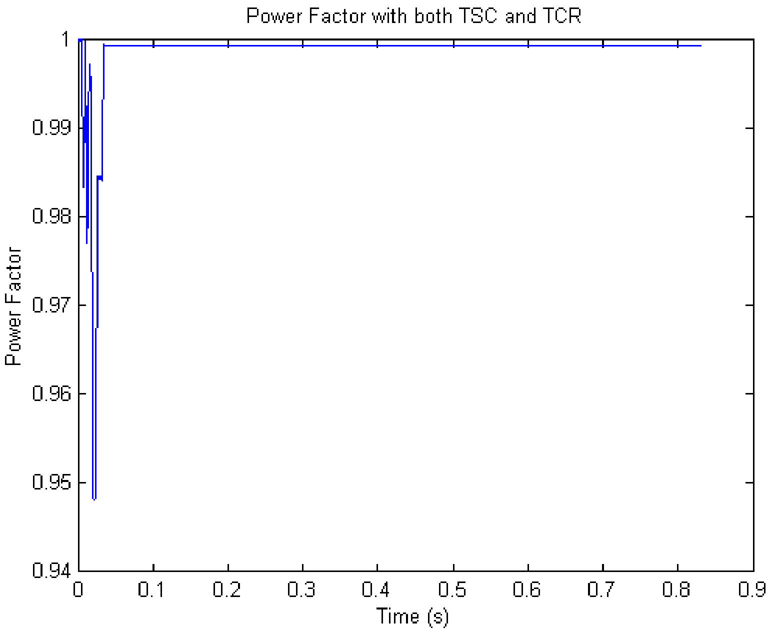


Fig. 3.2.14. Simulation of Both TSC and TCR Branches – Load Displacement PF

3.3 PSVC Circuit and Hardware Design

The PSVC circuitry can be divided into three logical divisions: 1) Waveform measurement for the microprocessor, 2) The microprocessor and PSVC branch (TSC and TCR) interfacing, and 3) The high power TSC and TCR PSVC branches. Design steps, circuit descriptions, and circuit drawings will be presented for each of the divisions.

3.3.1 Waveform Measurement for the Microprocessor

The microprocessor chosen to control the PSVC is the 8-bit PICMicro® ("PIC") 16F877 manufactured by Microchip. Also note that the PIC16F874 can also be chosen. This microprocessor was chosen for its low cost ($8.00), small size, and versatility. The PIC is a RISC processor having only 35 single word instructions [15]. 20MHZ, the maximum clock speed available for the PIC, was chosen to ensure ample time for mathematical calculations. The PIC contains 368 bytes of RAM, interrupt capability, three timers, and digital input and output [15]. Note that the smaller 24 -pin PIC16F874 can be substituted for the PIC16F877. This section will present the microprocessor hardware and any supporting circuitry needed to perform the important function of load displacement power factor (PF_D) calculation. The PF_D is the control variable used for both the TSC and TCR branches. PIC software design is discussed in section 3.4.

To measure the PF_D, the PIC must be provided with the waveforms of both the voltage and the current. Because of the higher voltage used, both the voltage and current waveforms must be conditioned so that they are acceptable as digital inputs to the PIC. The load voltage waveform is stepped down from $120V_{AC}$ RMS to $5V_{AC}$ RMS using a voltage divider circuit with a ratio of 27. The load current waveform was extracted by using a current transformer with a 180 Ω resistor. The current transformer wire windings and the resistor were chosen to yield a waveform of approximately $5V_{AC}$. Fig. 3.3.1 shows the extraction of the voltage and current waveforms.

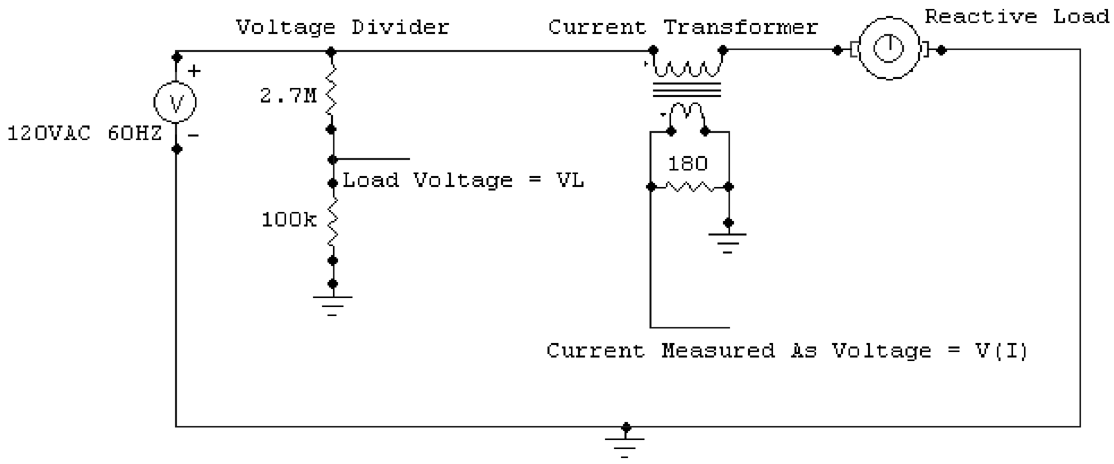


Fig. 3.3.1. Voltage and Current Waveform Extraction

The current waveform extracted by the current transformer must be filtered so that any frequencies other than the 60HZ fundamental can be eliminated for accurate displacement power factor measurement (PF_D). The filter chosen is the UAF42 universal active filter manufactured by Burr-Brown (Texas Instruments) [16]. The UAF42 can be configured to provide either low-pass, high-pass, or bandpass filters. The UAF42 contains two integrators and one inverting amplifier. A digital filter such as the MF8 switched capacitor filter was first considered, however the digital filter requires an external clock source and often suffers from switching noise.

Burr-Brown provides a software tool that aids in the UAF42 filter design [17]. This software tool allows the user to specify the desired filter and when finished, will provide the user with the appropriate external resistor values needed to implement the filter. For the PSVC current waveform filter design, a 2nd order bandpass filter with a center frequency of 60 HZ (bandwidth of 5 HZ) was designed. If a 4th order filter was desired then two UAF42 IC's would be required in a back-to-back configuration. A 2nd order filter was chosen for evaluation, design simplification, and cost reduction. When the UAF is configured as a 2nd order filter it assumes the characteristics of a tuned L - C - R circuit. The following external resistors were used to implement the bandpass filter: two 2.55M Ω and one 2.74M Ω. Fig. 3.3.2 shows the placement of the filter in the circuit.

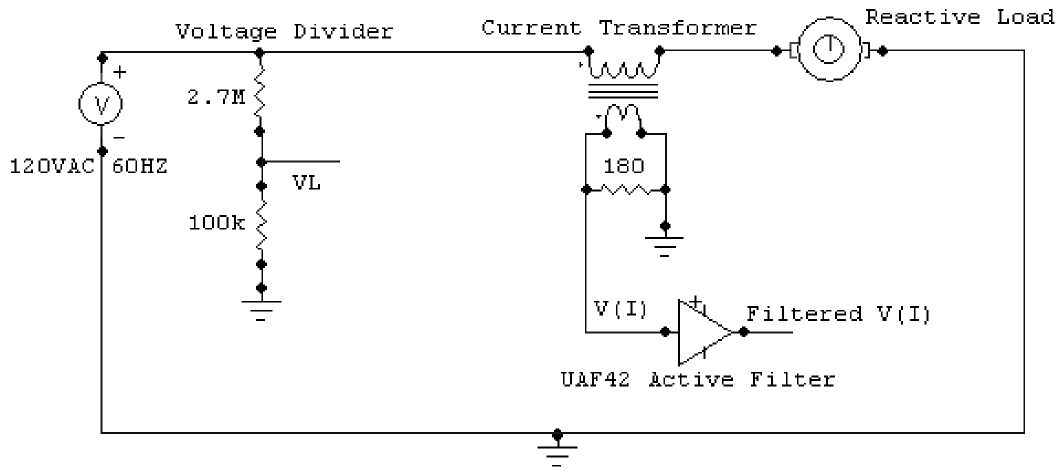


Fig. 3.3.2. Filtering of the Current Waveform

After the waveforms are extracted and the current filtered, the waveforms are passed through individual diode voltage clamps that are designed to ensure that the conditioned waveforms will not exceed a particular voltage. The diode voltage clamps are only present to ensure that the waveforms do not ever exceed a particular voltage. Although the clamps are not entirely necessary for the functionality of the circuit, they are recommended safety devices to ensure that appropriate voltage levels are maintained. Fig. 3.3.3 shows the voltage clamps. For the PSVC, ± 5 volts were chosen as the voltage limits for the clamps.

Because the waveforms are converted into CMOS digital inputs by the PIC input ports, the waveforms must be converted from sinusoidal to square waves external to the PIC to ensure that the two waves are comparable at the same time. In other words, the sinusoidal waveforms are not acceptable as inputs to the PIC because of the CMOS input logic level ranges. As illustrated in Fig. 3.3.4, the HIGH logic level is approximately 70% of the supply voltage (V_{DD} = 5 Volts), while the LOW logic level is approximately 30% of the supply voltage [18].

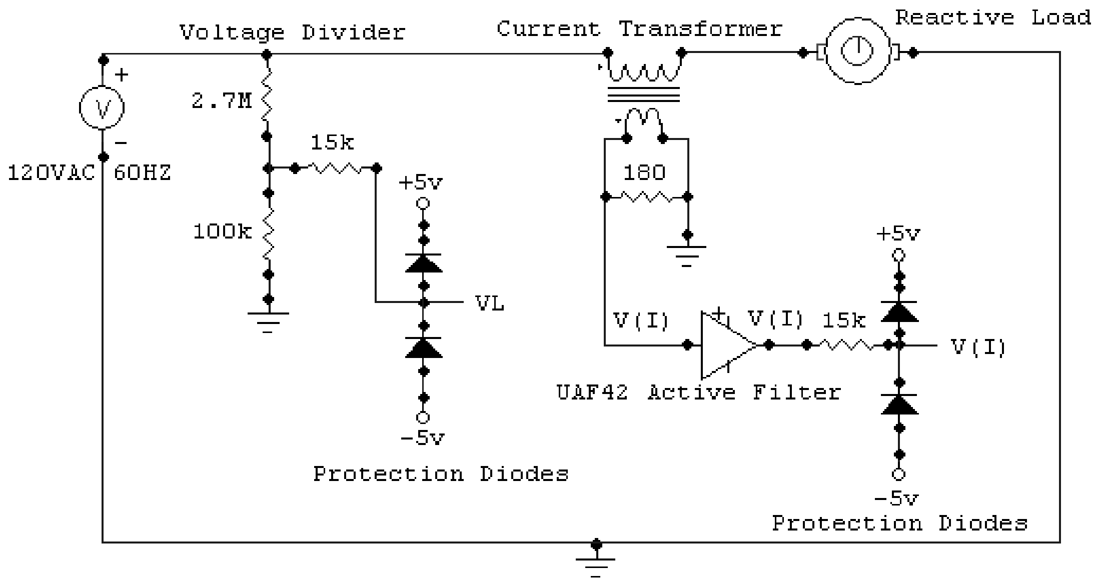


Fig. 3.3.3. Diode Voltage Clamps for Safety

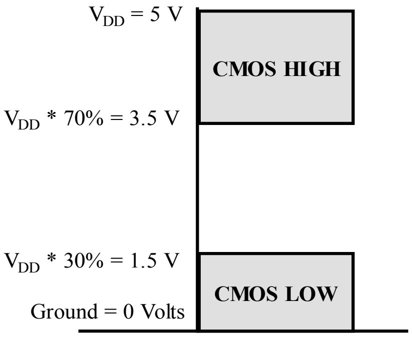


Fig. 3.3.4. CMOS Input Logic Levels

If the input waveforms are sinusoidal then the corresponding CMOS digital signals will not be comparable at the same time by the PIC. To remedy this situation a voltage comparator is used prior to the waveforms being applied as inputs to the PIC. Both the voltage and the current waveforms are applied to individual voltage comparators, located on a single LM339 voltage comparator, that compares the input signal to 0 volts. If the input signal is less than 0 volts, then the signal will be chopped at –5 volts. Likewise, if the input signal is greater than 0 volts, then the signal will be chopped at +5 volts. Fig. 3.3.5 shows the schematic for the voltage comparators while figures 3.3.6 and 3.3.7 show an example waveform before and after being conditioned by the voltage comparators.

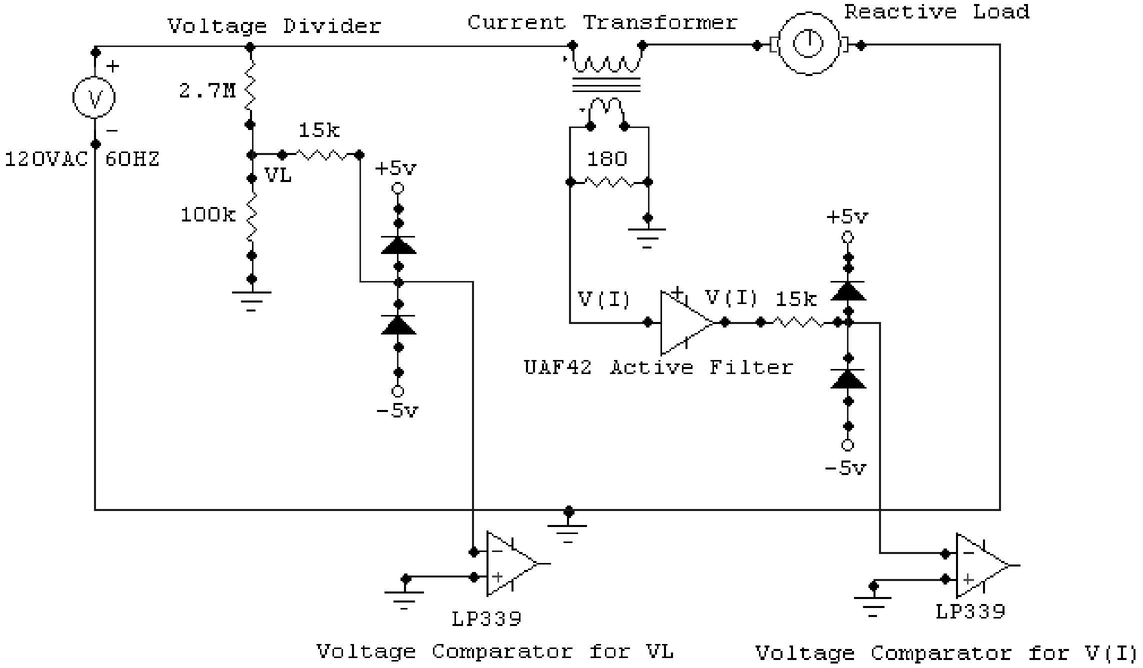


Fig. 3.3.5. Voltage Comparator Utilization for Waveform Conditioning

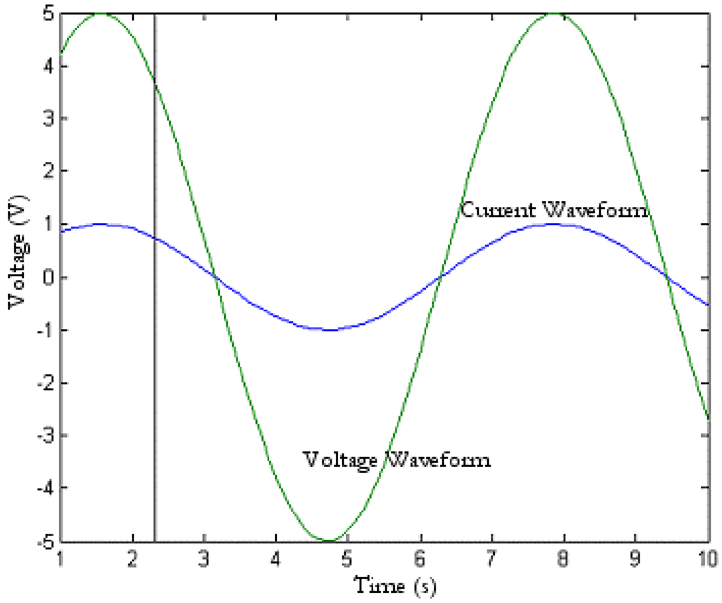


Fig. 3.3.6. Example Waveform Before Voltage Comparators

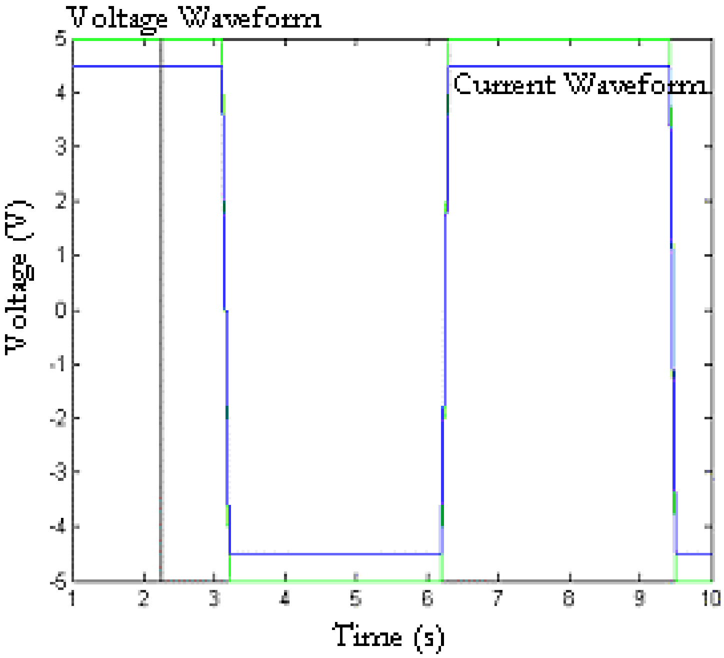


Fig. 3.3.7. Example Waveform After Voltage Comparators

The ± 5 volt square wave is now applied to two individual digital input ports on the PIC. These CMOS ports consist of two protection diodes and a pair of MOSFETS. The protection diodes are similar to those used externally to the PIC, however these clip the input to either 0 volts or $V_{DD}+V_{DIODE}$ volts. The MOSFETs, serving as a zero-crossing detector, will cause the PIC to read either a HIGH or LOW on the port depending on the polarity of the clipped input [19]. Fig. 3.3.8 shows a typical PIC input port. As described in 3.4.1.1, strategic polling of the input pins is how the PIC detects zero-crossings as well as whether the current is leading or lagging. Fig. 3.3.9 on the following page shows the complete electrical schematic for the load waveform measurements needed for the microprocessor.

