LABVIEW Implementation of Central Controller for Active Power Load Balancing

J. W. Hagge, *Senior Member, IEEE*, and L. L. Grigsby, *Fellow, IEEE*

***Abstract* — Increased use of renewable energy generators requires effective power management because renewable energy is variable by nature. Sometimes, a small grid or a microgrid is able to function independently of the broader power grid. Power and load required have to be predicted based on a variety of constantly changing factors. This task is referred to as Active Power Load Balancing. This is performed by a Central Controller which takes in power and load data about the power system and combines it with continuing factors and updates to manage the power. The goal of this project is to implement the Central Controller in LABVIEW from Simulink so that the program can interact with the hardware as LABVIEW has a broader hardware interface.**

***Index Terms* – Electrical Engineering, load flow, load management, Photovoltaic cells, Power control, Power engineering, Power engineering-computing, Power system control.**

I. Introduction

There has been an increase in the use of distributed energy generators in recent years as a result of both a desire to lower gas pollutants in the atmosphere and liberalization electricity market. Because renewable energy production is based on natural phenomenon such as the amount of solar irradiance or water supply available which are more variable than a set amount of fuel in a coal plant. The variance in the amount of power available results in output power fluctuations. In order to compensate for output power fluctuations, the generator has to vary its angular velocity and as a result its frequency. Therefore, variation in output power results in a variation in grid frequency which may adversely impact the device performance of dependent electronics.

In order to address this problem, a range of storage systems can be designed so that power is available when needed. One way in which this can be done is to introduce a demand response capability to home electronics. Demand response capability measures the amount of electricity being used by a given device and accordingly adjusts power production and allocation. So, from the perspective of the power generator certain devices can function as both energy producers and consumers or energy ‘prosumers’.

Shifting focus to long term energy management, renewable energy sources are difficult to predict. For instance, in wind power management wind speed could change drastically over a relatively short period of time. In a solar plant, there no solar irradiance over night or on a cloudy day, and the amount of solar irradiance available can also change drastically within a short period of time. The demand for power also varies over the day and usually peaks around midday.

There are two aspects to effective power management: forecasting the available power and forecasting the load at any given point in time. These are done based on the same data for the previous day. Models of forecasting are complex as both the availability and the demand for power depend on a variety of changing factors which are constantly changing over time: weather, economic activity, time of day, etc. The modeling algorithms are often based on historical consumption and weather data.

The task of predicting and planning and balancing available power with future power consumption is referred to as active power load balancing. The goal of active power load balancing is to ensure that the power difference between demand and supply never exceeds the battery power when the micro grid is islanded. Put mathematically:

$$|P_{PV} - P_{load}| \leq P_{batt}$$

For the power system in question, this will be achieved through a central controller which combines forecasting data with data about the current state of the system to plan for power use. This central controller has been implemented in MATLAB and Simulink. The goal of this project is to implement the central controller in LABVIEW so that it can interact with a broader range of hardware needed to implement the central controller in the local power system.

II. Central Controller

The central controller in this case has two components: the planning block and monitoring block. The planning block operates every minute and calculates long term power references whereas the monitoring block operates 4 seconds after the planning block does and every two seconds thereafter; it takes in some references from the planning block, updates the power references and sends data back to itself to be used in the next iteration.

A. Planning Block

The planning block in this central controller takes in 4 types of values: the PVforecast, the loadforecast, the Battery State of Charge, and present PV. The PVforecast is used in conjunction with the load forecast to create a load reference point which is then used to determine the final critical and non-critical load of the planning block. The present PV is effectively added to the load reference point and boundaries are set depending on battery state of charge. The planning block is triggered at the point of islanding and every minute thereafter. A depiction of the planning block can be seen below:

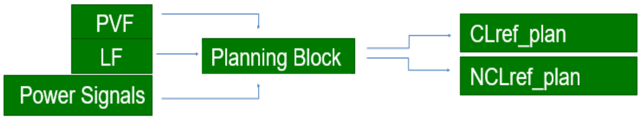


Figure 1. Planning Block I/O

A picture of the planning block implemented in Simulink can be seen here:

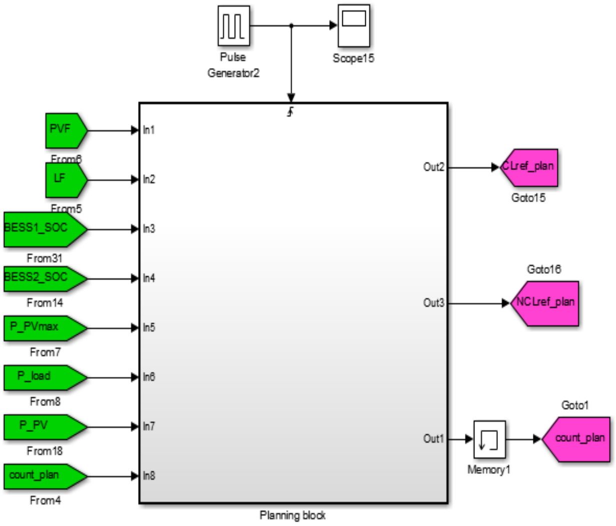


Figure 2. Planning Block in Simulink.

This Simulink Block converted into LABVIEW:

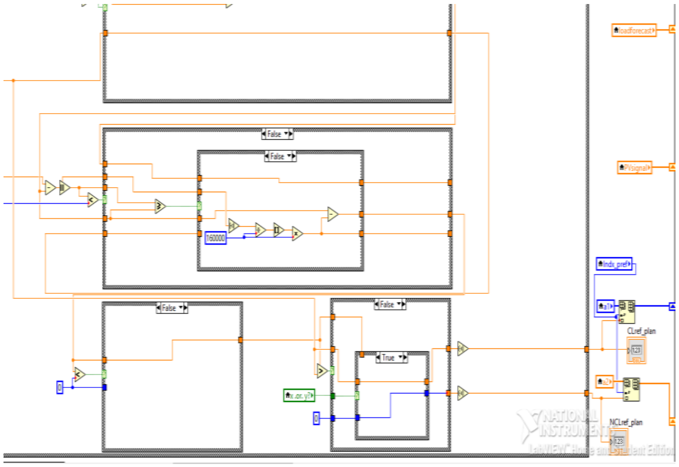


Figure 3. Planning Block in LABVIEW.

B. Monitoring Block

The monitoring block takes in 5 types of values: the present load reference, the power load, both of which are generated with input from the planning block, the present power and load, and battery state of charge. Both the power load and the maximum power load derived from the signal care combined to generate a load reference similarly to the planning block. Then bounds are placed on the values of the final critical and non-critical load reference for the monitoring block. The monitoring block is first triggered four seconds after the planning block and every two seconds thereafter.

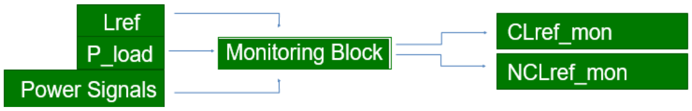


Figure 4. Monitoring Block I/O.

A picture of the monitoring block implemented in Simulink can be seen here:

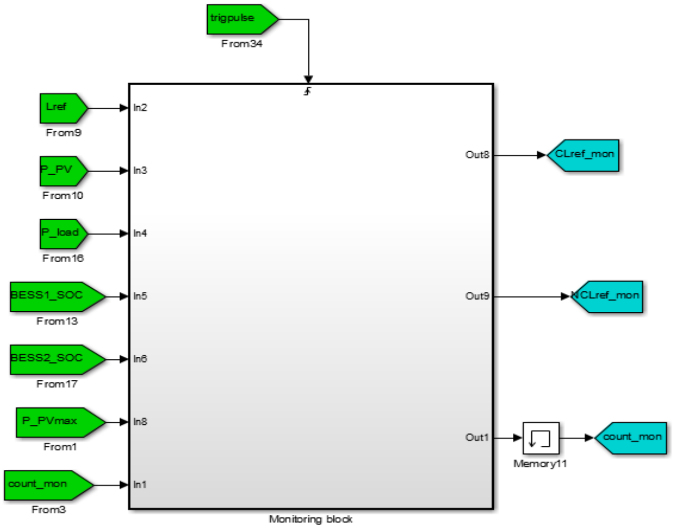


Figure 5. Monitoring Block in Simulink.

