MICROGRID MANAGEMENT ARCHITECTURE CONSIDERING OPTIMAL BATTERY DISPATCH

by

Tim George Paul

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2015

Approved by:

____
Dr. Sukumar Kamalasadan

____
Dr. Yogendra Kakad

____
Dr. Valentina Cecchi

ABSTRACT

TIM GEORGE PAUL. Microgrid management Architecture considering optimal battery dispatch. (Under the direction of DR. SUKUMAR KAMALASADAN)

Energy management and economic operation of microgrids with energy storage systems at the distribution level have attracted significant research interest in recent years. One of the challenges in this area has been the coordination of energy management functions with decentralized and centralized dispatch. In this thesis a distributed dispatch algorithm for a microgrid consisting of a photovoltaic source with energy storage which can work with a centralized dispatch algorithm that ensure stability of the microgrid is proposed. To this end, first a rule based dispatch algorithm is formulated which is based on maximum resource utilization and can work in both off grid and grid connected mode. Then a fixed horizon optimization algorithm which minimizes the cost of power taken from the grid is developed. In order to schedule the battery based on changes in the PV farm a predictive horizon methodology based optimization is designed. Further, the rule based and optimization based dispatch methodologies is linked to optimize the voltage deviations at the microgrid Point of Common Coupling (PCC). The main advantage of the proposed method is that, an optimal active power dispatch considering the nominal voltage bandwidth can be initiated for the microgrid in both grid connected or off grid mode of operation. Also, the method allows the grid operator to consider cost based optimal renewable generation scheduling and/or the maximum power extraction based modes of operation simultaneously or separately based on grid operating conditions and topologies. Further, the methods allows maintaining PCC voltage within the limits during these modes of operation and at the same time ensure that the battery dispatch is optimal.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the constant support and prayers of my parents. They have supported me financially and helped me remain calm all throughout my thesis work.

I would like to express my utmost gratitude to my advisor, Dr. Sukumar Kamalasadan for his guidance and support during these two years of my master's studies. His patience and constant encouragement even when I was getting stuck in a problem have really helped me to go through this process and learn to think through the problem.

I sincerely thank the committee members Dr. Valentina Cecchi and Dr. Yogendra Kakad for taking time to be on my committee and assess my work.

I would also like to express my gratitude to Dr. Sudipta Ghosh who has been a guide for me throughout the period of my research work. I am thankful to my friends for their love and support.

Finally, I would like to thank all the professors and staff at the University of North Carolina At Charlotte who have contributed in enriching my knowledge.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

To face the increased electricity demand and at the same time satisfy the global environmental regulations, we need to reduce our dependence on fossil fuel resources for reducing CO2 emissions. Grid connected renewable power systems have gained great interest in this respect. Two main renewable energy systems that have gained increased attention and has the most update implementable technology are the wind energy systems and the Photo-Voltaic (PV) systems. Various incentive policies by the government have helped in PV development. Other than the incentives the technological expansion also keeps the cost of PV systems decreasing year by year. Taking into account this reduction of the costs, expansion of grid connected PV systems is expected to continue in the next decade. One of the major challenges for PV generation is in managing the intermittent energy production with varying power demand which makes managing the grid demand harder. One way of mitigating this intermittency is to provide additional energy storage to the PV farm. This is mainly provided using a battery energy storage. An integrated energy source system that serves some local load is generally named as microgrid and if this microgrid can be developed with a PV-storage integrated system then it will be a feasible renewable energy based microgrid option. The microgrid is a group of distributed generation resources, energy storage and loads that is normally connected to a low voltage power network [15]. The point of common coupling (PCC) with the grid can be connected or disconnected based on which it can be in grid connected or islanded mode of operation.

Lot of research work has been done for energy management and control of microgrid with renewable sources and storage [33-38,49,50].The hierarchical control structure of microgrid can be found in [15], [16]. In this thesis only the tertiary control for calculating the active power set points is considered. So the work is mainly on the energy management and economic operation of these microgrids with distributed energy storage systems. One of requirements in this area is an architecture for coordination of energy management functions with decentralized and centralized dispatch management architectures in spite of the mode of operation of the microgrid. The main aim of these energy management systems is to minimize the operation costs based on a short term optimization module, which provides the optimal schedule where the time scale is for a 24 hour daily operation. The existing energy management system architectures are reviewed in [36], where centralized and distributed architectures are identified as common control schemes. The centralized model collects all the necessary information for microgrid scheduling and performs centralized operation and control [37]-[40]. In the distributed scheme each component of the microgrid is considered separate and makes its own decisions. The optimal schedule is obtained by iterative data transfers among the components [41]-[43]. The main drawbacks of the centralized scheme are reduced flexibility in adding new components and extensive computational requirements [36]. The hybrid system can perform many applications [4] for utility side and for the consumer side. Utility side applications focus on optimizing properties of microgrid output for distribution upgrade deferral, transmission support, etc. Most of the work in the past has been in applying the battery constraints to the complete grid optimal power flow optimization problem [14, 44, and 51] such that it benefits the utility. But in this case any change of a particular unit dispatch would require a new optimal power flow to be run for the system, because they are all integrated in a single optimization problem. Second the energy storage brings in dynamic time domain couplings among all other decision variables which are the dispatch values of all generators in the system. It creates problems of size since all the previous battery dispatch values need to be stored, computational complexity based on the size of the problem, also the forecasting errors of each time stage will influence all decision variables in all time steps of optimization horizon. The other main problem which is now considered in all energy management systems of microgrid is to handle the PV intermittency. For which the future PV generation data is used to obtain an optimal dispatch which reduces the cost of power taken from the grid over a horizon. This can be found in references [45-48] and [4].

1.1. Optimization

The optimization techniques commonly used to solve the energy management problem described above are linear programming, dynamic programming, and quadratic programming and mixed integer programming. Since the microgrid has storage it makes the optimization problem dynamic i.e. it is time dependent. In this section the basic optimality conditions for static optimization and how it is applied to solve dynamic optimization problem is explained.