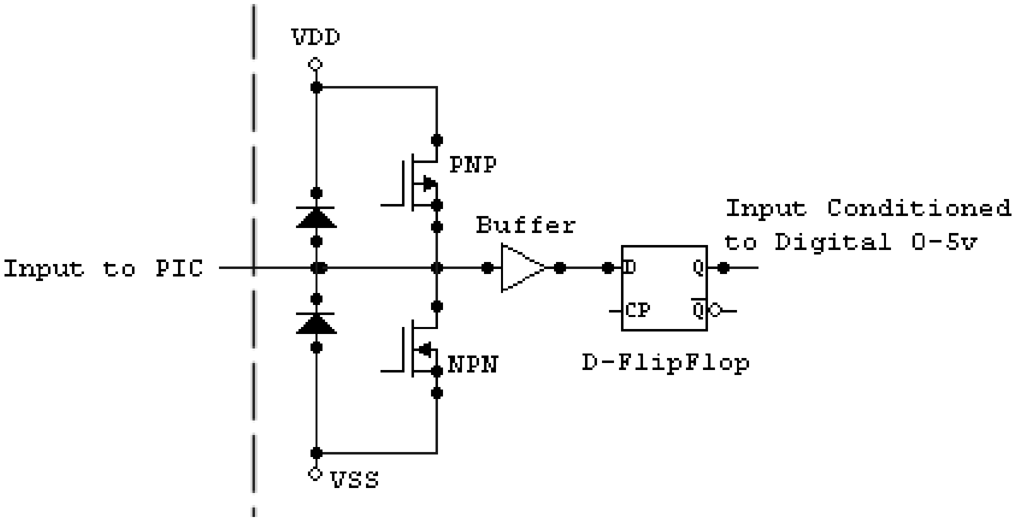


Fig 3.3.8. Internal Components of the PIC Input Port

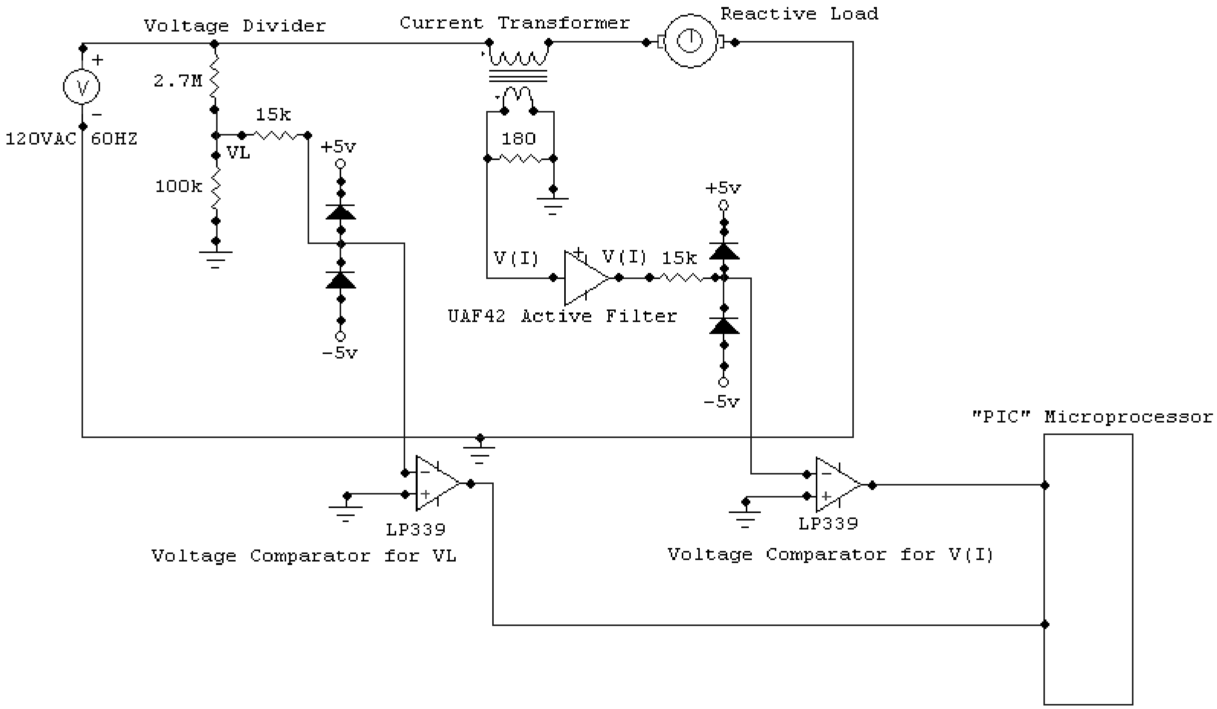


Fig. 3.3.9. Complete Load Waveform Measurement Schematic

3.3.1 Microprocessor and PSVC Branch (TSC and TCR) Interfacing

The PIC controls the TSC and TCR branches by outputting a control signal to a MOC3051 triac driver. A triac driver is needed to control each individual branch used. For example, if a bank of 8 capacitors (TSC) are used and 1 inductor (TCR), then 9 triac drivers will be needed. Likewise, each branch must be given a dedicated output port on the PIC.

The MC3051 triac driver is a random-phase optoisolator. It consists of an infrared LED optically coupled to a non-zero-crossing bilateral AC switch [20]. The purpose of the MOC3051 is to isolate the higher voltage triacs from the lower voltage control. This electrical isolation helps to ensure that the microprocessor remains isolated from the high voltage. Since the MOC3051 is a current-activated device, the proper resistor to be placed between the MOC3051 and the microprocessor must be chosen carefully. From Fig. 11 referenced in [21], the equation for the resistance needed to activate the triac driver is shown in Eq. 3.5, where V_{CC}=5Volts, V_{FLED}=forward voltage of the diode (LED), and I_{FT} = LED forward current.

.
$$R_{LED} = (V_{CC} - V_{FLED}) / I_{FT} \tag{3.5}$$

The pulse-width of the signal applied to the PIC is also a factor when choosing the appropriate resistor. From Fig. 4 referenced in [22], a relatively long 20us pulse-width signal was chosen so only 5mA would be sufficient to activate the diode (LED). With a 20us pulse-width signal I_{FT} = 5mA, and from Fig. 1 in [23], V_{FLED} = 1.1 volts. Using these values, Eq. 3.5 yields a 780 Ω resistor. A standard resistor value of 750 Ω was chosen. Fig. 3.3.10 shows the typical connections between the PIC and the MOC3051 triac drivers. In this example, two output PIC ports were utilized to drive the two triac drivers.

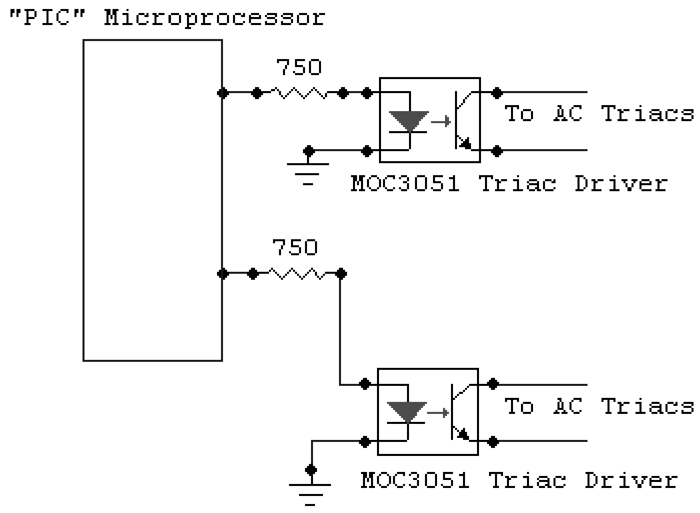


Fig. 3.3.10. Typical Interface Between the PIC and MOC3051 Triac Drivers

3.3.2 High Power TSC and TCR PSVC Branches

The AC triac is the device that performs the high powered switching of the TSC and TCR branches. The AC triac is activated by a gate signal that is applied by the MOC3051 triac driver. The AC triac consists of two main terminals (MT1 and MT2) as well as a gate (G) for control signals. The particular triac chosen for the PSVC branches is the Alternistor Triac manufactured by Teccor Electronics Inc. This particular triac is specifically designed to be able to switch highly inductive loads without the need for costly external snubber circuits. It is able to accomplish this because its SCR's (silicon controlled rectifiers) are electrically separated providing better than average turn off capabilities [24]. The particular alternistor triac chosen for the PSVC is capable of handling 15 A at 600 volts. A typical alternistor triac configuration is shown in Fig. 3.3.11.

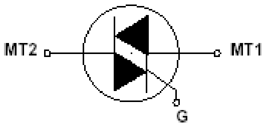


Fig. 3.3.11. Typical Configuration of an Alternistor Triac

Applying an in-phase signal to its gate turns on the triac. No turn-off control is needed because the alternistor triac will stop conducting at the next current zero crossing, also known as "self-commutation". For the TSC branch, a constant gate signal will effectively keep the capacitor banks switched-in. For the TCR branch, an intelligent triggering mechanism is needed to turn on the TCR triac at exactly the correct time to implement a particular firing angle delay. The microprocessor's software is responsible for intelligently implementing the TCR firing angle. Fig. 3.3.12 shows the interface between the MOC3051 triac driver and the alternistor triac. Fig. 3.3.13 is PSVC block diagram illustrating the interactions between the reactive load waveform measurements, microprocessor, and TSC/TCR branches. For simplicity, only a single TSC and TCR branch are shown – although the actual PSVC prototype utilized six TSC branches and one TCR branch. An entire electrical schematic of the PSVC can be found in the appendix A.

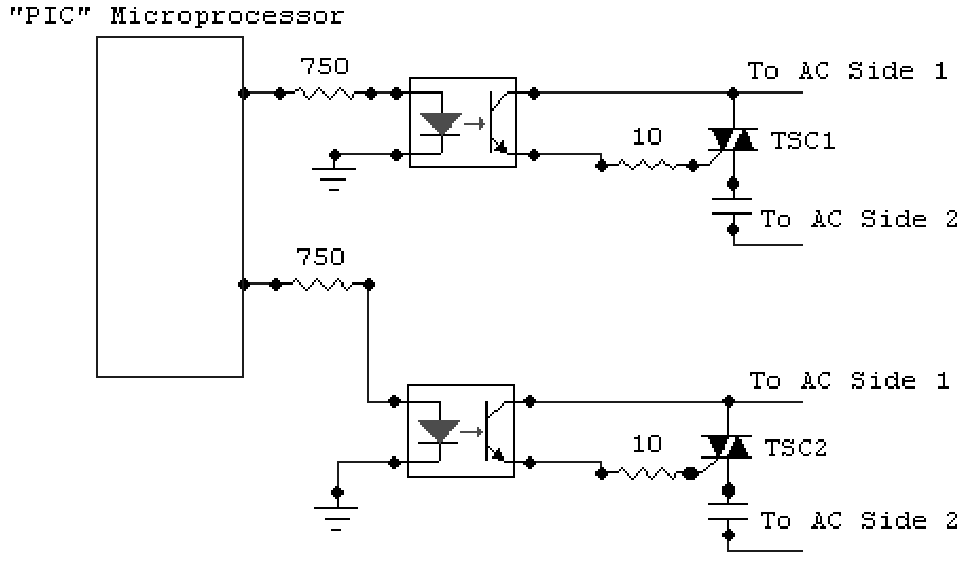


Fig. 3.3.12. Interface Between the MOC3051 Triac Driver and the AC Alternistor Triac

3.4 Microprocessor Software Design and Implementation

The PIC microprocessor is responsible for the following key tasks: Displacement load power factor (PF_D) calculation through zero crossing detection, TSC switching control, fuzzy control of the TCR branch, and LCD output. Each task is detailed individually. A final section is also devoted to software implementation that describes the programming language, compiler, software tools, and any other necessary programming hardware that was needed.

3.4.1 Calculation of Displacement Power Factor (PF_D) of Reactive Load

Recall from Eq. 3.1 that the displacement power factor (PF_D) is the ratio of real power to apparent power, only considering the fundamental frequency. The PF_D, which is indirectly the control input for both the TSC and TCR branches, is calculated by having the microprocessor determine the time difference between the reactive load's voltage and current waveforms. The following list summarizes the functional and mathematical steps taken by the microprocessor to calculate the PF_D. The list will be expanded into subsections to provide specific microprocessor implementation details.

1. Determine Whether the Reactive Load is Leading or Lagging

2. Calculate the Time Difference Between Voltage and Current Waveforms

3. Convert the Time Difference to a Phase Angle Difference

4. Convert Phase Angle Difference to PF_D

3.4.1.1 Determining Whether the Reactive Load is Leading or Lagging

To determine whether the reactive load is leading or lagging, the PIC will first wait for the voltage waveform to cross zero. Once it crosses zero the PIC will record the state of the voltage waveform and then immediately record the state of the current waveform. By comparing the waveform states, the PIC can determine whether the current is leading or lagging the voltage. To detect a zero crossing, the PIC continuously polls a waveform watching for its value to flip. For example, if the PIC read a HIGH for the voltage, then a loop continuously polling the voltage would detect when the voltage input flipped to a LOW. When that occurs, the PIC has detected a zero crossing.

If the voltage waveform was recorded as HIGH after the zero crossing, and the current waveform was recorded as LOW, then it is determined that the current is lagging the voltage. Likewise, if the voltage was recorded as HIGH, and the current was already HIGH, then the current would be considered leading the voltage. Because a voltage zero crossing is needed for synchronization, the PIC needs half a cycle to determine whether the current waveform is leading or lagging. Table 3.4.1 shows all of the possible combinations of the current waveform leading/lagging logic. Note that the voltage waveform is the first waveform to be polled after the zero-crossing is detected.

	Current HIGH	**Current LOW**
Voltage HIGH	LEADING	LAGGING
Voltage LOW	LAGGING	LEADING

Table 3.4.1. Logic to Determine Whether Current is Leading/Lagging Voltage

3.4.1.2 Calculating the Time Difference (ΔT) Between Voltage and Current Waveforms

Once it has been determined whether the current is leading or lagging the voltage, the PIC will calculate the time difference separating the two waveforms. If the current is lagging the voltage, the PIC will wait for the voltage to cross zero at the next half cycle and then start its 16-bit timer. The PIC will then wait for the current to cross zero. Once the current crosses zero, the timer is stopped and its value is stored in a variable.