The Simulink Block converted into LABVIEW:

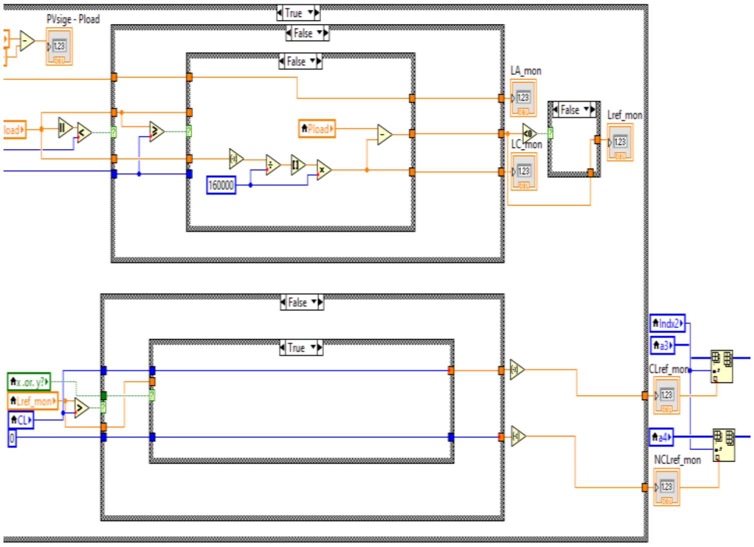


Figure 6. Monitoring Block in LABVIEW.

C. Triggering

Neither the planning nor the monitoring operate continuously and as mention earlier, are triggered at various times throughout the operation of the program. This is done through triggers. For the planning block, the trigger is on for periods of 30 seconds each, and the calculations in the planning block are triggered at the rising edge of the signal. This can be seen in the graph below:

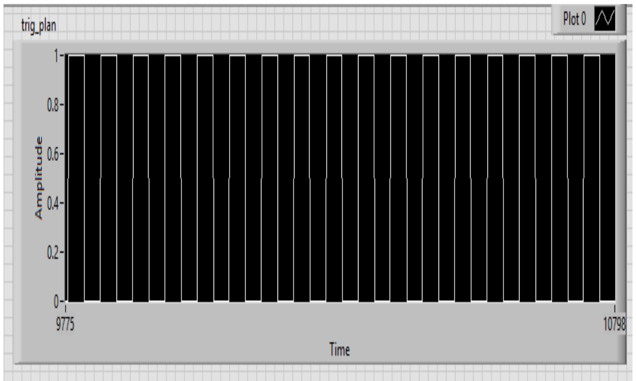


Figure 7. Triggering signal for planning block.

For the monitoring block, the trigger is on for periods of 1 second each, and the calculations in the monitoring block are also triggered at the rising edge of the signal. This can be seen in the graph below:

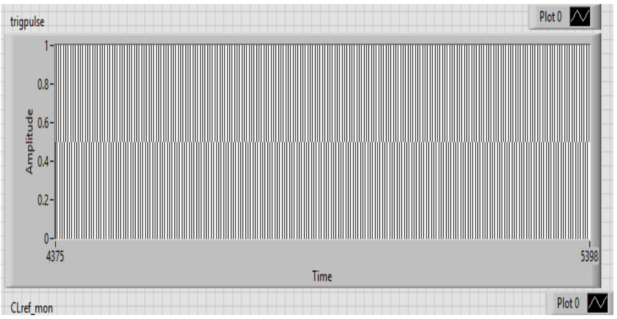


Figure 8. Triggering signal for planning block.

D. Switches

To finally select a value for the critical and noncritical loads for the islanded mode, switch signals are used dictating whether the data from the planning or monitoring block should be used. The data from the monitoring block is used for the first 60 seconds and then the data from the planning block is used for the subsequent second. This is accomplished by creating a switch signal which is on for the first 60 seconds and off for the subsequent second.

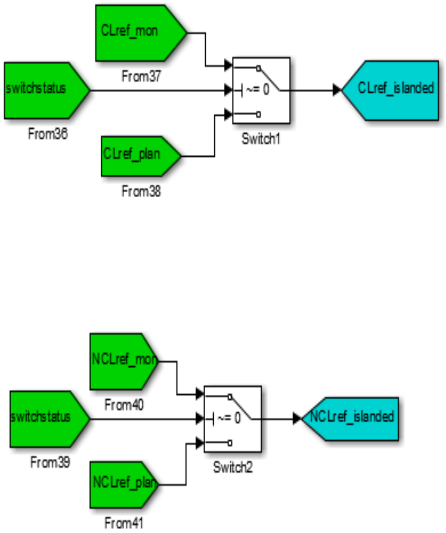


Figure 9. Switch signal shown in Simulink.