1.1.1. Static Optimization

The value the objective function must take so that the function is minimum or maximum is the aim of the optimization problem [31]. It is called static optimization since the solution is calculated for that instant of time and is independent of future time. The basic conditions that need to be checked for the optimality (maximum or minimum) for a single variable function is shown in equation 1.1 and the conditions shown in equations 1.3 and 1.4.

$$f(x) \quad (1.1)$$

$$\frac{df}{dx} = 0 \quad (1.2)$$

$$\frac{d^2f}{dx^2} > 0 \; for \; minimum \; (1.3)$$

$$\frac{d^2f}{dx^2} < 0 \; for \; maximum \; (1.4)$$

The points where all these conditions are satisfied is called a stationary point. But for a multi-dimensional case as shown in equation 2.5, the gradient of the function with respect to all the variables need to be found to obtain the stationary points.

$$f(x_1, x_2, \ldots \ldots . x_n) \quad (1.5)$$

$$\frac{df}{dx_n} = 0 \quad (1.6)$$

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

FIGURE 1.1: Hessian Matrix [31]

To check whether the function has a maximum or minimum at these stationary points the hessian matrix as shown above is calculated. The conditions for optimality are then listed below:

- Calculate the eigenvalues of the Hessian matrix at the stationary point. If all the eigen values are greater or equal to zero then the matrix is positive semi-definite and the stationary point is a minimum. If the eigen values are less or equal to zero then the matrix is negative semi-definite and the stationary point is a maximum.
- If some or the eigen values are positive and others are negative then the point is a saddle point. A saddle point is a point in the domain of the function that is a stationary point but not a local extremum.

These conditions can be used to solve unconstrained optimization problems. Now the optimization problem will have constraints which can be equality and inequality constraints. The optimization of functions with equality constraints are solved using lagrange multiplier method. But when inequality constraints are also included then the problem is solved using the Karush-Kuhn-Tucker (KKT) first order conditions. The KKT conditions are used in the solution of nonlinear programming problems [31]. The conditions are described as shown below.

$$\text{Minimize } f(x_1, x_2, \dots \dots . x_n) \quad (1.7)$$

$$\text{Subject to: } w(x_1, x_2, \dots \dots . x_n) = 0 \quad (1.8)$$

$$g(x_1, x_2, \dots \dots . x_n) \leq 0 \quad (1.9)$$

Where

$w(x_1, x_2, \dots \dots . x_n) = 0$, are the equality constraints.

$g(x_1, x_2, \dots \dots . x_n) \leq 0$, are the inequality constraints.

The Lagrange function is formed as shown below in equation 1.10 with the optimality conditions.

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} w_i(x)\lambda_k + \sum_{j=1}^{p} \mu_j g_j(x) \quad (1.10)$$

$$\frac{\partial L}{\partial x_i} = 0\ , i = 1, \dots n \quad (1.11)$$

$$\frac{\partial L}{\partial \lambda_k} = 0\ , k = 1, \dots m \quad (1.12)$$

$$g_j\ (x) \leq 0, j = 1, ..\, p \quad (1.13)$$

$$\mu_j\ g_j\ (x) = 0\ , j = 1, \dots p \quad (1.14)$$

$$\mu_j \geq 0\ , j = 1, \dots p \quad (1.15)$$

1.1.2. Dynamic Optimization :

The above explanation was considered for static optimization problems at a single point in time. But in this thesis the optimization solution deals not only with the present, but also with the future time periods as well. The optimization problem is solved for a horizon. Hence it is called Dynamic optimization. The main variable definitions used in this type of optimization are control variable and state variables. The control variable is a variable you can control, for example the most common example used is to describe how much you consume at each interval of time. In the case of this study it is the battery dispatch values. The things which we cannot control completely, but that are affected by what we choose as our control are called state variables. In the case of storage it is the state of charge of the battery. So in dynamic optimization, we want to solve for the control variables at every point of time. The state variables can show up in the objective function or in the constraints, but will be determined by the path of the control variables. The dynamic optimization problem can be solved in discrete and continuous time. In discrete time the time period is fixed and the solution is obtained for the fixed intervals. In this thesis the discrete time is considered. The dynamic optimization problem can be solved using dynamic programming or using lagrangean function method. Here the lagrangean function method is used. The most common dynamic optimization problem is the life time consumption problem with fixed assets in discrete time which is described in reference [30]. So now if the lagrangean function is expressed over a time horizon and then can be solved using various optimization techniques like linear programming and quadratic programming. The only difference is that the lagrange function is defined for the full time horizon as shown in equation 1.16 when compared with equation 1.10.

$$L(x, \lambda, \mu) = f(x) + \sum_{t=1}^{T} \sum_{i=1}^{m} w_i(x)\lambda_i + \sum_{t=1}^{T} \sum_{j=1}^{p} \mu_j g_j(x) \quad (1.16)$$

The various methods used for solving the microgrid dispatch solution is described in the next sections.

1.1.3. Linear Programming:

Linear programming maximizes or minimizes the linear objective function with linear constraints. The linear programming method has been used in references [4], [17], [18] and [19] to solve the dispatch solution for the microgrid.

$$\text{Minimize } f(x) = c^T * x \quad (1.17)$$

$$A * x \leq b \quad (1.18)$$

Where

c is the cost coefficient of the decision variables

x the vector of variables to be determined

1.1.4. Dynamic Programming:

In dynamic programming the system refers to a discrete time system with finite time period [5]. The dynamic system can be defined as

$$x_{k+1} = f_k(x_k, u_k) \quad (1.19)$$

Where

k = 0, 1,..N-1.

x_k is the system state variable as described above

u_k is decision variable or control variable

Then the cost of the dynamic system is defined as follows

$$g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k) \quad (1.20\ )$$

Where

$g_k(x_k, u_k)$ is cost in period k and $g_N(x_N)$ is the terminal cost of the system i.e. at the final state.