Likewise, if the current is leading the voltage, the PIC will wait for the current to cross zero at the next half cycle and then start its 16-bit timer. The PIC will then wait for the voltage to cross zero, and when this occurs, the PIC will stop its timer and record the value to a variable. Calculating this time difference takes one half cycle because the waveform zero crossings are needed to synchronize the timer.

3.4.1.3 Converting the Time Difference (ΔT) to a Phase Angle Difference (θ)

The PIC's 16-bit timer was configured via software to tick once every 1.6us, which provides for a maximum time count of 0.1 sec before the timer will roll over. 1.6us was chosen because it is a multiple of the clock frequency, 20MHZ. The 0.1s rollover provides ample time for the cycle-by-cycle calculations that are performed every 8.33ms and 16.66ms. The 1.6us ticks also provide a high measurement resolution of the time difference. For only a 1° (46us) phase difference, the PIC will record 28 clock ticks.

Using Eq. 3.6 the timer "ticks" can be converted into a time difference (ΔT). After ΔT is found, the phase angle difference (θ) of the load can be found by using Eq. 3.2 introduced earlier. If the load is lagging then the phase angle difference is given a positive sign. If the load is leading then the phase angle is assigned a negative sign.

$$\Delta T = Ticks * (20MHZ / 32)^{-1} \tag{3.6}$$

3.4.1.4 Converting the Phase Angle Difference (θ) to PF$_D$

The PF$_D$ can be calculated easily by taking the COSINE of θ. Performing this calculation is optional for the PIC because θ is used as the control variable for the TSC and TCR branches. If a slower clock frequency was chosen this calculation could be omitted to save processing time.

3.4.2 TSC Switching Control

As first discussed in Section 2.2, the optimum capacitor configuration for the individual TSC branches can be found by sizing the capacitors in binary steps. For the PSVC prototype, the binary steps of 1uF, 2uF, 4uF, 8uF, 16uF, and 32uF were chosen to provide a total of 64uF in discrete steps of 1uF with only using 6 capacitors. This flexibility allows for a tightly configured displacement power factor compensation that can be easily programmed to be either leading or lagging. Fig. 3.4.1 illustrates the TSC branches configured in binary steps.

The binary configuration is important to the software design because the PIC must be able to intelligently switch in and out the appropriate TSC branch at the appropriate time. Every 5 seconds the PIC is programmed to examine the current phase angle difference (θ) and switch in or out the next capacitor size in binary order. Generally, if θ positive (lagging PF_D) the PIC will switch in TSC branches in binary order (increments of 1uF) until θ is no greater than $-15°$ (leading PF_D). Likewise, if θ is negative (leading PF_D) the PIC will switch out TSC branches until θ is approximately $-15°$ (leading PF_D). This "leading overcompensation" by approximately 15° is needed so the TCR branch can fine-tune the PF_D to a value closer to unity.

As presented earlier, the only time the TSC branch should be switched is during the zero crossing of the current so transients are avoided. The PIC is programmed is examine θ every 5 seconds, but this time value is arbitrary and should be configured to best match the specific application. It is recommended that a time value not less than 5 seconds be used so that the PIC is not continuously updating the TSC branches. When the time interval expires the PIC will wait for the next current zero crossing. This is accomplished by having the PIC poll the current waveform input looking for a state change. For example, if the current waveform is initially read as HIGH, then when the waveform switches to a LOW, a zero crossing has occurred. Having already made its TSC switching decision based upon θ, the PIC will either switch in or out a single TSC branch at this exact instant. Fig. 3.4.2 summarizes the steps taken by the PIC during TSC branch switching.

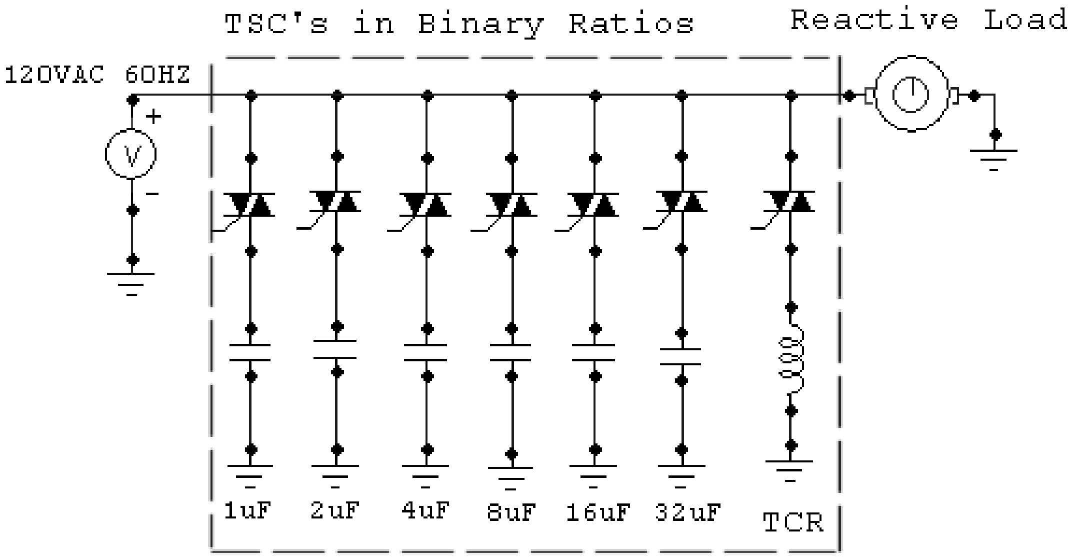


Fig. 3.4.1. TSC Branches Configured in Binary Steps

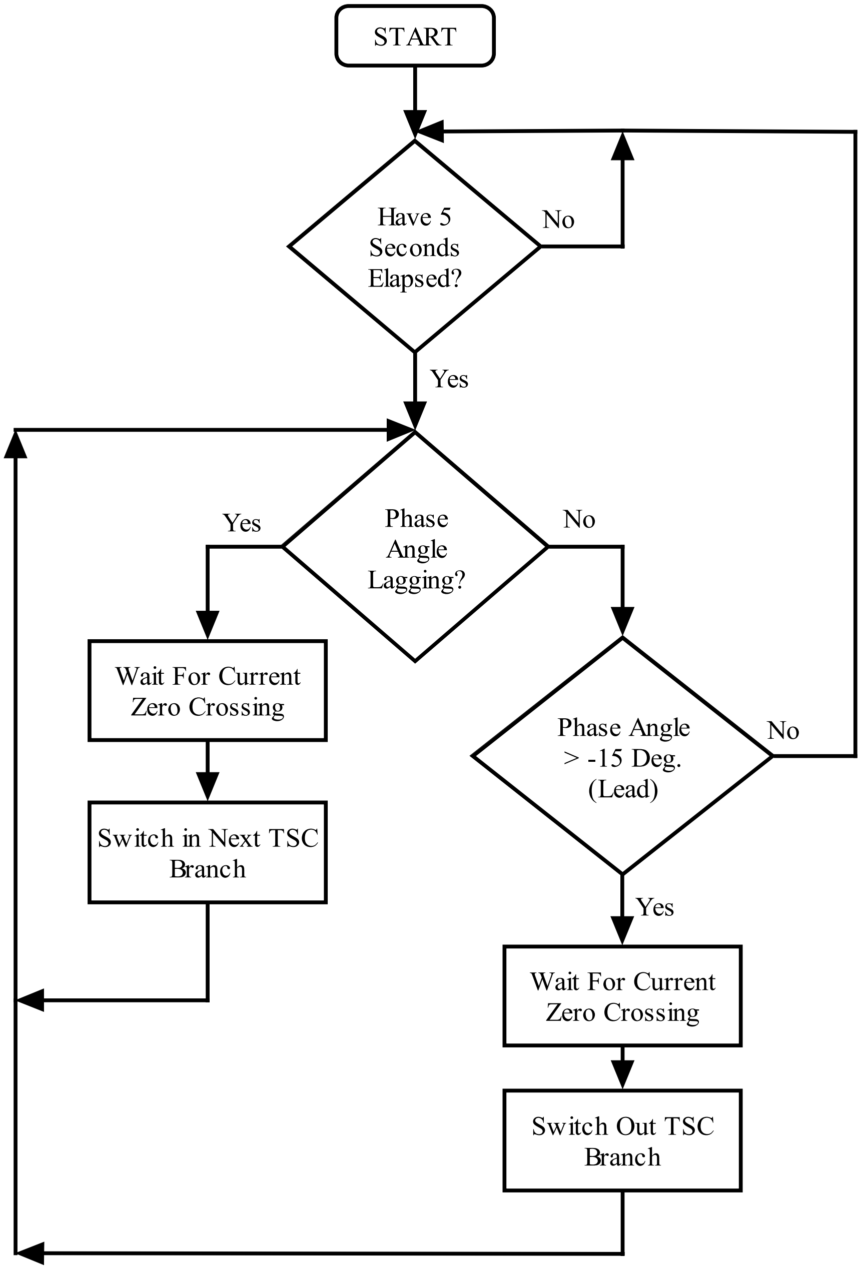

START
Have 5 Seconds Elapsed?
No
Yes
Phase Angle Lagging?
Yes
No
Wait For Current Zero Crossing
Switch in Next TSC Branch
Phase Angle > -15 Deg. (Lead)
No
Yes
Wait For Current Zero Crossing
Switch Out TSC Branch

Fig. 3.4.2. Software Steps Taken To Switch In/Out a TSC Branch

3.4.3 Fuzzy Control of the TCR Branch

The purpose of the TCR branch is to fine-tune the PF_D to a value that is near unity. This is accomplished by having the PIC determine the optimum firing angle for the TCR branch every cycle. Since it takes a complete cycle for the PF_D to be calculated, the firing angle for the TCR is updated once per cycle, also. For example, the TCR is controlled on the half-cycles using the firing angle that was calculated during the previous cycle. This open-loop control scheme guarantees that the TCR will be controlled within one cycle.

To generate the firing angle α, the load's phase angle difference θ is used as an input to a one input – one output TSK (Takagi – Sugeno – Kang) fuzzy controller [25]. The output of the fuzzy controller is the firing angle α that is translated into a control time delay, which is applied directly to the TCR thyristor. Since α directly affects the susceptibility of the inductor/reactor, the current and the vars absorbed can be controlled continuously to provide a finely tuned PF_D.

A TSK fuzzy controller was chosen for its simplicity and lack of complex mathematical calculations. The TSK model does not rely on a complex mathematical defuzzification method, but instead, maps the crisp input using simple rules or functions to a defuzzified output. The TSK model is described by the following general rule (Equation 3.7) where x and y are crisp inputs and f (x ,y) is a defined function or rule mapping. A and B are defined fuzzy input rules and z is a crisp output.

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B\text{, THEN } z = f\ (x, y) \qquad (3.7)$$

Since the TCR fuzzy controller is a one input – one output TSK controller, its rules follow the general form (Equation 3.8), with x representing θ and z representing α.

$$\text{IF } x\text{, THEN } z = f\,(x) \qquad (3.8)$$

The TCR fuzzy controller consists of 9 rules, which are graphically shown in Fig. 3.4.3 along with their respective membership functions. In this example, the rules follow a one-to-one mapping between the input and the output, but this does not necessarily always have to be true. The rules and membership functions require manual tuning so that the optimum firing angle for the particular reactive load is centered in the output membership functions. As shown in Fig. 3.4.3, the optimum firing angle for the particular reactive load is approximately 130°.

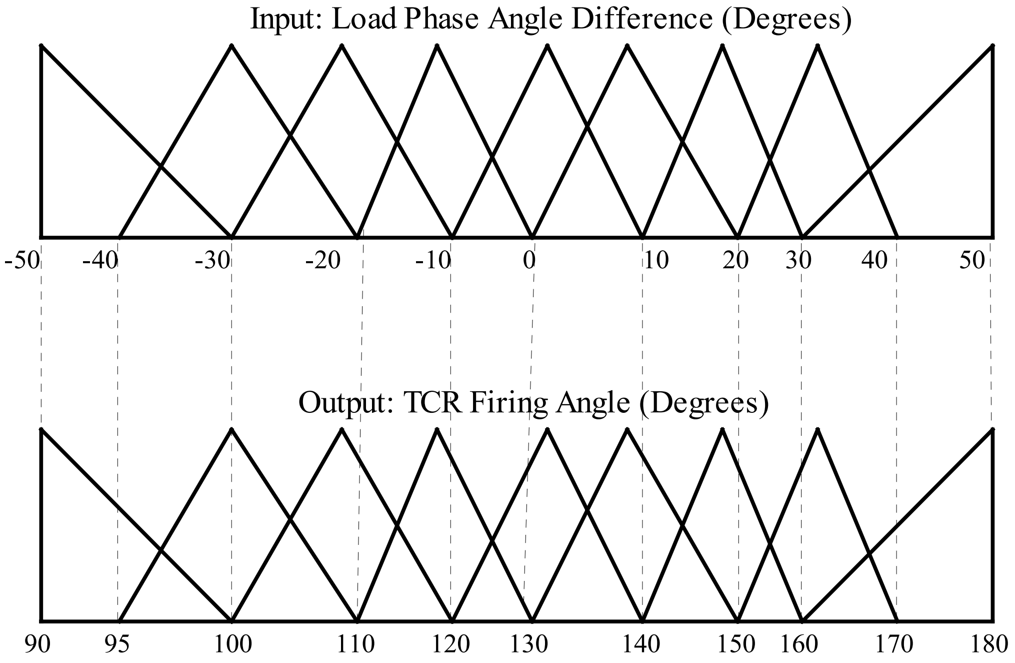

Input: Load Phase Angle Difference (Degrees)
-50
-40
-30
-20
-10
0
10
20
30
40
50
Output: TCR Firing Angle (Degrees)
90
95
100
110
120
130
140
150
160
170
180

Fig. 3.4.3. Fuzzy Membership Functions and Rules for TCR

The fuzzy controller is implemented as a single function that accepts the crisp input θ and returns the crisp output α. The input and output membership functions are specified using arrays of constants that allow for easy modification. Likewise, the fuzzy rules are constants in an array that can also be modified easily. The fuzzy function accepts θ and compares it to the input membership functions. The particular membership functions that encompass the input are flagged and their corresponding rules are then followed. The rules are mapped to the appropriate output membership functions and a crisp value for ∝ is generated. Using Eq. 3.9, ∝ is converted into a firing time delay.

$$T = [\theta(1/60)]/360 \tag{3.9}$$

Once ∝ has been determined and the firing delay calculated, the PIC will initialize an interrupt which becomes responsible for handling the firing of the TCR for the next cycle. Once the cycle has ended, ∝ will be updated once again, and the same interrupt will be reinitialized. The interrupt is initialized with the time to wait (firing time delay) after the zero crossing of the current. To achieve this, Timer 0 on the PIC is initialized with the count that will cause the timer to interrupt after the delay time has expired. This interrupt outputs a control signal to the MOC3051 triac driver thus turning on the TCR thyristor. As previously mentioned, no explicit "turn-off" control is needed because the thyristor will turn off at the next zero crossing of the current. Fig. 3.4.4 on the following page summarizes the entire sequence of events that occur.