As can be seen, the signal here is on for the vast majority of the time but briefly touches zero.

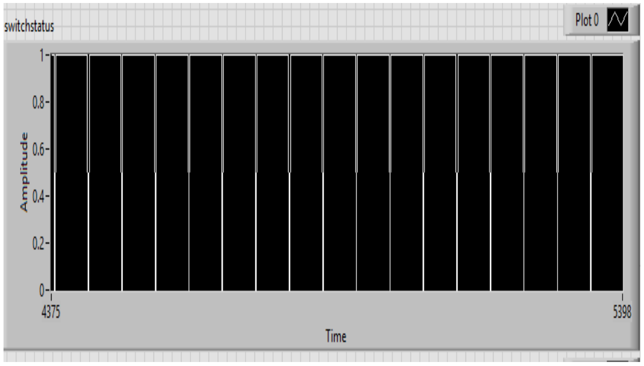


Figure 10. Switch signal shown in LABVIEW.

E. Initial Conditions

There is one additional signal which must be taken into consideration: the initial condition. This signal is off for the first few seconds and on for the rest of the time. The monitoring block inputs are partially dependent on its outputs. So, the inputs are initially set to certain values. To enforce them, an initial condition input is used which sets the constant values when it is off and allows the outputs of the monitoring block to pass through to its inputs when on.

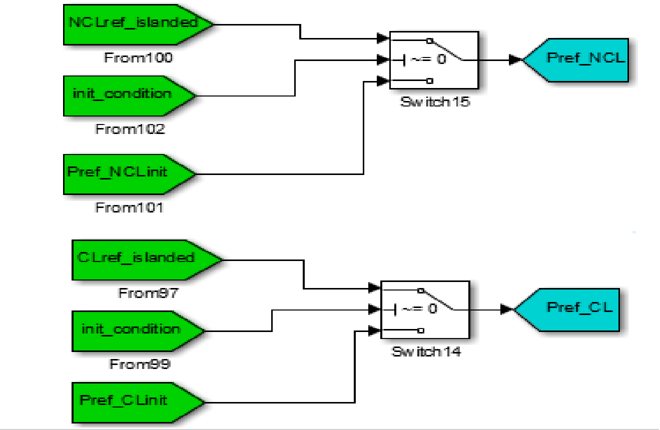


Figure 11. Initial Condition Signals shown in Simulink.

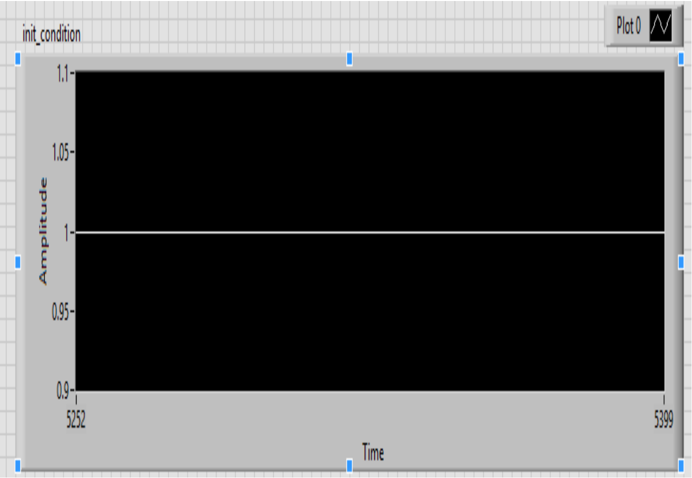


Figure 12. Initial Condition Signals shown in LABVIEW.

III. Results

There are certain key outputs of the Simulink files which have to be compared with the results of the LABVIEW files. The first such data value to be compared is the Signal or a plot of the incoming solar irradiance:

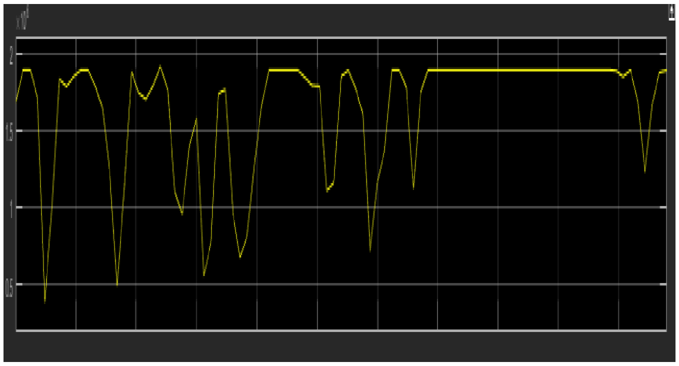


Figure 13. PVSignal vs. Time graph in Simulink.