There is a functional relation between the states and the decision variables also. The aim of the problem is to determine an optimal solution sequence for the defined time period. The aim is to find the solution u_k to minimize the performance index or objective function. In a microgrid generally the states are considered as the state of charge of the battery and the dispatch solution is the decision variable [5].The general algorithm of solution is explained as follows. A cost-to-go function as mentioned in equation 1.20 is defined, which usually starts from the terminal state and moves towards the previous states and it expresses the cost needed to move from N state to state N - 1. The cost-to-go function for the terminal state is fixed. The problem is divided into sub-problems. Every sub-problem is considered as a sub-optimal problem, whose solution could construct a feasible global solution. The total cost of solution will be the sum of each sub-optimal solution. Possible state transitions from each stage to stage are determined based on the limitation of control variable and the limitation of cost function variables which are the constraints [28]. The advantages of using dynamic programming is that the performance index that is the cost functions can be linear, differential ,convex or concave and no specific matlab solver is needed. The disadvantage of this technique is that it requires high memory when studied over long period and the states are discretized with small time step [32].

1.1.5. Quadratic Programming:

A linearly constrained optimization problem with a quadratic objective function is called a quadratic program [24]. Here the Karush-Kuhn-Tucker conditions, which are the first order necessary conditions for a solution in nonlinear programming to have a maximum or minimum, are analyzed for the quadratic program, with a set of linear equalities and inequalities. The general quadratic program can be written as

$$\text{Minimize } f(x) = \frac{1}{2} * x^T * Q * x + c^T * x \qquad (1.21)$$

$$A * x \ \leq b \qquad (1.22)$$

$$x \geq \ 0 \qquad (1.23)$$

Where

c is an n-dimensional row vector describing the coefficients of the linear terms in the objective function.

Q is an (n*n) symmetric matrix describing the coefficients of the quadratic terms

If a constant term exists it is dropped from the model. As in linear programming, the decision variables are denoted by the n-dimensional column vector$x$ and the constraints are defined by an A matrix and an m-dimensional column vector b .When the objective function is strictly convex for all feasible points the problem has a unique local minimum which is also the global minimum. The concept of local and global minimum can be explained using the figure 1.1, points A and D satisfy the conditions of maximum but only point D is global maximum. But if we consider convex or concave functions as in figure 1.2 there is only one minimum or maximum. The sufficient condition to guarantee strict convexity is for Q to

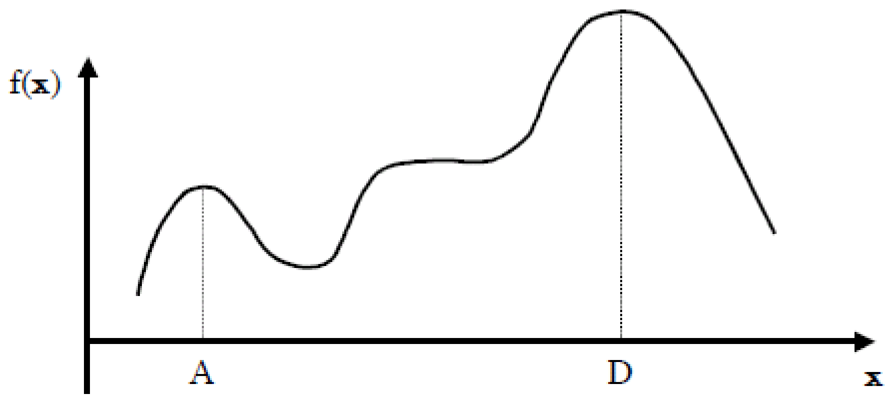

FIGURE 1.2: Local and Global maximum [31]

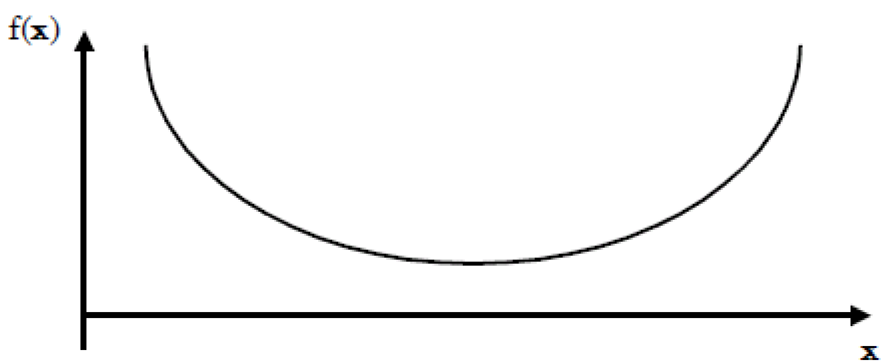

FIGURE 1.3: Convex function [31]

be positive definite[24]. These conditions are sufficient for a global minimum when Q is positive definite. Excluding the non-negativity conditions, the lagrangian function for the quadratic program is shown in (1.4)

$$L(x,\mu) = \frac{1}{2} * x^T * Q * x + c^T * x \ + \ \mu\,(Ax - b) \quad (1.24)$$

Where μ is a row vector, The Karush-Kuhn-Tucker conditions for a local minimum are given as follows.

$$\frac{\partial L}{\partial x_j} \geq 0 , j = 1, \dots n \qquad x^T * Q + c^T + \mu * A \geq 0 \quad (1.25)$$

$$\frac{\partial L}{\partial \mu_i} \leq 0 , j = 1, \dots m \qquad (Ax - b) \leq 0 \quad (1.26)$$

$$x_j \frac{\partial L}{\partial x_j} = 0 , j = 1, \dots n \qquad x^T(Q * x + c^T + \mu A^T) = 0 \quad (1.27)$$

$$\mu_i * g_i (x) = 0 , j = 1, \dots m \qquad \mu(Ax - b) = 0 \quad (1.28)$$

$$\mu_i \geq 0 , i = 1, \dots m \qquad \mu \geq 0 \quad (1.29)$$

$$x_j \geq 0 , j = 1, \dots n \qquad x \geq 0 \quad (1.30)$$

To the equations 1.5 to 2.0 non negative surplus variables y to the inequalities in 1.5 and non-negative slack variables v ε to the inequalities in 2.6 to obtain the equations

$$(Qx + c^T + \mu^T A^T) - y = 0 \quad (1.31)$$

$$(Ax - b) + v = 0 \quad (1.32)$$

The KKT conditions can now be re written as

$$(Qx + + \mu^T A^T) - y = -c^T \quad (1.33)$$

$$(Ax) + v = b \quad (1.34)$$

$$x \geq 0, \mu \geq 0, y \geq 0, v \geq 0 \quad (1.35)$$

$$y^T x = 0, \mu v = 0 \quad (1.36)$$

The equations 1.33 and 1.34 are linear equalities, 1.35 restricts all the variables to be non negative, and the fourth prescribes complementary slackness. An interior point convex algorithm can be used to solve the above equations [25, 24]. The quadratic programming method for solving the microgrid dispatch problem can be found in references [3], [6]. It gives very good results since the problem is formulated in a relaxing form through the lagrangian multiplier [24]. Also quadratic programming is suitable for small problem sizes than 50 variables [2]. This technique has been used in HEV applications as in [26] and [27].