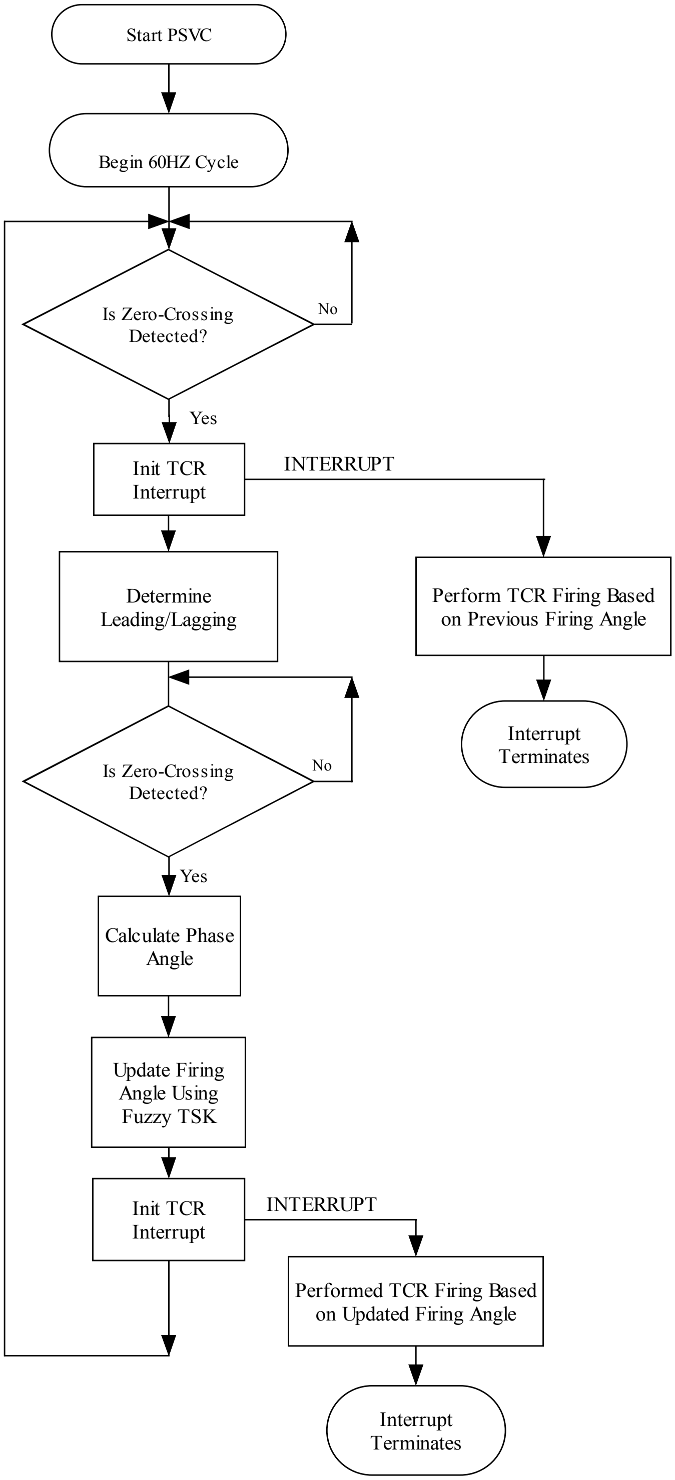

Start PSVC
Begin 60HZ Cycle
Is Zero-Crossing Detected?
No
Yes
Init TCR Interrupt
INTERRUPT
Perform TCR Firing Based on Previous Firing Angle
Interrupt Terminates
Determine Leading/Lagging
Is Zero-Crossing Detected?
No
Yes
Calculate Phase Angle
Update Firing Angle Using Fuzzy TSK
Init TCR Interrupt
INTERRUPT
Performed TCR Firing Based on Updated Firing Angle
Interrupt Terminates

Fig. 3.4.4. Sequence of Events for Controlling the TCR Branch

3.4.4 LCD (Liquid Crystal Display) Output

The LCD is updated every three seconds using an interrupt on the PIC. This is accomplished by programming the PIC's timer 2 to interrupt every 30ms. A variable counter is used to count the interrupts and after 100 have occurred, the LCD display is updated with the phase angle difference θ, firing angle α, PF_D, and the number of TSC branches currently in use The interrupt is synchronized so as to not interrupt during the 60HZ cycle, but instead, will wait until the end of a cycle. Since writing to the LCD can take a considerable amount of time, it is recommended that the LCD not be updated too often and that the amount of text printed should be kept at a minimum. The LCD is wired to the PIC output port B.

3.4.5 Software Implementation

The software for the PIC was developed using the C programming language and a specific PIC C compiler, known as "PICC", which is distributed by CCSInfo [26]. This compiler compiles the C source code and generates the equivalent assembly instructions. A software PIC programmer known as MPLAB, distributed by Microchip, was chosen as the tool that would program the PIC microprocessor once the assembly code had been generated [27]. MPLAB transfers the microprocessor instructions from the PC to the PIC via a serial cable. Once the PIC has been programmed, the PC, serial cable, and developer software is no longer needed. The PIC is capable of executing completely autonomously. Fig. 3.4.5 on the following page summarizes the functionality of the microprocessor. The entire PSVC source code listing can be found in the appendix B.

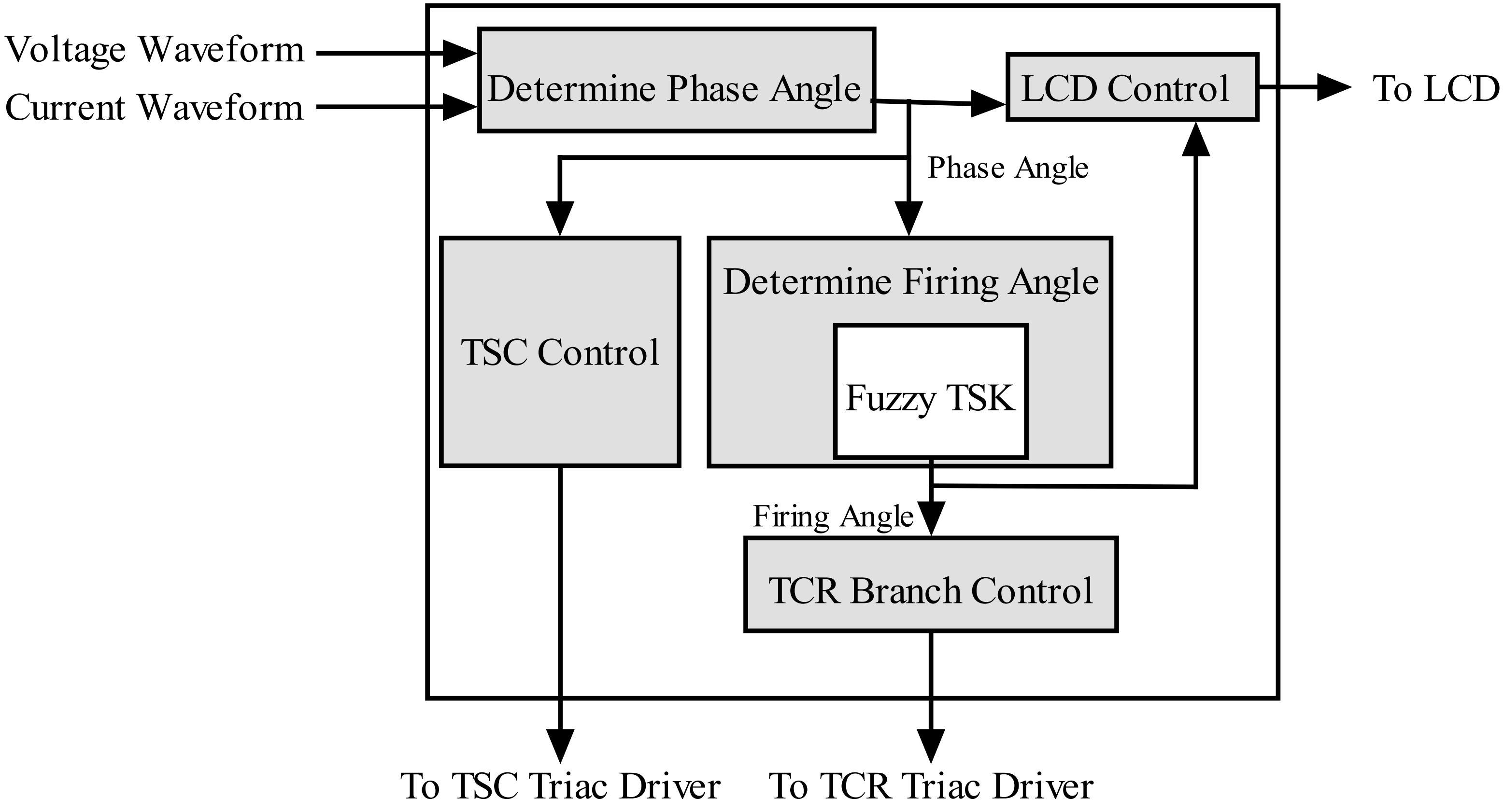


Fig. 3.4.5. Microprocessor Functional Block Diagram

3.5 Design of a Printed Circuit Board

A printed circuit board was designed and created using the etcher/plotter and drilling machine at West Virginia University. The first step in creating a printed circuit board is to draw the layout of the copper traces using a software tool. In this case, the software tool *Traxmaker* by MicroCode Engineering was utilized to layout the copper traces.

If possible, it is recommended that a single (bottom) layer printed circuit board be designed. A single layered board is less complex and allows for easy soldering underneath the board. The *Traxmaker* design is given in appendix C. The blue copper traces indicate the bottom layers, while components such as chip sockets, resistors, etc., were placed on the topside of the board. When high-powered components are being used such as the AC alternistors and triac drivers, it is recommended that the copper trace size be increased to at least 50 mils. For digital logic, small traces such as 15 mils are adequate.

CHAPTER 4 - PSVC EVALUATION

The PSVC was evaluated in terms of performance and cost. To evaluate the performance of the PSVC, a small 55-watt AC motor was chosen as the "test" reactive load so that RMS current reduction and displacement power factor correction (PF_D) could be measured. Specific design steps and experimental results from the AC motor and PSVC are supplied. The described PSVC has been designed for maximum flexibility because it utilizes six TSC branches sized in binary steps and also provides for the use of the TCR branch for when a finely tuned power factor is needed.

4.1 PSVC Design for an AC Motor

The 55 Watt AC motor was first introduced in Section 3.2.2 when control verification simulations were discussed. Refer to Table 3.2.1 for the characteristics of the AC motor.

As described, the PSVC can provide compensation up to 64uF in discrete steps of 1uF. Since the AC motor absorbs 78 vars of reactive power, then the TSC branch(es) must provide at least 78 vars of reactive power to raise the PF_D. Using Eq. 4.1 (Q=Needed Vars, V=Supply Voltage, f=60HZ, C=Needed Capacitance) the optimum capacitor value can be calculated. Theoretically, the capacitance of 14uF will provide the 78 leading vars needed by the AC motor.

$$C = Q/(2\pi f V^2) \qquad (4.1)$$

A TSC overcompensation will be needed by the TCR branch to yield a finely tuned PF_D to as close to unity as possible. Programming the PSVC to use this overcompensation only requires modifying a constant variable in the source code of the PIC. The constant variable is used to define an "allowable overcompensation range" for the phase angle difference θ. Since the PSVC dynamically assigns TSC branches based upon the current value of θ and its "allowable overcompensation range", the actual chosen overcompensation by the PSVC cannot be predicted exactly. However, in the case of the AC motor, the "allowable overcompensation range" was set to no more than -15° (PF_D = 0.96 leading) meaning that a TSC compensation of approximately 20uF or less should be expected. Eq. 4.2 (P=Real Power) verifies this prediction by showing the overcompensating vars produced as a result of setting the "allowable overcompensation range" equal to 0.96 leading.

$$Q_{OVER} = \sqrt{\;}[(P/0.96)^2 - P^2] \tag{4.2}$$

Using Eq. 4.3, which is derived from the conduction angle equation in [28], a theoretical firing angle for the TCR branch (X_{TCR} = 62.5) can be calculated and used as an initial "first guess" when fine-tuning the fuzzy membership functions and rules. A major strength of the PSVC is that any load change, whether it be leading or lagging, will be compensated dynamically by the PSVC.

$$X_{TCR(\alpha)} = [2(\pi - \alpha) - \sin(2(\pi - \alpha))/(\pi X_{TCR})]^{-1} \tag{4.3}$$

Fig. 4.1.1 shows the entire reactance range for the particular TCR used in the prototype PSVC. As the firing angle α increases, the reactance of the TCR also increases. When α = 90°, the TCR is considered to be "full-on", allowing current to flow freely (low reactance). When α = 180°, the TCR is considered to be "full-off", and no current is allowed to flow (high reactance). This controllable current is equivalent to a variable inductance because the amount of reactive power absorbed by the TCR can be controlled over a particular range. Fig. 4.1.2 shows the reactive power absorption range for the specific TCR used in the prototype. Note that firing angle values between 90° and 150° are the most useful for controlling the inductor.

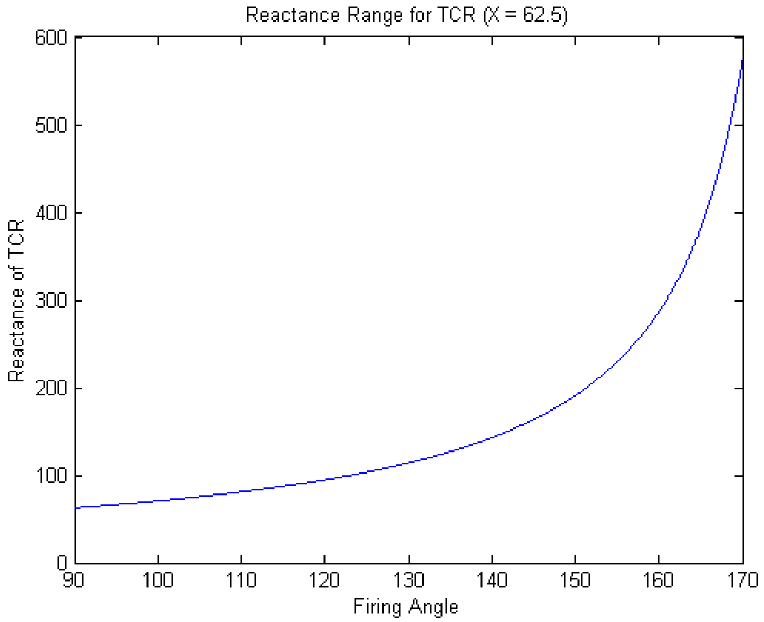


Fig. 4.1.1. Reactance Range for the PSVC TCR

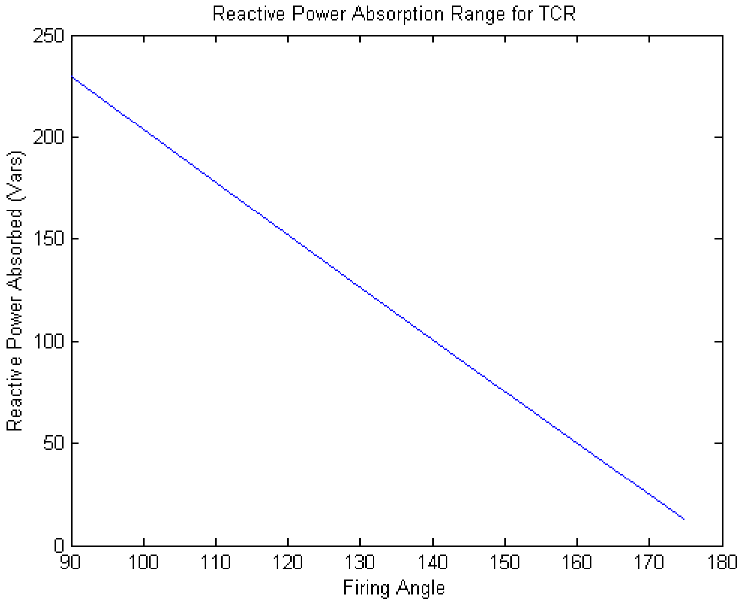


Fig. 4.1.2. Reactive Power Absorption Range for the PSVC TCR

The fuzzy membership functions and rules will need to be configured to yield the desired center-firing angle α that is appropriate for the reactive load. The following presents the membership functions and rules that were used for the AC motor and a center firing angle α of 115°.

Nine membership functions were used for the phase angle difference θ. The valid range of θ is from -20° (leading) to 25° (lagging). As shown in Fig. 4.1.3, triangle membership functions were used and named from left to right as: LargeLeading, MediumLeading, SmallLeading, Neutral, SmallLagging, MediumSmallLagging, MediumLagging, LargeLagging, and LargestLagging.

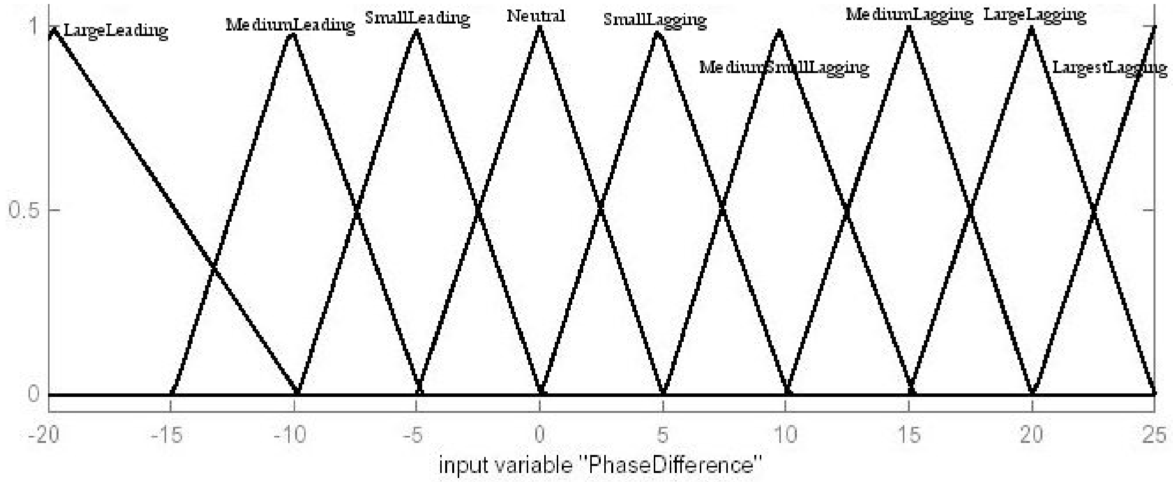


Fig. 4.1.3. Fuzzy Input: Phase Angle Difference θ

Ten membership functions were used for the fuzzy output, Firing Angle α. These membership functions are shown in Fig. 4.1.4. From left to right, the membership functions were named: ON, Mostly_ON1, Mostly_ON2, Medium_ON1, Medium_ON2, Medium_ON3, Slightly_OFF1, Slightly_OFF2, Mostly_OFF, and OFF. The OFF membership function encompasses firing angles from 128° to 180° because these values are not particularly effective in controlling the TCR.