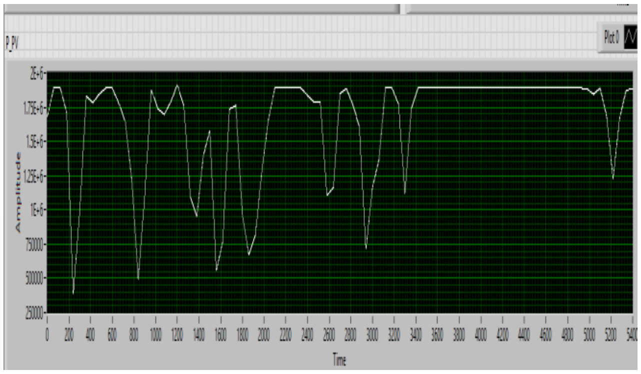


Figure 14. PVSignal vs. Time graph in LABVIEW.

Through the actual PV data for each case has not been compared, the similar apprearance of each graph is a source of confidence that the central controller has been implmented correctly in LABVIEW.

Another pair of signals which can be compared are the critical loads:

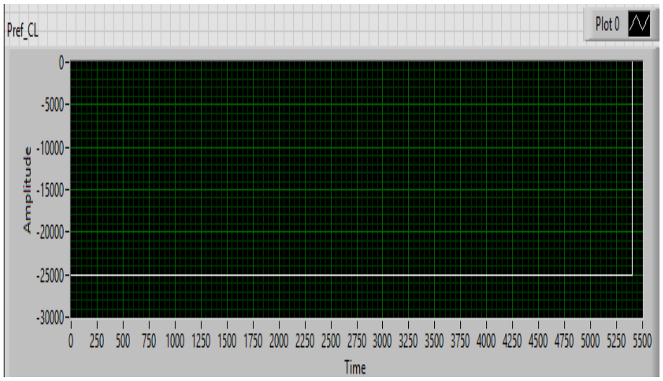


Figure 15. Power Critial Load Reference in LABVIEW.

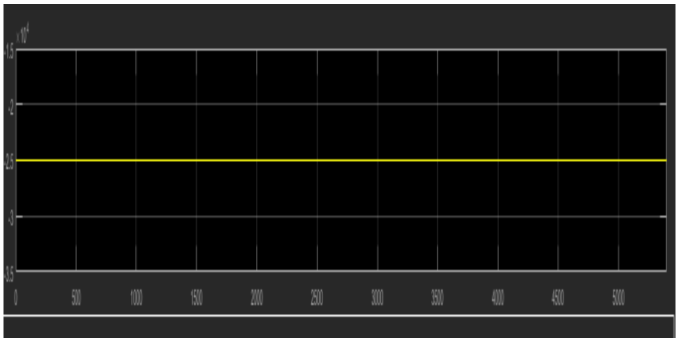


Figure 16. Power Cirtical Load Reference in Simulink.

It is clear from this that the figures 15 and 16 are the same constant value.

Finally, a pair of graphs which need to be examined are the non critical loads from the planning and monitoring blocks:

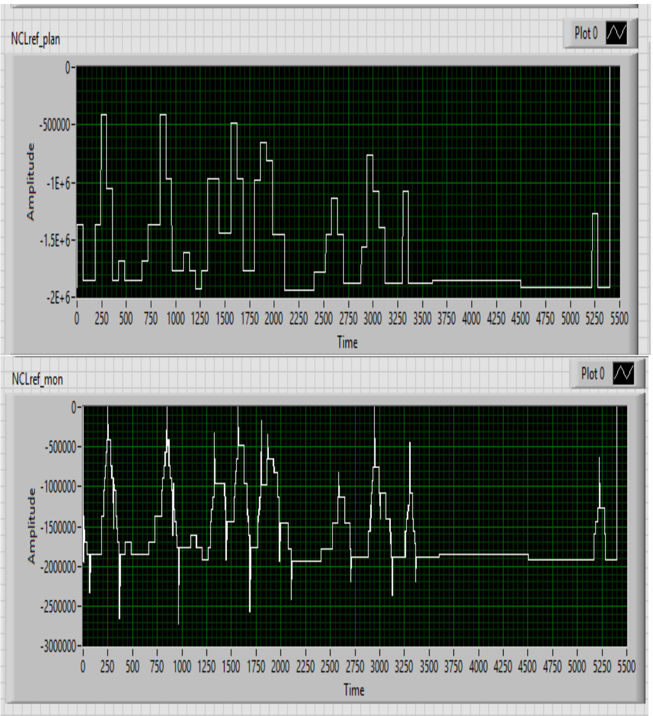


Figure 17. Non Critical loads for the planning and monitoring blocks in LABVIEW shown above.

