Introduction

Power quality is becoming an important issue nowadays in domestic and industrial fields. Even in small industries sensitive loads like personal computers, servers, UPS, and distortions due to power electronic devices, adjustable speed drives, microprocessors, logic controllers, switched mode power supplies and Energy efficient lightings cause severe problems to other equipment present in the industry, hence monitoring of power quality is more essential.

Electric energy meters are devices that are often installed in buildings and businesses in order to measure the amount of consumed electric energy (i.e., these meters are installed for billing purposes) However, the increasing awareness about energy consumption is not the only concern today. The quality of the supplied energy is also an important feature, so it is necessary to introduce new technologies that provide end-users with up-to-date, online, real-time information about the quantity and quality of the power supply they receive from the utility.

This project focuses on certain quality factors of single-phase power like active power, reactive power, power factor, phase difference and the RMS values of voltage and current.

Work done

Electric energy consumption consists of three primary variables as follows:

$$P = V_{rms} \times I_{rms} \times \cos\emptyset \ \ (Watt)$$

$$S = V_{rms} \times I_{\ rms} \ \ (VA)$$

$$Q = V_{rms} \times I_{rms} \times \sin\emptyset \ \ (VAR)$$

P is the actual power required by the load in (Watt). S is the apparent electrical power supplied by the primary generator to the consumer in Volt-Ampere (VA). Q is the power produced by the load which is inductive in Volt-Ampere Reactive (VAR).

Power factor is the ratio of true power to apparent power and can be calculated by finding the cosine of the phase angle between the supply voltage and current (see attached image from Google). This is significant in AC applications as "Apparent Power" (Volt-Amperes) can be easily calculated using Voltage multiplied by Current. However, to get the real power or "True Power" (Watts) apparent power must be multiplied by the power factor to make a true measurement of power in Watts. This only applies to loads which have a significant inductive or capacitive component (such as a motor). Purely resistive loads such as electric heaters or incandescent bulbs have a power factor of 1.0 (unity) and therefore True Power and Apparent Power are the same.

$$pf = \frac{Real\ Power}{Apparent\ Power}$$

$$\therefore pf = \cos \emptyset$$

All these parameters can be calculated using instantaneous values of current and voltage. Averaging the product of current and voltage over a time period gives us the value of real power.

The average value obtained was the average power (P) at the particular instant. During the same sampling period, the value of the current and voltage obtained were squared and averaged. The value was then square rooted to obtain the root mean square (rms) value. Both the rms value of current and voltage were multiplied to obtain the apparent power (S). In order to obtain the value of power factor (p.f) of the circuit, the average power (P) was divided by the apparent power (S).

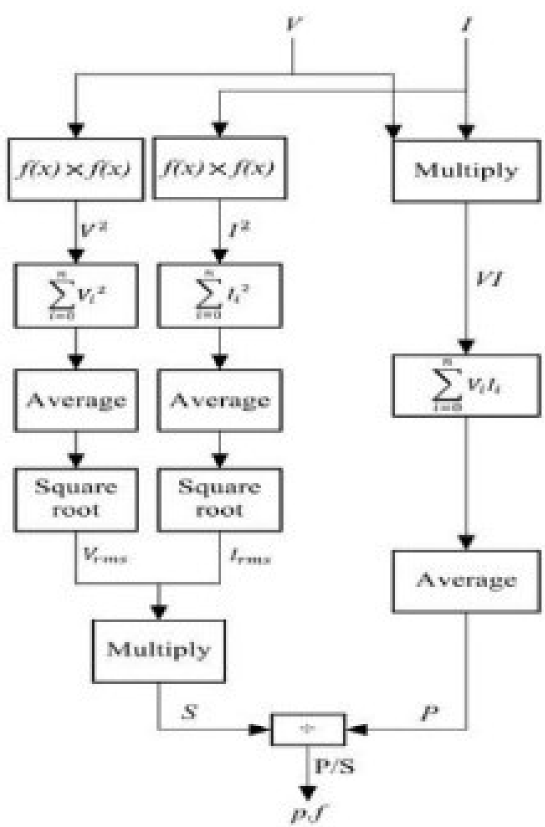


Fig: Flow chart of the required process

All steps from the flow chart were replicated in Simulink using the fundamental blocks available and the power quality factors of a single-phase voltage source along with a RLC load were calculated. For the sampling period of time interval of 1 ms, the instantaneous current and voltage were multiplied to obtain the instantaneous power. The total number of samples of instantaneous power obtained was then average by the total number of samples collected in 1 ms. The values were verified against the ready-made blocks and both models yielded similar results.