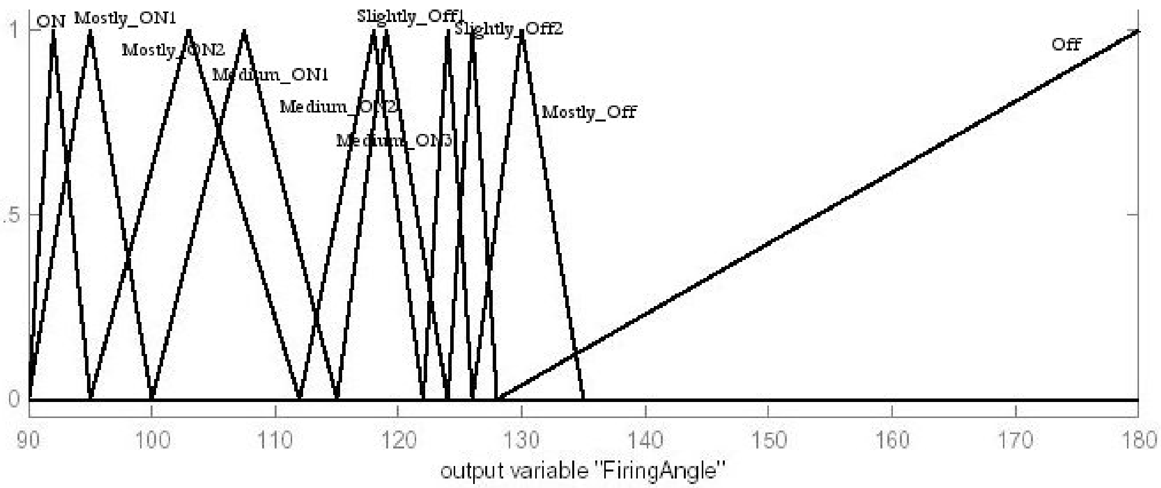


Fig. 4.1.4. Fuzzy Firing Angle Output α

The TSK rules used are listed below:

IF PhaseDifference is LargeLeading, THEN FiringAngle ON
IF PhaseDifference is MediumLeading, THEN FiringAngle is Mostly_ON1
IF PhaseDifference is SmallLeading, THEN FiringAngle is Mostly_ON2
IF PhaseDifference isNeutral, THEN FiringAngle is Medium_ON1
IF PhaseDifference is SmallLagging, THEN FiringAngle is Medium_ON2
IF PhaseDifference is MediumSmallLagging, THEN FiringAngle is Medium_ON3
IF PhaseDifference is MediumLagging, THEN FiringAngle is Slightly_OFF1
IF PhaseDifference is LargeLagging, THEN FiringAngle is Slightly_OFF2
IF PhaseDifference is LargestLagging, THEN FiringAngle is Mostly_OFF

The surface coverage of this fuzzy control scheme is shown in Fig. 4.1.5. Note that near a 0° phase angle difference the firing angle is between 110° and 115° degrees. This fuzzy controller is most effective between the firing angles of 95° to 120° because these firing angles have the most effect upon the reactance of the TCR, which is illustrated in Fig. 4.1.1.

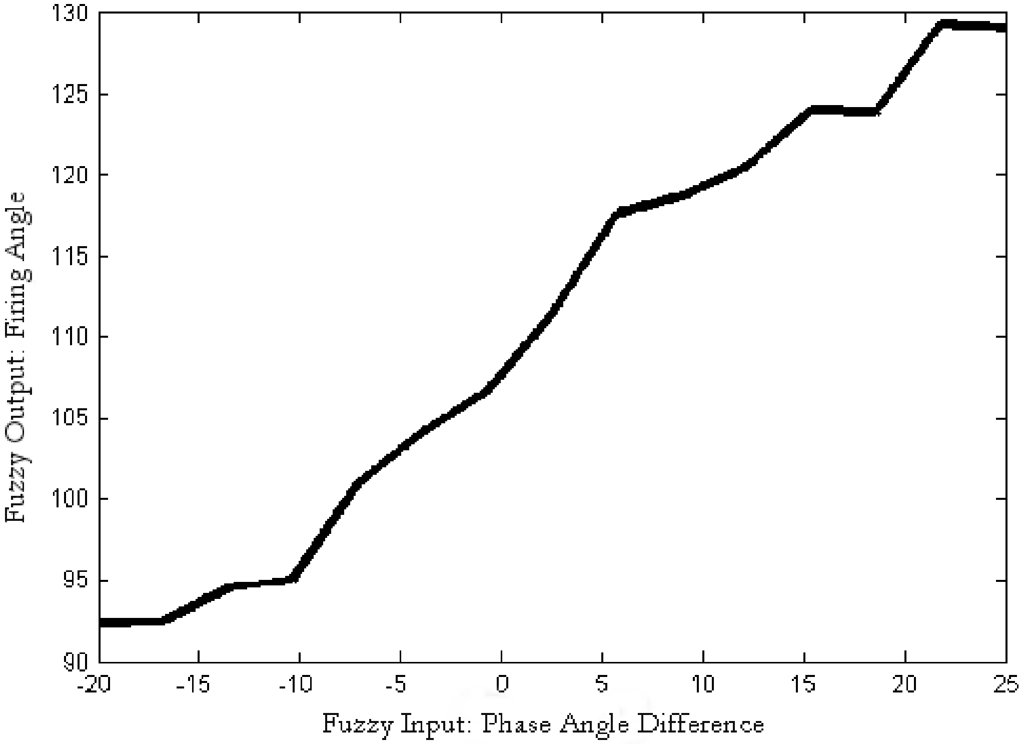


Fig. 4.1.5. Fuzzy Controller Surface Values

4.2 PSVC and AC Motor Experimental Results

The PSVC experimental results obtained are presented in this section. All measurements were taken using a *Yokogawa PZ4000* power analyzer [29]. PSVC and load measurement results are presented in the following order: AC Motor Only, TSC Overcompensation Only, Both TSC and TCR (no control scheme), and both TSC and TCR (fuzzy control scheme) with varying load. A subsection is also devoted to true power factor (PF_{TRUE}) and the elimination of harmonics.

4.2.1 AC Motor Only

The AC motor characteristics were described in Table 4.1.1. Before activating the PSVC, the AC motor is first turned on. The AC motor has a supply of 120V_{AC} RMS has a lagging PF_D of 0.58.

4.2.2 TSC Overcompensation Only

When the PSVC is first activated it will determine the number of TSC branches to utilize, ensuring that the phase angle difference will stay within the "allowable overcompensation range". For the AC motor, an "allowable overcompensation range" of –15° (leading) has been defined. The PSVC switches in 22uF resulting in a –13° phase angle difference (0.96 leading PF_D). The PSVC will verify that the overcompensation is within the allowable range once every 5 seconds. As discussed previously, this 5-second range is arbitrary, but it should not be configured for less than a couple of seconds. Table 4.2.1 summarizes the load characteristics with the PSVC's TSC branch activated and compares those values with just the AC motor (no TSC).

Motor Characteristic	**Without TSC**	**With TSC**
Total Power	95 VA	**58 VA**
Real Power	55 Watts	**56 Watts**
Reactive Power	78 Vars	**16 Vars**
Current Draw	0.8 A	**0.65 A**
Power Factor (PF_D)	0.58 Lag	**0.96 Lead**

Table 4.2.1. AC Motor with TSC Branch Activated

4.2.3 Both TSC and TCR (No Control Scheme)

Once the TSC branch settles on a specific capacitance overcompensation, the TCR branch will activate and fire the inductor based upon the determined firing angle α. For this particular test, a firing angle α was fixed so that the TCR RMS current could be measured as well as the PF_D. Three measurements were taken with α = 90° (TCR full on), α = 115° (TCR medium on), and α = 140° (TCR mostly off). A measurement of α = 180 was not taken because that case would be equivalent to only the TSC activated (TCR off), as described in 4.2.2. Table 4.2.2 summarizes the measurements taken.

Motor Characteristic	**α = 90°**	**α = 115°**	**α = 140°**	**TCR Off**
Real Power	68 W	60 W	56 W	55 W
Reactive Power	98 Vars	8.5 Vars	11.4 Vars	78 Vars
Current Draw	0.71 A	0.59 A	0.63 A	0.8 A
Power Factor (PF_D)	0.57 Lag	0.99 Lag	0.98 Lead	0.58 Lag

Table 4.2.2. TCR Measurements Taken

4.2.4 Both TSC and TCR (Fuzzy Control Scheme) with Varying Load

The fuzzy engine, with center firing angle α = 115°, was activated. The settled 119° firing angle yielded a PF_D of 0.99 lagging. To verify the fuzzy logic control scheme, a slight reactive load change was implemented that caused the AC motor's PF_D to rise slightly. This load change was implemented by placing a 4uF capacitor in parallel with the AC motor. This rise in PF_D caused the fuzzy engine to settle on a new firing angle of α = 112°. This smaller firing angle was needed so that the added reactive power could be absorbed by the TCR, thus still maintaining a near unity PF_D. The smaller firing angle decreases the reactance of the TCR, allowing more current to flow. Table 4.2.3 shows the detailed characteristics of the reactive load change and fuzzy controller behavior.

Motor Characteristic	Original	Original Load	Modified	Modified Load
Real Power	56 W	60 W	56 W	61 W
Reactive Power	78 Vars	8.5 Vars	54 Vars	8.7 Vars
Current Draw	0.8 A	0.60 A	0.65 A	0.59 A
Power Factor (PF_D)	0.58 Lag	0.99 Lag	0.72 Lag	0.99 Lead

Table 4.2.3. Fuzzy Controller Behavior with Slight Load Change

4.2.5 True Power Factor (PF_{TRUE}) and the Elimination of Harmonics

THD (total harmonic distortion) is composed of the voltage harmonic distortion (V_{THD}) and the current harmonic distortion (I_{THD}). The voltage THD (V_{THD}) is a measure of the harmonic components in the voltage waveform. The current total harmonic distortion (I_{THD}) is a measure of the effective value of the harmonic components in the current waveform [30].

The AC motor alone produces an I_{THD} of 8%, with a negligible V_{THD}. When the PSVC is applied to the AC motor, the I_{THD} rises from 8% to 20% due to increases in the 3rd and 5th harmonics, while the V_{THD} still remains negligible. When harmonics are present the effectiveness of the PSVC is lessened and the true power factor (PF_{TRUE}) must be considered in lieu of the displacement power factor (PF_D). In the special case of no harmonics, $PF_{TRUE} = PF_D$, because only the 60HZ fundamental frequency is considered. However, when harmonics are present, the PF_{TRUE} formula from [31] must be considered. This formula, which considers all frequency components of the voltage and current, is given in Eq. 4.4. Table 4.2.4 summarizes the PF_D and PF_{TRUE} measurements that were taken of the AC motor. For the motor alone, the PF_D can be approximated to PF_{TRUE} because the 8% I_{THD} effect is negligible. However, in the case of the AC motor and the PSVC, the 20% I_{THD} has approximately a 12% reduction effect upon the PF_D.

$$PF_{TRUE} = kW / KVA = P / (V_{RMS} * I_{RMS}) \qquad (4.4)$$

	Without PSVC	**With PSVC**
Power Factor (PF$_{TRUE}$)	0.58 Lag	0.87 Lag
Power Factor (PF$_{D}$)	0.58 Lag	0.99 Lag
RMS Current	0.8 A	0.59 A
Harmonic Dist. (I$_{THD}$)	8 %	20 %

Table 4.2.4. AC Motor Measurements of Both PF$_D$ and PF$_{TRUE}$

Eliminating the harmonics is traditionally accomplished by designing specific harmonic filters to eliminate any harmonics that might be created. In this case of the AC motor, filter design is difficult because motor characteristics are unknown, as well as the exact cause of the harmonics. The harmonics could be caused by either of the following:

a. Motor Saturation

b. Resonance Created by the Motor and PSVC

However, since it is known that the harmonics occur in the 3rd (180 HZ) and 5th (300 HZ) frequencies of the fundamental, a passive harmonic filter can be designed to reduce the total (V_{THD} + I_{THD}) harmonic distortion to less than 5%, and I_{THD} to less than 20%, as outlined in IEEE standard 519-1992 [32]. In [33] I_{THD} ranges are given as the following: **Low** (I_{THD} ≤ 20%), **Medium** (20% < I_{THD} ≤ 50%), and **High** (I_{THD} > 50%). Although the PSVC and motor narrowly fall into the "acceptable-low harmonic range" as outlined in [32] and [33], action must be taken to reduce the harmonics so as to maintain high power quality. Several solutions are described below.

Solution 1: Passive Shunt Harmonic Filter

A shunt harmonic filter, consisting of a series inductor and capacitor (LC), can be used to effectively remove harmonics near the source. The inductor and capacitor are designed to resonate at a particular frequency. Shunt filters will create leading vars that must be considered in the PSVC design. One such shunt filter configuration is shown in conjunction with the PSVC in Fig. 4.2.2. In the case of the PSVC, the 3rd and 5th harmonics make up most of the current distortion. Often inductors can be placed in series with PF correcting capacitors to eliminate any resonance.

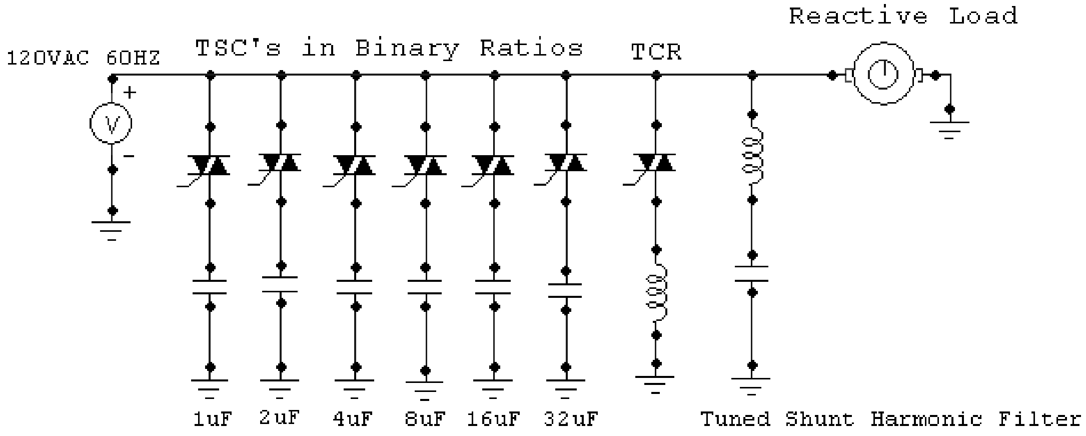


Fig. 4.2.2. Placement of the Passive Filter with the PSVC and Reactive Load

Solution 2: Passive Series Harmonic Filter

Similar to the shunt filter, series harmonic filters consist of a parallel inductor-capacitor (LC) circuit that is designed to resonate at a particular frequency.

Solution 3: Adaptive Injection

Harmonic currents can be eliminated by injecting currents from a transformer which are 180° out of phase from the reactive load's harmonic currents, thus canceling the harmonic current components. Constant monitoring of the load current is required for intelligent current injection. Adaptive injection techniques are currently under development. In [34] a current harmonic injection system is being developed to help lessen the noise of motor drives by utilizing a lookup table of sinusoidal values that describes the switching pattern needed for current injection.