1.2. Scope and Organization

The microgrid interconnection can be at the feeder level or at the consumer level at the end of the lateral feeders as shown in figure 1.4. However, the ability to interconnect the microgrid considering the grid operational benefits and maximizing the supply at the same time meeting the demand depends much on the need and ability of the feeder. For example, if the interconnection point is at the home or neighbourhood level, the need must be to maximize the customer side benefits. Customer side benefits could be to utilize the generation much to satisfy the load taking care of the voltages. On the other hand, if the interconnection point is at the feeder level, the need must be to maximize the grid side benefits. Thus there is a need to modularize the microgrid operation based on the various benefits and at the same time provide a way to link these benefits together. The main scope of the thesis is to develop modular microgrid management algorithms considering

- Maximum use of renewable energy resources in the microgrid and meet the changing demand
- Maximizing the operational profit keeping the optimal use of renewable energy resources.
- Provide a way to improve the operational cost and at the same time maximizing the stored energy.
- Optimally manage the operational cost and at the same time consider the grid voltage stability.

The management of the microgrid is based on the optimal scheduling of the available battery energy storage such that the power taken from the grid is minimized and also be able to calculate dispatch values in off grid mode of operation based on flexible rules. Using this optimal schedule the developed algorithm dispatch values can be applied as reference power set points to the local controllers. To accomplish the above mentioned scope three algorithms are designed. The first one is a rule based dispatch algorithm which works in off grid mode. The second one is a fixed optimization based algorithm which works on grid connected mode of operation. The third algorithm is a predictive horizon optimization based on the concept of Receding Horizon Control [23]. Also, an algorithm that links the rule based and predictive horizon optimization based on the voltage deviations at the Point of Common Coupling is designed. This approach allows the grid operator to automatically switch the dispatch setpoints based on the microgrid mode of operation. The main advantage of the proposed architecture is the developed algorithms can be further scaled based on the PCC considered in the figure 1.4. If the interconnection is at the substation level then the various microgrid component constraints need to be developed in the optimization algorithm and the rule based dispatch algorithm need to be modified to include the other microgrid connected at the various feeders to calculate the dispatch values. Other advantages are listed below.

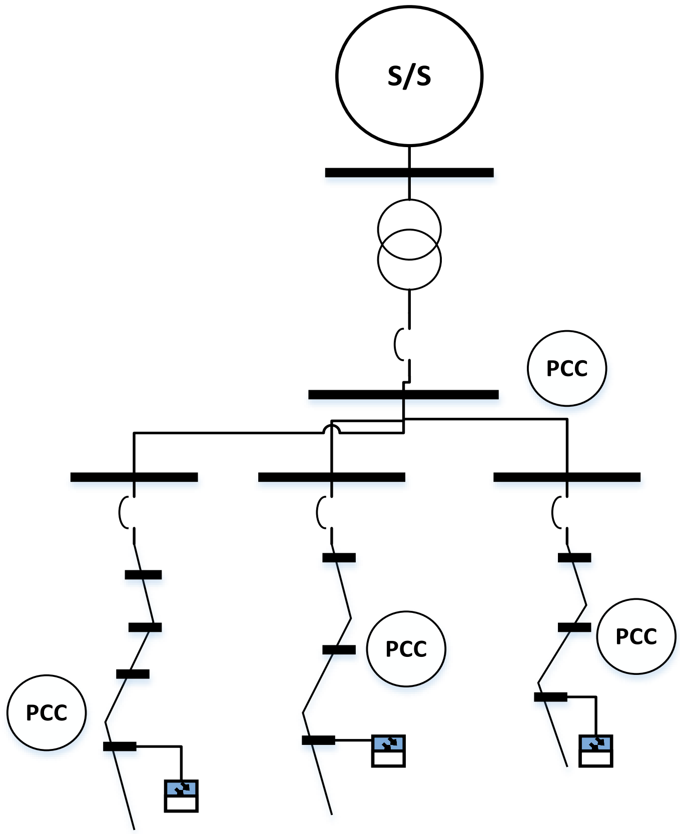

FIGURE 1.4: General distribution feeder

- Rule based dispatch algorithm is flexible and can be applied to a PV+ storage microgrid.
- Dispatch algorithm provides a way to minimize the grid level power use such that the cost is minimum over a fixed horizon.
- The above algorithm is based on dynamic optimization which is further modified to a predictive horizon based optimization algorithm.
- The architecture provides for the seamless transfer or switching of the dispatch values obtained from both the algorithms based on the microgrid status.
- Provides a way to link the new dispatch which minimizes the voltage deviations at the PCC.

This thesis is organized as follows:

- Chapter 2 describes the rule based dispatch algorithm which can be applied to a microgrid in off grid mode of operation and can also provide the power required from the grid in case the load demand cannot be met.
- Chapter 3 presents the dynamic optimization based dispatch algorithm which provides optimal dispatch values for the battery such that the grid power required is minimized.
- Chapter 4 a predictive horizon based optimization which was based on the concept of Receding Horizon Control to develop a method for enhancing the capabilities of dynamic optimization.
- Chapter 5 presents a methodology by which we can switch between the rule based and dynamic optimization algorithms based on the voltage deviations at PCC
- Chapter 6 presents conclusion and future work.