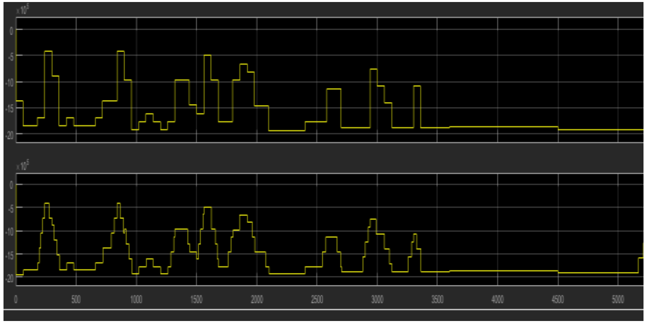


Figure 18. Non Critical loads for the planning and monitoring blocks in Simulink shown above.

Although the graphs appear generally similar, there may be some discrepencies between the LABVIEW and Smulink versions. This is also evident in the comparison of the overall non critical load:

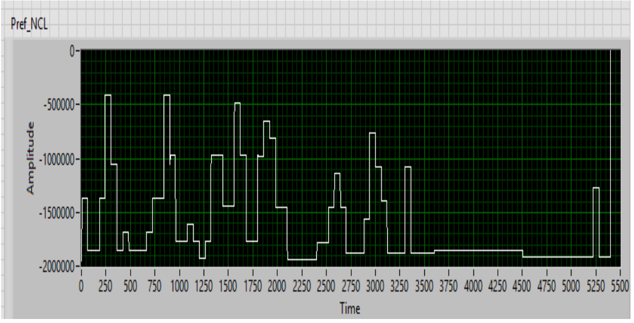


Figure 19. Non critical load for the overall central controller implemented in LABVIEW.

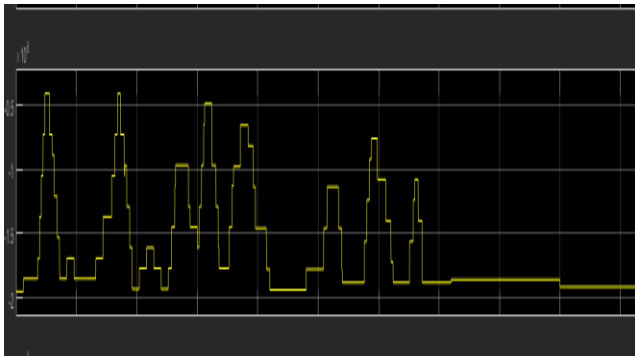


Figure 20. Non critical load for the overall central controller implemented in Simulink.

The graphs shown here are similar in shape however there are certain dissimilarities evident in the graph. A subject of further research could be looking deeper within the LABVIEW program to determine exactly how this could be remedied.

s

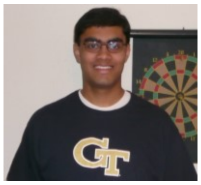
IV. Acknowledgment

The author gratefully acknowledges the contributions of Aakansha Pasricha for the Simulink pictures in this paper and also the CURRENT program and the U.S. DOE for their generous support.

V. References

[1] Hristiyan Kanchev, Di Lu, Frederic Colas, "Energy Management and Operational Planning of a Microgrid with a PV-Based Active Generator for Smart Grid Applications" *IEEE Trans. Industrial Electronics*, vol. 58, pp. 4583-4591, Oct. 2011.

VI. Biographies



Shashwat Sitesh was born in Bristol, UK Dec 23, 1994. He is currently a senior student at the Georgia Institute of Technology in Atlanta, GA.

His employment experience included experience as an Undergraduate Research Assistant at the Georgia Tech Research Corporation working in Bioengineering. He is currently conducting research in the area of Power Systems at UTK through the CURRENT program.