The most common type of harmonic filter is the single-tuned ("notch") filter. This filter is popular because of its economical advantages and performance. Notch filters are series tuned and connected in shunt with the load so as to divert harmonic currents from the AC line into the filter. One common method of creating a notch filter is to place an inductor in series with a power factor compensation capacitor bank. This inductor in series with the capacitor will ensure that any resonance will be kept below the resonant frequency. In [35] it is advised that an inductor sized to 20% of the capacitor reactance will be needed to prevent third harmonics at 60HZ. Due to time and hardware limitations the notch filter was not implemented in the PSVC prototype. However, it is the opinion of this author that an adaptive harmonic filter would be most advantageous for the prototype PSVC. Rather than relying on load-specific passive harmonic filters, efforts are currently being focused on designing an active harmonic filter. The active harmonic filter would reshape the current waveform making it more sinusoidal thus causing PF_{TRUE} = PF_D. One such active harmonic filter design being investigated uses a power factor controller (PFC) integrated circuit, which has traditionally been applied to DC loads. Developing an economical adaptive harmonic filter is the next step in development of the PSVC prototype. Such an adaptive harmonic filter is a new and novel idea currently only in the research phase.

4.3 PSVC Cost Evaluation

The components of the PSVC were priced individually with consideration given to quantity and characteristics. A complete component list is given in appendix D. Smaller semiconductor components, not including the microprocessor and thyristors (alternistors), have been neglected. Since the capacitors and inductor that make up the PSVC branches are the most expensive items, they will draw the most focus. Finally, power factor correction cost savings from a major power utility will be presented.

4.3.1 Cost of Capacitors

Capacitor cost data was taken from two leading mail-order retailers: Newark (www.newark.com) and DigiKey (www.digikey.com). Three capacitor manufacturers were chosen for their differing price spread: Panasonic (cheapest), Mallory (moderate) and GE (most expensive). Individual capacitor data can be found in appendix E. Capacitors rated at $250V_{AC}$ and less were chosen because the operating voltage of the PSVC prototype is $120V_{AC}$ to $220V_{AC}$. Fig. 4.3.1 illustrates the price of a capacitor versus its maximum var rating (92 Var – 2.3kVar), when a quantity of one is purchased (Q = 1). As shown in the figure, the cost of the capacitors remains relatively low, even for the higher var rated capacitors. Once the var rating reaches approximately 1.5kVar, the cost increases quickly for additional vars. The best-fit polynomial for the capacitor cost (Q = 1) is given in Eq. 4.5. When Q = 1, the average capacitor cost is $15.73.

$$y = - 0.0025x^5 + 0.0601x^4 - 0.6442x^3 + 3.1549x^2 - 5.9987x + 6.9861 \tag{4.5}$$

Fig. 4.3.1 also illustrates the individual price of a capacitor when a quantity of a thousand (Q = 1000) is purchased. As expected, this decreases the individual cost of each capacitor, lowering it on average by $10.25. The best fit polynomial for the capacitor cost (Q = 1000) is given in Eq. 4.6. When Q = 1000 the average capacitor cost is $5.48.

$$y = - 0.0006x^5 + 0.0131x^4 - 0.1351x^3 + 0.628x^2 - 0.9902x + 2.3282 \tag{4.6}$$

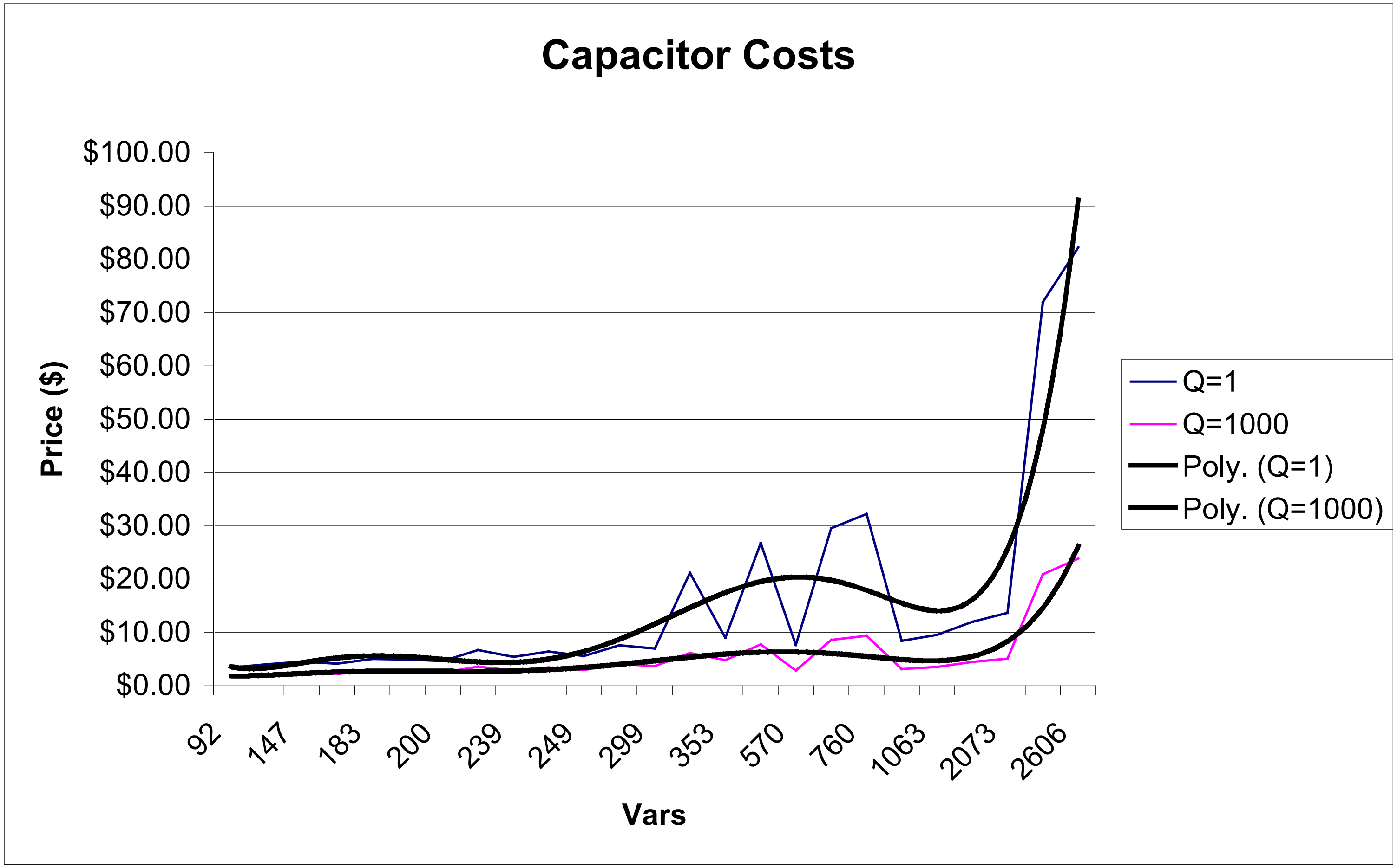


Fig. 4.3.1. Cost of Capacitors for Q = 1 and for Q = 1000

4.3.2 Cost of TCR (Inductor)

The cost of the TCR (inductor) can be considerable if custom made by a third party for small quantities. A price quote was obtained from *Toroid,* a company located in Maryland. This company provided a price quote of $55 for a single 166mH inductor. In large quantities the price would drop below $25 each, possibly even lower. For smaller quantities such as prototypes, the author recommends winding your own inductor to the needed size. An E-Core and E-Core bobbin can be purchased for $2.40 and $0.80 respectively. When purchased in a quantity of 1000, the price drops to $1.23 for the E-Core and $0.43 for the bobbin. Once purchased, wind the bobbin until the desired inductance is created.

4.3.3 Cost of Semiconductors

The PIC16F877 microprocessor can be purchased for $8.00 and the power thyristors (alternistors) can be purchased for $2.00 each. The MOC3051 triac drivers cost between $0.39 (Q=1000) and $0.90 (Q=1) depending upon the quantity ordered.

4.3.4 Total PSVC Cost

The approximate total cost of the PSVC can be found by summing the cost of the larger components. The quantities given are for the single hardware prototype that was developed. The total cost for the developed PSVC was $132.00. The prototype is capable of providing 1.5kVar of reactive power to a specific load.

Item	Quantity Used in Prototype	Cost
PIC16F877 Microprocessor	1	$8.00
Thyristors / Alternistor	7	$14.00
Triac Drivers (MOC3051)	7	$6.30
Capacitors	6	$95.38
Inductor (E-Core and Bobbin)	1	$3.20
Wound Inductor (Optional)	1	$55.00
LCD	1	$5.00

Table 4.3.1. Costs of Components Used in the PSVC Prototype

If PSVC's were to be manufactured in large quantities, the individual component costs would be considerably lessened. Table 4.3.2 shows the costs of one PSVC if 1000 units were to be manufactured. For a quantity of 1000, the individual PSVC would cost $55.00, assuming you wind your own inductor. A custom made inductor adds $25 to the price of the PSVC. As previously mentioned, the PSVC is capable of supplying 1.5kVar to a reactive load.

Item	Cost
PIC16F877 Microprocessor	$6.64
Thyristors / Alternistor	$8.12
Triac Drivers (MOC3051)	$2.59
Capacitors	$32.88
Inductor (E-Core and Bobbin)	$1.66
Wound Inductor (Optional)	$25.00
LCD	$2.50

Table 4.3.2. Component Costs per PSVC When Q = 1000

4.3.4 Cost Effectiveness of Power Factor Correction

Commercial and industrial power customers would expect to see a quick rate of return on PSVC installations. Often, power utility companies charge commercial and industrial customers based on their total power demand, including vars. One such utility, *Entergy - Arkansas*, charges its commercial and industrial customers based on total KVA consumed [36]. Their rate formula is shown in Eq. 4.7. Also, when grid expansion is physically difficult or too costly, PSVC's could be utilized to increase power flow capacity, thus offsetting the immediate need for expansion.

$$\text{Billing KW} = \text{Metered KVA} * 0.9 \tag{4.7}$$

Another example is the *Dixie Escalante Rural Electric Association (Utah)* and its strict guidelines for small commercial (50kW maximum demand) Arizona customers.

> "POWER FACTOR: The customer shall maintain *not* less than a 95% lagging power factor. Those customers which are determined to have power factors of less than 95% lagging shall have the demand for billing purposes increased by 1% for every 1% that the power factor is less than 95%." [37]

More recently the *St. George, Utah, Municipal Power and Light System* has also enacted similar power factor penalties for small general service (60 kW maximum demand) customers.

> "Power Factor Adjustment: If the customer power factor is found to be less than 95 percent, the customer will be penalized 1 percent of the overall billing for each 1 percent below the 95 percent power factor." [38]

The power utility companies are aware that increasing the power factor to near unity can reduce distribution system losses. In [38] it is reported that losses can vary between 2% and 8% of the load kilowatt-hours. Reducing losses by 50% would result in a 1% to 4% electric bill savings.

> "For large industrial plants this [reducing losses] can lead to a welcomed savings. Losses and, hence savings, in rural distribution or commercial buildings, served at utilization voltages, will be more substantial." [39]

CHAPTER 5 - CONCLUSION

5.1 Summary

This thesis describes a "personal" static var compensator (PSVC) that has the ability to compensate the displacement power factor (PF_D) to near unity of reactive loads. The PSVC is responsible for distributive var control and is modeled after the well-known static var compensator (SVC), an early FACTS device. The PSVC compensates the PF_D of the reactive load by intelligently switching TSC (thyristor switched capacitor) and TCR (thyristor controlled reactor) shunt branches. The TSC branches consist of capacitors sized in binary ratios to provide discrete steps of leading vars. The TCR branch consists of a fine-tuning inductor that is controlled continuously using a firing angle and an open-loop fuzzy logic control scheme.

The proposed PSVC incorporates the concepts of the adaptive power factor controller and merges them with the SVC. The result is a robust device capable of providing PF_D compensation to reactive loads in an economical fashion. The displacement power factor is extremely important to industrial customers and utilities, as well as becoming increasingly important to small commercial and residential consumers. Poor PF_D's are associated with increased power costs, reduced system capacity, and diminished power quality.

A 55-Watt AC motor was chosen as the test reactive load for evaluation of the PSVC. The PSVC was able to reduce load RMS current draw by 26% while increasing the load's PF_D to 0.99 lagging. This was accomplished by intelligently switching TSC branches while continuously controlling the current though the TCR branch.

5.2 Future Work

Although within acceptable limits, the motor's current total harmonic distortion (I_{THD}) rose from an 8% (no PSVC) to a 20% (with PSVC). Since harmonics are often the cause of overheating and motor wear-and-tear, the generation of harmonics needs to be addressed. Although not directly related to the PSVC, the power industry has a large interest in the reduction of harmonics because of the ever-increasing usage of consumer electronic devices. These devices are notorious for inserting large harmonic distortions into the power system.

The "hot-topic" of harmonic suppression research is the development of an adaptive harmonic filter that intelligently monitors the load and injects canceling currents, thus eliminating any harmonics that are present. Such a device working in conjunction with the PSVC could raise true power factors (PF_{TRUE}) to near unity. The adaptive harmonic filter could be designed to either work in conjunction with the PSVC or independently. Either way, the adaptive filter could be evaluated using the PSVC and a reactive load.

5.3 Lessons Learned

One difficulty that needed to be overcome was the design and development of complex software for limited microprocessors. Although the PIC is a very robust microprocessor, complex software such as the fuzzy controller is still challenging to design and implement due to RAM limitations. Also, limitations (mainly in the timers) were noticeable because the chosen microprocessor is only 8-bit. Consideration should be given to using a 16-bit PIC microprocessor for future designs. The clock speed of 20MHZ, however, is considerably fast enough for all mathematical calculations needed.

Another difficulty overcome was the move from a low power to a higher-powered application. Initially, the PSVC was implemented and tested using only lower powered components. Once the design was deemed feasible, the lower powered components were replaced with higher-powered components with slight design changes. Using the higher-powered application required developing different measurement techniques as well as different hardware implementations.

BIBLIOGRAPHY

[1] Hingorani, N.G., "Flexible AC Transmission", IEEE Spectrum, April 1993, p. 40-44

[2] NASA Success Story - Power Factor Controller Commercialized, Success Story ID # 2200531,Press Release, ref: 06/11/98, **Industry Partner:** Power Efficiency Corporation,**NASA Partner:** Marshall Space Flight Center http://technology.nasa.gov/scripts/nls_ax.dll/w3SuccItem(2200531

[3] US Department of Energy (DOE), Energy Information Administration, *A Look At Residential Energy Consumption in 1997*, DOE/EIA-0632 (97) pg. 9,22,15 http://www.eia.doe.gov/pub/pdf/consumption/063297.pdf

[4] "Refrigerator Monitoring, A Sequel", Larry Kinney, *Home Energy Magazine Online*, September/October 2000, http://www.homeenergy.org/magazine.html, http://hem.dis.anl.gov/eehem/00/000916.html

[5] K. Stahlkopf, M. Wilhelm, "Tighter Controls for Busier Systems", IEEE Spectrum, April 1997, p. 49-52

[6] "Transient Stability Enhancement Using Static Var Compensator", J. Graham, West Virginia University Thesis, 378.7543 Eng'g.G76tc.2, October 1995, p. 7

[7] L. Gyugyi, R. Otto, T. Putnam, "Principles and Applications of Static Thyristor Controlled Shunt Compensators", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-97, No. 5, October 1978, p. 1935-1944

[8] Reactive Power Control in Electric Systems, Edited by T. Miller, General Electric Company Corporate Research and Development Center, John Wiley & Sons, NY: 1982 p. 187

[9] M. El-Sharkawi, T. Williams, N. Butler, "An Adaptive Power Factor Controller for Three-Phase Induction Generators", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No. 7, July 1985, p. 1825-1831

[10] M. El-Sharkawi, N. Butler, R. Yinger, "Development and Field Testing of an Adaptive Power Factor Controller", IEEE Transactions on Energy Conversion, Vol. EC-2, No. 4, Dec. 1987

[11] S. Paul, S. Basu, R. Mondal, "A Microcomputer Controlled Static Var Compensator for Power Systems Laboratory Experiments", IEEE Transactions on Power Systems, Vol. 7, No.1, Feb. 1992 p. 371-376

[12] A. Exposito, F. Vazquez, C. Mitchell, "Microprocessor – Based Control of an SVC for Optimal Load Compensation", IEEE Transactions on Power Delivery, Vol. 7, No. 2, April 1992, p. 706-712