CHAPTER 2 : RULE BASED RENEWABLE ENERGY INTEGRATED POWER DISPATCH ALGORITHM WITH ENERGY STORAGE FOR A MICROGRID

Chapter 2 explains the dispatch algorithm that is based on some flexible rules that can be used in off grid mode of operation. Section 2.1 is the brief description of the rule based dispatch algorithm. The description of the rule based dispatch solution and the simulation results are described in sections 2.2 and 2.3 respectively and the summary is provided in section 2.4

2.1. Overview

As levels of penetration of renewable energy rise, the impact of this on grid operation led to the application of energy storage for renewables. There are various papers proposing this application like various schemes to charge and discharge the battery energy storage system when the solar power output exceeds a threshold and discharges when the load demand is high for various applications like PV smoothing, peak load reduction etc. As a large battery energy storage system is an expensive option for dispatching renewable resources, a control strategy is necessary for optimal use of the available Battery Energy Storage System (BESS). At the distribution level there is a significant market for these hybrid systems especially at the homeowner level due to the economic incentives. But it requires an algorithm that can work in both grid connected and off grid mode that can send power dispatch set points to the controllers in both conditions. This chapter is about one such algorithm called the rule based dispatch algorithm based on [1], where the above dispatch algorithm was used to make the renewable source dispatch able. Similar short term scheduling of battery can also be found in [7] and [2] but it will work only in grid connected mode. But here the algorithm has been modified for battery dispatch scheduling based on a rule based management which can work in island mode of microgrid also, which is based on various conditions of state of charge constraints of the battery. The power scheduling is based on certain constraints, but the final dispatch may not be the optimal dispatch. In this chapter a non-optimal rule based algorithm has been developed that is based on maximum renewable resource utilization. Similar, battery scheduling in a PV/battery system predefined by rules considering the system load profile and the generation characteristic of PV power can be found in references [8]-[10],[20] and [14] where a PV limitation mode was considered. But here the PV output power was not curtailed and the excess power was used by the battery. Also the rules of charging and discharging the energy storage can be modified by each user.

2.2. Description of the Rule Based Dispatch Algorithm

The main components of the microgrid are the PV, energy storage, load and the system is described as shown in figure 2.1.

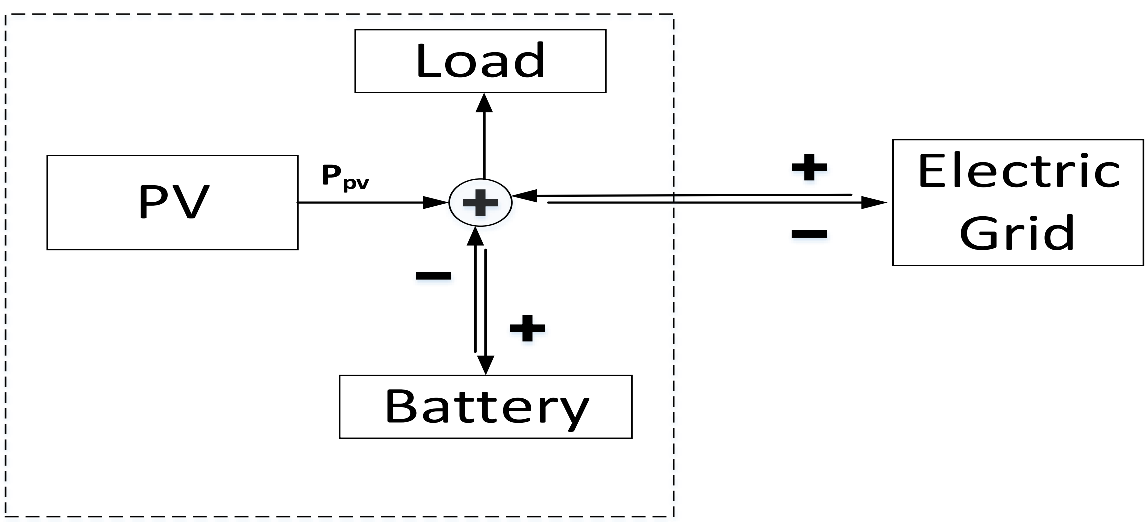

FIGURE 2.1: Microgrid Architecture

As shown in the figure 2.1 the sign convention for power flowing to the grid is taken as negative and the power flowing into the microgrid is taken as positive. The battery dispatch is taken as positive for charging and negative for discharging. The storage charge equation is modelled as shown in equation 2.1. The state of energy stored in the storage at each time step is dictated by the preceding time step energy stored, it's charging rate and discharging rate.

$$E(t) = E(t-1) + dispatch * dt \qquad (2.1)$$

Where

$dispatch\ (t)$ is the battery charge or discharge value at that instant.

$E(t)$ is the energy stored in the battery at that instant.

The above equation has been implemented with a single variable for charge and discharge. So based on the sign of the dispatch solution it will determine if the net battery energy capacity is increased or decreased for that interval. So if the dispatch variable is negative which means the battery is discharging based on the assumed sign convention, then the net charge stored is decreased. Similarly if the dispatch solution is positive then the battery is charging and the net charge available is increased. The value of the $dispatch$ variable is obtained from the rule based algorithm which will be described in the next section. Also the efficiency of charging and discharging process is taken as unity. The state of charge of a battery is its available capacity expressed as a percentage of its rated capacity. Knowing the amount of energy left in the battery compared with the energy it had when it was fully charged gives us the indication of how much longer the battery will continue to discharge before it needs recharging. It is the relative energy level of the Battery. As it is not desired to deplete or overcharge the battery, the SOC of the battery should be kept within proper limits i.e. within (20% and 100%).Furthermore, by limiting the SOC, the charge/discharge time of the battery will also be limited according to the energy left in the battery. The SOC is calculated as shown in equation (2.2) and the constraint equation is (2.3).

$$SOC(t) = E(t)/E_{ref} \qquad (2.2)$$

$$SOC(t)_{ll} \leq SOC(t) \leq SOC(t)_{ul} \qquad (2.3)$$

Where

$SOC(t)$ is the state of charge at that instant.