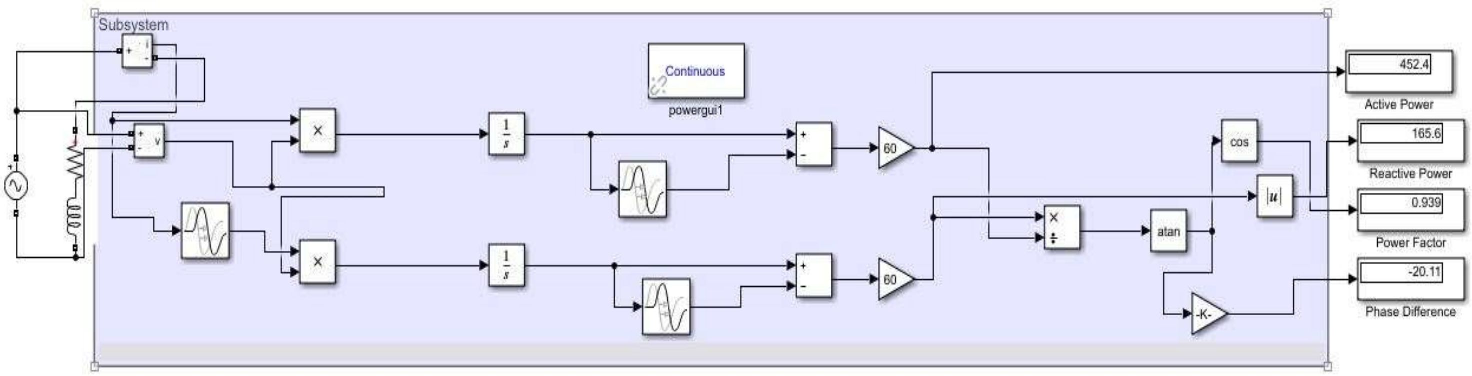


Fig: Screenshot of the Simulink model

Similar method was followed in Arduino IDE. Due to lack of real inputs, sinusoidal voltage and current waves with a phase difference were created for simulation purposes. **Additionally, a feature has been provided to increment or decrement the voltage amplitude by 5 units so that the program can be tested in real time for fluctuating voltages.** The following is the Arduino code for the same:

```
#include <math.h>

int A=240;

void setup() {

Serial.begin(2000000); // baud rate

}


void loop() {

  float f=60; // frequency of source

  float fs=1000; // sampling frequency

  float w=2*PI*f; // angular frequency
```