[13] T. Hiyama, W. Hubbi, T. Ortmeyer, "Fuzzy Logic Control Scheme with Variable Gain for Static Var Compensator to Enhance Power System Stability", IEEE Transactions on Power Systems, Vol. 14, No. 1, Feb. 1999 p. 186-191

[14] R. Dugan, M. McGranaghan, and H. Beaty, *Electrical Power Systems Quality*, McGraw-Hill, NY: 1996 pg.205

[15] PIC16F87X 28/40 Pin 8-Bit CMOS Flash MicroControllers Data Sheet, Online Datasheet provided by Microchip Corp. http://www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf

[16] Texas Instruments Universal Active Filter (UAF42) Online Documentation http://focus.ti.com/docs/prod/folders/print/uaf42.html

[17] Application Report Abstract: Filter Design Program For The UAF42 Universal Active Filter, Texas Instruments, http://www.ti.com/sc/docs/psheets/abstract/apps/sbfa002.htm

[18] R. Inamine, "Logic Levels for CMOS", Honolulu Community College, http://www.hcc.hawaii.edu/~richardi/113/c113_2d/2d_4/2d4.htm

[19] Microchip Corp. Application Note 40171a: PICREF-4, *PICDIM Lamp Dimmer for the PIC12C508*: 1997 pg. 3-4 http://www.microchip.com/download/appnote/rdesigns/40171a.pdf

[20] MOC3051 Triac Driver Data Sheet, Fairchild Semiconductor, p. 1 http://www.fairchildsemi.com/ds/MO/MOC3051-M.pdf

[21] MOC3051 Triac Driver Data Sheet, Fairchild Semiconductor, p. 6 http://www.fairchildsemi.com/ds/MO/MOC3051-M.pdf

[22] MOC3051 Triac Driver Data Sheet, Fairchild Semiconductor, p. 3 http://www.fairchildsemi.com/ds/MO/MOC3051-M.pdf

[23] MOC3051 Triac Driver Data Sheet, Fairchild Semiconductor, p. 2 http://www.fairchildsemi.com/ds/MO/MOC3051-M.pdf

[24] Alternistor Triac Data Sheet, Teccor Electronics
http://www.teccor.com/thyristor/altquad.pdf

[25] J. Yen, R. Langari, *Fuzzy Logic*: *Intelligence, Control, and Information*, Prentice-Hall, NJ: 1999 pg. 209-214

[26] PIC Compiler: PICC, Developed by CCSInfo, PICC Users Manual,
http://www.ccsinfo.com/piccmanual3.zip
http://www.ccsinfo.com/

[27] MPLAB PIC Programmer and Debugger IDE v. 5.6, Microchip Corp.,
http://www.microchip.com/1010/pline/tools/picmicro/devenv/mplabi/

[28] T.J.E Miller, Ed. , General Electric Company, *Reactive Power Control in Electric System*, John Wiley & Sons, NY: 1982 pg. 187, 207

[29] PZ4000 Power Analyzer, Manufactured by Yokogawa,
http://www.yokogawa.com/tm/Bu/PZ4000/

[30] R. Dugan, M. McGranaghan, and H. Beaty, *Electrical Power Systems Quality*, McGraw-Hill, NY: 1996 pg.129

[31] R. Dugan, M. McGranaghan, and H. Beaty, *Electrical Power Systems Quality*, McGraw-Hill, NY: 1996 pg.206

[32] IEEE Standard 519-1992: *Recommended Practice and Requirements for Harmonic Control in Electric Power Systems, 1992*

[33] W. Mack Grady, *Harmonics and How They Relate to Power Factor*, Proceedings of the EPRI Power Quality Issues and Opportunities Conference (PQA '93), San Diego, CA., Nov. 1993

[34] Brzezinski, B., "Inductive Load Current Harmonic Injection", *Proc. Southeastern Symp. on System Theory,* Huntsville, AL, March 18-19, 2002

[35] "Power Factor and Harmonics", Online *Robicon* Technical Document,
http://www.robicon.com/library/acrobat/pfactor.pdf

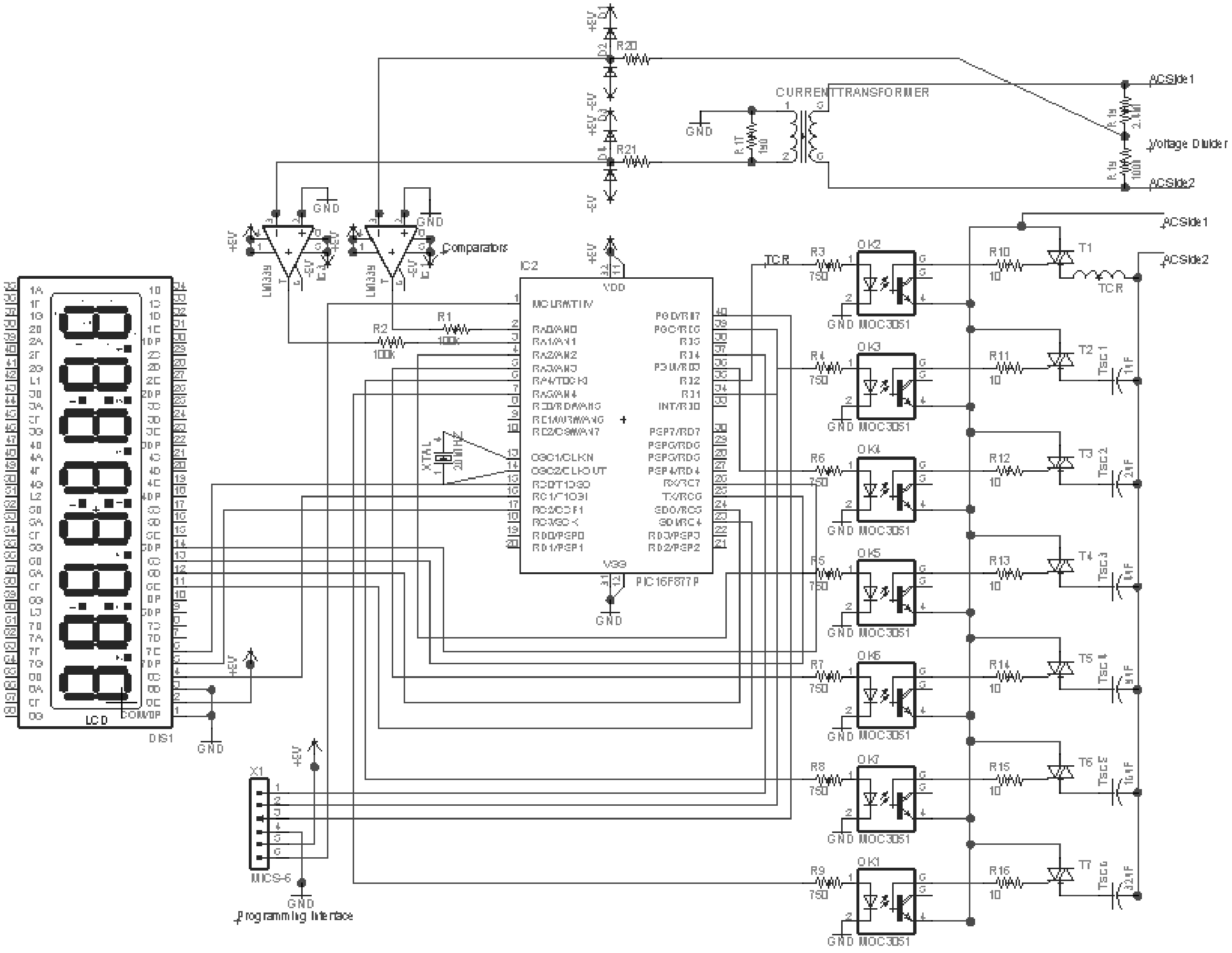
[36] Arkansas Public Service Commission, Entergy-Arkansas, *Standby Service Rider (SS) – Part III. Rate Schedule Number: 20*, 03/09/2001, p. 4,
http://www.entergy-arkansas.com/tariffs/20ss.pdf

[37] DIXIE ESCALANTE RURAL ELECTRIC ASSOCIATION, BERYL, UTAH 84714 Effective Rates: April 1998 http://www.cc.state.az.us/utility/tariff/dixie.pdf

[38] Summary of Rate Schedules - St. George, Utah, Municipal Power and Light System Effective Rates October 1, 2001, http://www.sgcity.org/utilities/rates.asp

[39] Nicholas Abi-Samra, "Benefits of Power Factor Correction Capacitors", Fellow Engineer, Westinghouse Electric Corporation http://www.geocities.com/SiliconValley/Pines/3406/paper1.htm

APPENDIX A – PSVC ELECTRICAL SCHEMATIC



APPENDIX B – PSVC SOURCE CODE

```
/* PSVC Software */
/* Scott Zemerick */
/* MSEE Thesis */
/* Version 2 */

#include "c:\picc\examples\16f877.h"
#use delay (clock=20000000)
#fuses HS, NOWDT, NOPROTECT, NOPUT, NOBROWNOUT
#include "lcd.h"

#use standard_io(a)
#use standard_io(b)
#zero_ram

#define ON 1
#define OFF 0
#define NUM_CAP_BANKS 3
#define TCR_LIMIT 10
#define LEADING_LIMIT -28
#define TURN_ON -1

// Unsigned INT (8 bit):   0   -> +256
// Signed INT (8 bit):   -128 -> +128
// CCS Compiler makes INTs unsigned by default


//-------------
//Fuzzy System
//-------------
//Input Membership Function
//                                -20  -15  -10   -5   0    5   10  15  20  25
signed int PhaseAngle_Input[10] = {40,  45,  50,  55,  60,  65, 70, 75, 80, 85};


//Output Membership Function
//                                           91  95  100  112  115  122  124  126 128 135
unsigned int FiringAngleChange_Output[10] = {1,  5,  10,  22,  25,  32,  34,  36, 38,
45};

//Fuzzy TSK Rules
unsigned int Rules[10] = {0, 1, 2, 3, 4, 5, 5, 7, 8, 9};

// One Input - One Output Fuzzy Function Prototype
unsigned int fuzzy_tsk(float in);

//General Prototypes
void turn_capacitors_on();
void turn_capacitors_off();
void turn_inductor_on();
void turn_inductor_off();
signed long int round(float x);
void synchronize_on_rising_edge();

//Globals - I don't like them, but necessary
signed long ticks = 0;
short int look_for_voltage = 1;
float phase_angle = 0.0;
unsigned long int firing_angle = 90;
short int voltage_state = 0;
short int current_state = 0;
short int bLagging = 0;
short int voltage_initial_state = 0;
short int current_initial_state = 0;
short int enable_firing_interrupt = 0;
float time_delay = -1.0;
int counter_set = 0;
signed long int interrupt_counter = 0;


/* -----------------------------------------------*/
/* Perform Capacitor Bank TSC Initialization Here */
/* -----------------------------------------------*/
int capacitor_banks[NUM_CAP_BANKS];

/* TSC Capacitor Bank Prototypes */
void set_capacitor_banks();
void control_capacitor_banks(int bank, int mode);
void turn_on_all_capacitor_banks();
void turn_off_all_capacitor_banks();
void control_capacitor_banks(int bank, int mode);
int find_number_of_cap_banks_on();

/* ------------- */
/* TCR Interrupt */
/* ------------- */
#INT_RTCC
inductor_firing_isr()
{
    if(enable_firing_interrupt == 1)
    {
          output_high(PIN_B2);
          delay_us(20);
          output_low(PIN_B2);
          set_timer0(0);
          enable_firing_interrupt = 0;
    }
}


/* ------------- */
/* LCD Interrupt */
/* ------------- */
#INT_TIMER2
timer_2_isr()
{
    interrupt_counter++;
    if(interrupt_counter >= 100)
    {
       interrupt_counter=0;
       printf(LCD_putchar, "\f%3.2f",phase_angle);

       lcd_gotoxy(0,1);
       printf(LCD_putchar, "FA: %ld", firing_angle);

       lcd_gotoxy(0,2);
       printf(LCD_putchar, "Cap: %d", find_number_of_cap_banks_on());

       //Synchronize Voltage Since Cycles Missed
       synchronize_on_rising_edge();
    }
}
```

```
/* ---------------*/
/* Main PSVC Code */
/* ---------------*/
void main()
{
    signed long int capacitor_counter = TURN_ON;

    //Set the LCD
    lcd_init();

    //Set Up Timer 0
    set_rtcc(0);
    setup_counters(RTCC_INTERNAL, RTCC_DIV_128);
    enable_interrupts(INT_RTCC);

    //Set Up Timer 2
    set_timer2(0);
    setup_timer_2(T2_DIV_BY_16, 0xFF, 16);
    enable_interrupts(INT_TIMER2);

    //Turn On All Interrupts
    enable_interrupts(GLOBAL);

    //Set up timer 1
    set_timer1(0);
    setup_timer_1( T1_INTERNAL | T1_DIV_BY_4);
    setup_timer_1( T1_INTERNAL | T1_DIV_BY_8);

    //Initialize Capacitor Banks to Off
    turn_off_all_capacitor_banks();

    //Synchronize Voltage
    synchronize_on_rising_edge();

    //-----------------------------
    //Main While Loop Controls PSVC
    //-----------------------------
    while(ON)
    {

        //Turn Off the LCD
        disable_interrupts(INT_TIMER2);

        //Update Capacitor Banks Once Every 5 Seconds
        if(capacitor_counter > 300 || capacitor_counter == TURN_ON)
        {
               //Reset Capacitor Counter
               capacitor_counter = 0;

               //Set Capacitor Banks Based On Previous phase_angle
               set_capacitor_banks();

               //Synchronize Voltage
               synchronize_on_rising_edge();
        }

        //Increment Capacitor Counter
        capacitor_counter++;
```

```
//-------------------------
//Continue Controlling PSVC
//-------------------------
//Determine Leading/Lagging
voltage_state = input(PIN_A0);
voltage_initial_state = voltage_state;
while(voltage_initial_state == voltage_state)
{
        voltage_state = input(PIN_A0);
}

//Start Inductor Countdown as Voltage Crosses Zero
if(time_delay != -1.0)
{
        set_timer0(counter_set);
        enable_firing_interrupt = 1;
}

//Take Current Reading
current_state = input(PIN_A1);

//Going from L->H
if((voltage_initial_state == 0) && (voltage_state == 1))
{
        if(current_state == 0)
        {
                bLagging = 1;
        }
        else
        {
                bLagging = 0;
        }
}


//Going from H->L
else if((voltage_initial_state == 1) && (voltage_state == 0))
{
        if(current_state == 0)
        {
                bLagging = 0;
        }

        else
        {
                bLagging = 1;
        }
}


if(bLagging == 1)
{
        //Read Voltage First
        //Wait for Voltage Zero Crossing
        voltage_state = input(PIN_A0);
        while(voltage_state == input(PIN_A0))
        {
        }

        //Start Inductor Countdown as Voltage Crosses Zero
        if(time_delay != -1.0)
        {
                set_timer0(counter_set);
                enable_firing_interrupt = 1;
        }
```

```
                set_timer1(0);


                //Read Current
                //Wait for Current Zero Crossing
                current_state = input(PIN_A1);
                while(current_state == input(PIN_A1))
                {
                }

                ticks = get_timer1();
        }

        else
        {
                //Read Current First
                //Wait for Current Zero Crossing
                current_state = input(PIN_A1);
                while(current_state == input(PIN_A1))
                {
                }

                //Set the Timer
                set_timer1(0);

                //Read Voltage
                //Wait for Voltage Zero Crossing
                voltage_state = input(PIN_A0);
                while(voltage_state == input(PIN_A0))
                {
                }

                //Start Inductor Countdown as Voltage Crosses Zero
                if(time_delay != -1.0)
                {
                        set_timer0(counter_set);
                        enable_firing_interrupt = 1;
                }

                ticks = get_timer1();
                ticks = -ticks;
        }


        //Calculate Phase Angle from Ticks Counted
        phase_angle = ( ((float)ticks * (float)0.0000016) / (1.0 / 60.0)) * 360.0;

        //Account for Rollover
        if((phase_angle <= -170.0) && (phase_angle >= -180.0 ))
        {
                phase_angle = -(180.0 + phase_angle);
        }

        //Weed out False Signals
        if((phase_angle > -50.0) && (phase_angle < 130.0))
        {

                //Ensure Lower Bound is Met
                if(phase_angle <= -20)
                {
                        phase_angle = -19;
                }
```