$SOC(t)_{ll}$ is the lower limit of the state of charge.

$SOC(t)_{ul}$ is the upper limit of the state of charge.

The operating range of the battery is also limited as shown in constraint (2.4) below. This is because every battery manufacturer provides a fixed charging and discharging rate. This constraint is added to limit the degradation and ageing of the battery.

$$Pbatt_{min} \leq Pbatt(t) \leq Pbatt_{max} \qquad (2.4)$$

Where

$Pbatt(t)$ is the battery dispatch value at that instant.

$Pbatt_{max}$ is the maximum charge or discharge rating.

$Pbatt_{min}$ is the minimum charge or discharge rating.

The first step in the rule based algorithm is to calculate the net power demand required from the battery based on equation (2.5).

$$Pbatt(t) = Pload(t) - Ppv(t) \qquad (2.5)$$

Where

$Pload(t)$ is the total load at that instant.

$Ppv(t)$ is the photovoltaic generation at that instant.

The value of $Pbatt(t)$ corresponds to the amount of load that will be satisfied by the BESS (Battery Energy Storage System), now if the value is negative then it shows that there is excess PV generation so the battery can be charged during that interval. If the value is positive then the PV generation is less than the load at that instant and the battery has to discharge to satisfy the load. As explained already the charging and discharging power limits of the battery need to be checked and if the value is violating the limits then it is set to the lower or upper limit value. Then the charge left in the battery is calculated for that instant, based on which the State of charge (SOC) can be obtained as shown in the equations above. Based on the calculated state of charge (SOC) values three conditions are considered. The first condition being that the state of charge is within the limits. The second condition being that the state of charge has hit the lower limit and the third condition is that the stateo f charge has hit the upper limit. These conditions are further explained in detail in the next sections.

Condition 1:

When the SOC value is within the limits, then the corresponding value of $Pbatt(t)$ calculated will be the dispatch solution, since the battery constraints are not violated. Based on the sign of $Pbatt(t)$ it will be to charge or discharge the battery. It is further explained using figure 2.2.

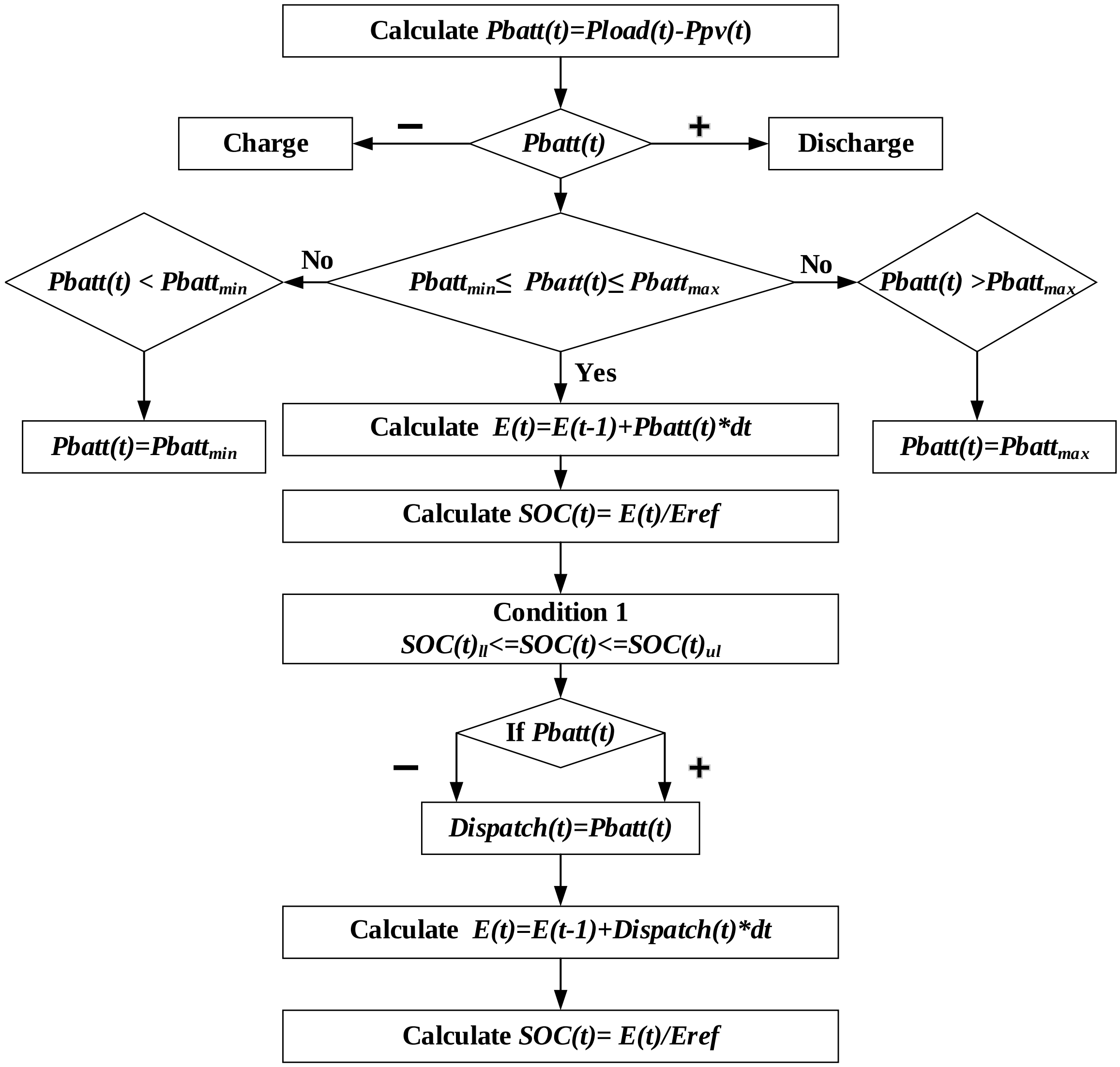

FIGURE 2.2: Flow chart for rule based dispatch-condition 1

Condition 2:

When the SOC value is less than or equal to the lower limit, then if net demand $Pbatt(t)$ is positive then it means the battery has to discharge, but the SOC is at the lower limit, so it cannot discharge and hence the $dispatch\ (t)$ variable is set as 0. If $Pbatt(t)$ is negative then it means excess generation available so battery can charge and the $dispatch\ (t)$ variable is set equal to $Pbatt(t)$ .The flow chart for the above condition is shown in figure 2.3

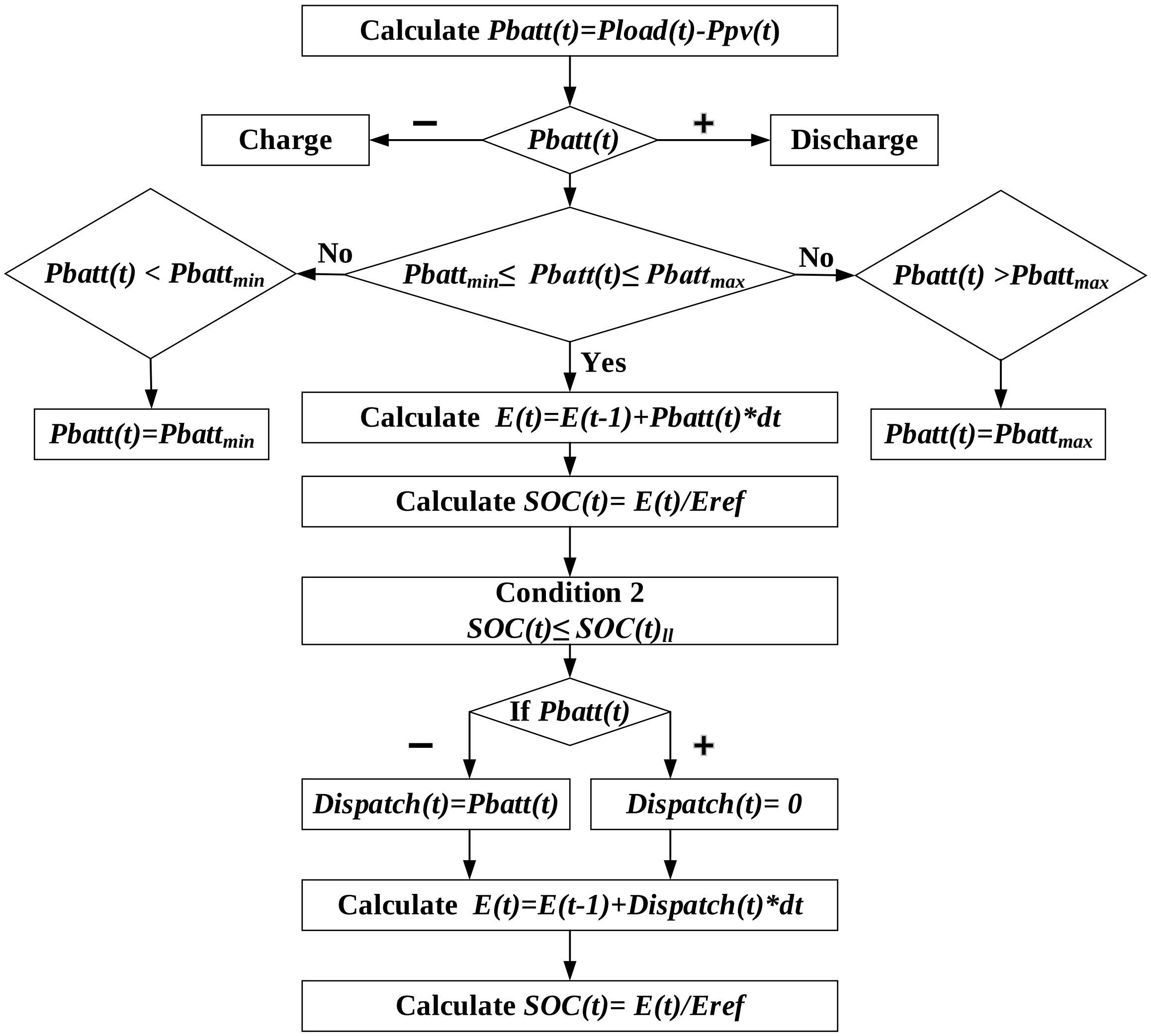

FIGURE 2.3: Flow chart for rule based dispatch-condition 2

Condition 3:

When the SOC value calculated is greater than or equal to the upper limit, then if the $Pbatt(t)$ value is positive, then the battery has to discharge and dispatch variable will be set as $Pbatt(t)$ value. Now if the $Pbatt(t)$ is positive then it means the battery has to charge, and the $dispatch\ (t)$ variable is set equal to 0, as the SOC has hit the upper limit. Based on the $dispatch\ (t)$ variable solution the new battery capacity for that instant is calculated. The flow chart is described in figure 2.4.