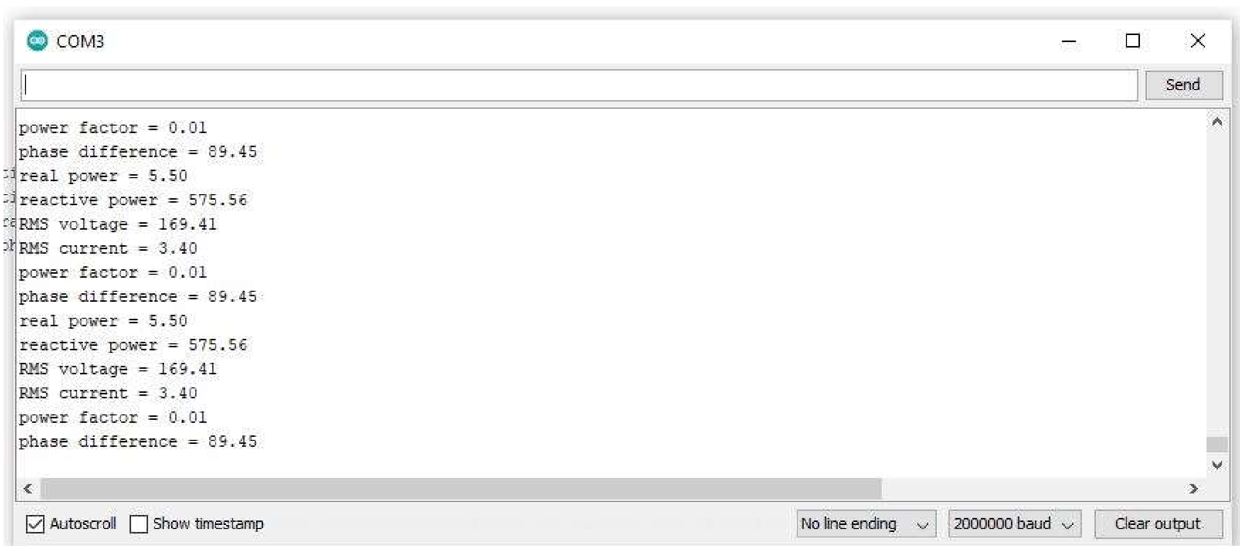
```
float pow_sum=0;
float prev_V=0;
float prev_I=0;
float prev_t=0;
float I_sum=0;
float V_sum=0;
float max_V=0;
float max_I=0;
float max_Vt=0;
float max_It=0;
if(Serial.available()>0){
    int input=Serial.read(); //to increment or decrement voltage amplitude for testing in real time
   if(input=='h') A+=5; // press 'h' in serial monitor to increment amplitude by 5 units
   if(input=='l') A-=5; //  press 'l' in serial monitor to decrement amplitude by 5 units
  }
for(float t=0; t<=1/f; t+=1/fs){
  float V=240*sin(w*t); // sinuosoidal voltage
  float I=5*sin(w*t); // sinuosoidal current
  V_sum+=(t-prev_t)*0.5*(V*V+prev_V*prev_V); // Integrating voltage for RMS
  I_sum+=(t-prev_t)*0.5*(I*I+prev_I*prev_I); // Integrating current for RMS
  pow_sum+=(t-prev_t)*0.5*(V*I+prev_V*prev_I); // Integrating product of current and voltage
  if(max_V<V){  // for finding if positive or negative phase difference
    max_V=V;
    max_Vt=t;
   }
  if(max_I<I){
    max_V=V;
    max_It=t;
```

```
    }
   prev_V=V; // current becomes previous for next cycle
   prev_I=I;
   prev_t=t;
  }
  float real_pow=pow_sum*f;
  float RMS_V=sqrt(V_sum*f);
  float RMS_I=sqrt(I_sum*f);
  float pow_fac=real_pow/(RMS_V*RMS_I);
  float phase=acos(pow_fac)*180/PI;
  float react_pow=(RMS_V*RMS_I)*sin(acos(pow_fac));
  if(max_Vt>max_It) phase*=-1;
  Serial.print("real power = ");
  Serial.println(real_pow);
  Serial.print("reactive power = ");
  Serial.println(react_pow);
  Serial.print("RMS voltage = ");
  Serial.println(RMS_V);
  Serial.print("RMS current = ");
  Serial.println(RMS_I);
  Serial.print("power factor = ");
  Serial.println(pow_fac);
  Serial.print("phase difference = ");
  Serial.println(phase);
  delay(10); // delay so that output is visible
}
```

Results

Calculations were done using a calculator to verify the code and it was verified that the model gives fairly accurate results for all the quality parameters. The results were mostly verified by using simple vales for phase shift like 0°, 30°, 45°, 60°, 90° etc. This concludes that the calculation technique is able to compute the accurate value for the power quality factor measurement. However, the slight inaccuracy of the measurement is caused by the rounding off errors due to the limitations of different data types in C++ language.

Above is the screenshot for in phase voltage and current and it can be verified that power factor is 1.



Above is the screenshot of a purely capacitive load and power factor and phase difference can be verified.

Above is the screenshot of a purely inductive load and power factor and phase difference can be verified.

Above is the screenshot of an RL load. This can be verified by checking the values of power factor and phase difference

Therefore, it is verified that the code runs well for all types of linear loads and also fluctuating values of voltage and current.