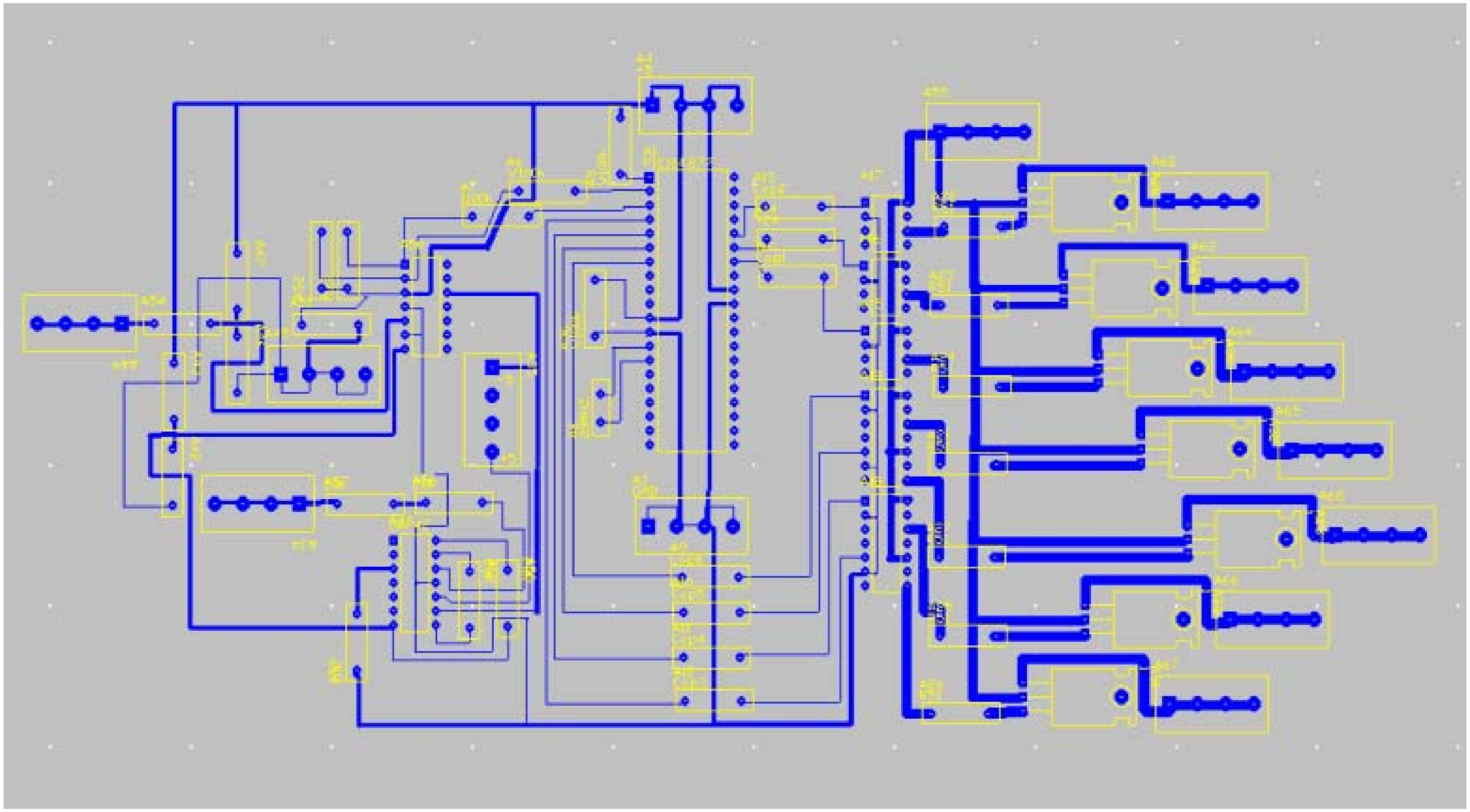
```
            if( phase_angle > -55 && phase_angle < 55 )
            {

                    //Invoke Fuzzy System
                    //Fuzzy System Returns Updated Firing Angle
                    firing_angle = fuzzy_tsk(phase_angle);

                    //Convert Desired Firing Angle to a Delay Time
                    time_delay = (((float)firing_angle / 360.0) * (1.0 / 60.0));
                    counter_set = round(time_delay / 0.0000256);
                    counter_set = 255 - counter_set;
            }

            //Show the Phase on LCD
            enable_interrupts(INT_TIMER2);
        }
 }
}




//------------------------------
//Set Number of Capacitor Banks
//------------------------------
void set_capacitor_banks()
{
    int i=0;

    //Check if Phase is Lagging - Only Add More Capacitors
    //If Phase Is Lagging
    if (phase_angle > 0)
    {
       //Only Add More Capacitors If Phase > TCR_LIMIT Degrees
       //10 Degrees Is the Current Span of the TCR Branch
       if (phase_angle > TCR_LIMIT)
       {
               //Add the Next Available Capacitor Bank
               for(i=0; i<NUM_CAP_BANKS; i++)
               {
                       if (capacitor_banks[i] == OFF)
                       {
                               //Turn On Capacitor Bank
                               control_capacitor_banks(i,ON);
                               break;
                       }
               }
       }
    }


    //Remove a Capacitor Bank If Leading Too Much
    else if (phase_angle < LEADING_LIMIT)
    {
       for(i=NUM_CAP_BANKS-1; i>=0; i--)
       {
               if (capacitor_banks[i] == ON)
               {
                       //Turn Off Capacitor Bank
                       control_capacitor_banks(i,OFF);
                       break;
               }
       }
   }
}
```

```
//----------------------------
//Turn ON All Capacitor Banks
//----------------------------
void turn_on_all_capacitor_banks()
{
    int i=0;
    for(i=0; i<NUM_CAP_BANKS; i++)
    {
       control_capacitor_banks(i,ON);
    }
}

//----------------------------
//Turn OFF All Capacitor Banks
//----------------------------
void turn_off_all_capacitor_banks()
{
   int i=0;
   for(i=0; i<NUM_CAP_BANKS; i++)
   {
       control_capacitor_banks(i,OFF);
   }
}

//-----------------------------------------
//Returns the Number Of Capacitor Banks On
//-----------------------------------------
int find_number_of_cap_banks_on()
{
   int i=0;
   int banks_turned_on = 0;
   for(i=0; i<NUM_CAP_BANKS; i++)
   {
       if (capacitor_banks[i] == ON)
       {
           banks_turned_on++;
       }
   }

   return banks_turned_on;
}

//----------------------------
//Control Capacitor Banks
//Individual PIC Pins Needed
//----------------------------
void control_capacitor_banks(int bank, int mode)
{

       //Wait for Current to Cross Zero Before Switching Capacitor
       current_state = input(PIN_A1);
       current_initial_state = current_state;
       while(current_initial_state == current_state)
       {
               current_state = input(PIN_A1);
       }


       //1st Capacitor Bank
       if (bank == 0)
       {
               if (mode == OFF)
               {
                   output_low(PIN_B1);
                   capacitor_banks[bank] = OFF;
               }
               else
               {
                   output_high(PIN_B1);
                   capacitor_banks[bank] = ON;
               }
        }


        //2nd Capacitor Bank
        else if (bank == 1)
        {
               if (mode == OFF)
               {
                   output_low(PIN_B3);
                   capacitor_banks[bank] = OFF;
               }
               else
               {
                   output_high(PIN_B3);
                   capacitor_banks[bank] = ON;
               }
        }

        //3rd Capacitor Bank
        else if (bank == 2)
        {
               if (mode == OFF)
               {
                   output_low(PIN_A2);
                   capacitor_banks[bank] = OFF;
               }
               else
               {
                   output_high(PIN_A2);
                   capacitor_banks[bank] = ON;
               }
        }
}


//-----------------------------------------
//Synchronize on Rising Edge of Voltage
//Set Capacitor Banks
//----------------------------------------
void synchronize_on_rising_edge()
{
    //Determine Leading/Lagging
    voltage_state = input(PIN_A0);
    voltage_initial_state = voltage_state;

    //Wait for Voltage to Change State
    while(voltage_initial_state == voltage_state)
    {
          voltage_state = input(PIN_A0);
    }

    //Take Current Reading
    current_state = input(PIN_A1);

    //Going from L->H
    if((voltage_initial_state == 0) && (voltage_state == 1))
    {
        if(current_state == 0)
        {
               bLagging = 1;
        }
        else
        {
               bLagging = 0;
        }
    }


  //Going from H->L
  else if((voltage_initial_state == 1) && (voltage_state == 0))
  {
        if(current_state == 0)
        {
               bLagging = 0;
        }
        else
        {
               bLagging = 1;
        }
  }


//Wait for Rising Edge of Voltage
 if(bLagging == 1)
 {
        if(input(PIN_A0) == 0)
        {
               while(input(PIN_A0) == 0)
               {
               //Just Wait for Voltage to Go from H->L
               }
        }
 }

 else
 {
        if(input(PIN_A1) == 0)
        {
               while(input(PIN_A1) == 0)
               {
               //Just Wait for Current to Go from H->L
               }
        }
 }
}


//-----------------------------
//Rounding Function on a Float
//-----------------------------
signed long int round(float x)
{
    //float decimal = 0.0;  //Commented to Save RAM
    int return_rounded = 0;

    if((x - ((signed long int)x)) < 0.5)
    {
        return_rounded = ((signed long int)x);      //This is the floor function
    }
    else
    {
       return_rounded = ((signed long int)(x+1));   //This is the ceil function
    }

    return(return_rounded);
}
```

```
//-----------------------------------
//One input-output TSK Fuzzy Engine
//----------------------------------
unsigned int fuzzy_tsk(float crisp_in)
{
   float fired[2] = {0,0};
   int rules[2] = {0,0};
   unsigned int out = 0;
   signed int a = 0;
   signed int b = 0;
   signed int c = 0;
   signed int d = 0;
   signed int e = 0;
   signed int f = 0;
   signed int g = 0;
   signed int h = 0;
   signed int i = 0;
   signed int j = 0;

   a = PhaseAngle_Input[0];   // -40
   b = PhaseAngle_Input[1];   // -30
   c = PhaseAngle_Input[2];   // -12
   d = PhaseAngle_Input[3];   // -5
   e = PhaseAngle_Input[4];   // 0
   f = PhaseAngle_Input[5];   // 5
   g = PhaseAngle_Input[6];   // 12
   h = PhaseAngle_Input[7];   // 30
   i = PhaseAngle_Input[8];   // 40
   j = PhaseAngle_Input[9];

   //Scale to Fit Fuzzy Rules
   crisp_in += 60.0;

   //----------------
   //Start Fuzzy TSK
   //----------------
   if ((crisp_in >= a) && (crisp_in < b))
   {
         fired[0] = (-1.0) * ( crisp_in / (float)a ) + 1.0;
         rules[0] = 0;
         rules[1] = 1;
   }

   else if ((crisp_in >= b) && (crisp_in < c))
   {
         fired[0] = ((-1.0 )* crisp_in + (float)c) /((float)(c-b));
         rules[0] = 1;
         rules[1] = 2;
   }

   else if ((crisp_in >= c) && (crisp_in < d))
   {
         fired[0] = ((-1.0)* in + (float)d) / ((float)(d-c));
         rules[0] = 2;
         rules[1] = 3;
   }

   else if ((crisp_in >= d) && (crisp_in < e))
   {
         fired[0] = ((-1.0)* crisp_in+ (float)e) / ((float)(e-d));
         rules[0] = 3;
         rules[1] = 4;
   }

   else if ((crisp_in >= e) && (crisp_in < f))
   {
         fired[0] = ((-1.0)* crisp_in + (float)f) / ((float)(f-e));
```

```
          rules[0] = 4;
          rules[1] = 5;
   }

   else if ((crisp_in >= f) && (crisp_in < g))
   {
          fired[0] = ((-1.0)*crisp_in+ (float)g) / ((float)(g-f));
          rules[0] = 5;
          rules[1] = 6;
   }

   else if ((crisp_in >= g) && (crisp_in < h))
   {
          fired[0] = ((-1.0)*crisp_in+(float)h) / ((float)(h-g));
          rules[0] = 6;
          rules[1] = 7;
   }

   else
   {
       rules[0] = 7;
       rules[1] = 8;
   }


   fired[1] = 1 - fired[0];
   out = (unsigned int)(FiringAngleChange_Output[Rules[rules[0]]]*fired[0] +
FiringAngleChange_Output[Rules[rules[1]]]*fired[1]);

   //Scale Firing Angle from Fuzzy Rules
   out += 90;
   return out;
}
```

APPENDIX C – PRINTED CIRCUIT BOARD DESIGN



APPENDIX D – PSVC COMPONENT LIST

SOCKETS/CONNECTORS	QUANTITY
40 PIN DIP	1
14 PIN DIP	4
08 PIN DIP	3
POWER CONNECTORS	13

SEMICONDUCTORS	QUANTITY
PIC16F877	1
UAF42 ACTIVE FILTER	1
LM336 QUAD COMPARATOR	1
MOC3051 TRIAC DRIVER	7
POWER TECCOR ALTERNISTOR	7
CRYSTAL – 20MHZ	1
CURRENT TRANSFORMER	1
LIQUID CRYSTAL DISPLAY	1

STATIC COMPENSATORS	QUANTITY
AC CAPACITORS RATED < 250 VAC	6 (for binary ratios)
166MH INDUCTOR RATED < 250 VAC	1

RESISTORS (OHMS)/DIODES

10 (Q=7), 51 (Q=7), 180, 2.7K, 3K, 5.1K (Q=2), 15K, 22K, 43K, 100K (Q=3), 2.7M (Q=3), 1N4007 DIODES (Q=4)

APPENDIX E – INDIVIDUAL CAPACITOR DATA

Farads (uF)	Manufacturer	Max. Voltage	Max. VARS	Price (Q=1)	Price (Q=1000)
7.5	Panasonic - ATG	180	**92**	$3.38	$1.80
10	Panasonic - ATG	180	**122**	$4.03	$2.14
12	Panasonic - ATG	180	**147**	$4.50	$2.39
10	Panasonic - ATG	210	**166**	$4.15	$2.21
15	Panasonic - ATG	180	**183**	$5.03	$2.67
10	Panasonic - ATG	230	**199**	$4.88	$2.59
12	Panasonic - ATG	210	**200**	$4.65	$2.47
10	Panasonic - ATG	250	**236**	$6.68	$3.55
12	Panasonic - ATG	230	**239**	$5.38	$2.86
20	Panasonic - ATG	180	**244**	$6.43	$3.42
15	Panasonic - ATG	210	**249**	$5.58	$2.97
12	Panasonic - ATG	250	**283**	$7.60	$4.04
15	Panasonic - ATG	230	**299**	$6.95	$3.70
15	GE (Power Supply)	240	**326**	$21.17	$6.14
15	Panasonic - ATG	250	**353**	$8.98	$4.77
25	GE (Power Supply)	240	**543**	$26.75	$7.76
125	Mallory (Motor Start)	110	**570**	$7.61	$2.82
30	GE (Power Supply)	240	**651**	$29.54	$8.57
35	GE (Power Supply)	240	**760**	$32.21	$9.34
189	Mallory (Motor Start)	110	**862**	$8.42	$3.13
233	Mallory (Motor Start)	110	**1063**	$9.50	$3.52
64	Mallory (Motor Start)	250	**1508**	$12.00	$4.45
88	Mallory (Motor Start)	250	**2073**	$13.63	$5.05
100	GE (Power Supply)	240	**2171**	$71.99	$20.88
120	GE (Power Supply)	240	**2606**	$82.21	$23.84
Average				**$15.73**	**$5.48**

VITA

Education

M.S. Electrical Engineering, May 2002
Thesis: Design of a Prototype Personal Static Var Compensator
West Virginia University

B.S. Computer Engineering, May 2000
West Virginia University

Oak Hill High School, June 1996

Honors/Awards

Completed Graduate School with 4.0/4.0 GPA, May 2002
West Virginia University

Graduated with Honors – Magna Cum Laude, May 2000
West Virginia University

High School Valedictorian, 1996

Publications

Southeastern Symposium on System Theory (SSST) Conference
S. Zemerick, P. Klinkhachorn, A. Feliachi, "Design and Implementation of a "Personal" Static Var Compensator," *Proc. Southeastern Symp. on System Theory,* Huntsville, AL, March 18-19, 2002.

Power Engineering Society (PES) Summer Conference
S. Zemerick, P. Klinkhachorn, A. Feliachi, "Design of a microprocessor – controlled personal static var compensator (PSVC)," *Proceedings of the IEEE Summer PES Meeting*, Chicago, Ill., 21-25 July 2002

Professional Society Memberships

Member of IEEE Since 1997