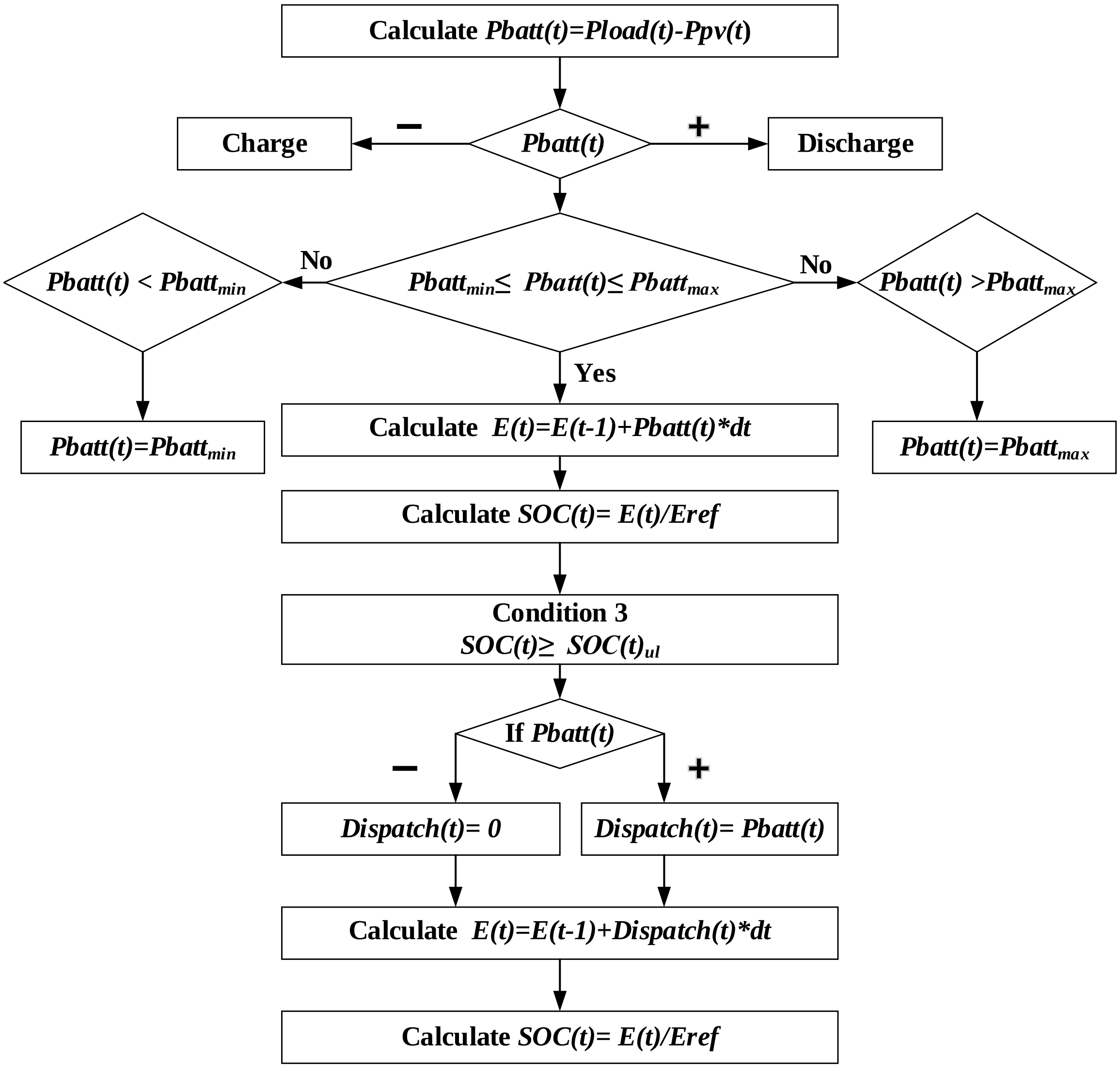

FIGURE 2.4: Flow chart for rule based dispatch-condition 3

The process is repeated in a loop for the full day horizon data. From the present dispatch values the excess power required from the grid can be calculated as shown in equation 2.6.

$$Pgrid(t) = Pload(t) - (Ppv(t) + dispatch\,(t)) \qquad (2.6)$$

Where

$Pgrid(t)$ is the Power from the Electric grid.

$Pload(t)$ is the load at that instant.

$Ppv(t)$ is the photovoltaic generation at that instant.

$dispatch\ (t)$ is the battery charge or discharge value at that instant.

The above equation (2.6) will give the net generation required from the grid to satisfy the load in the microgrid. The positive sign represents net power required from the grid and negative sign represents net power transferred to grid. The $Pgrid(t)$ value is calculated after the dispatch solution is obtained. The above equations and constraints are used in the dispatch algorithm to determine the battery dispatch.

2.3. Simulation Results

The day ahead forecasted photovoltaic generation data for a sunny day is used as the photovoltaic generation data. Half an hour interval data is read from the data and the following PV generation profile is obtained. The no of samples obtained from the initial data is explained as shown below and the generation profile is shown in fig 2.2

$$N * dt = 24\ (hr) \qquad (2.7)$$

$$N * \left(\frac{30}{60}\right) = 24\ (hr)$$

$$N = 48\ samples$$

TABLE 2.1: Simulation parameters

Simulation Parameter values	
T	24 hr
dt	30 min
SOC_{min}	0.2
SOC_{max}	1
BESS	1 MWhr/ 250 KW
Load	500 KW

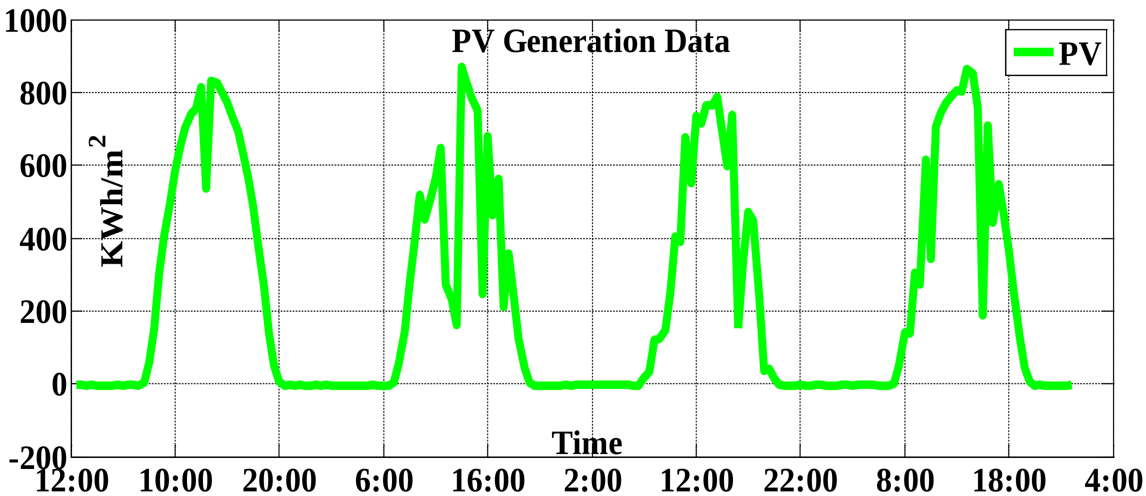

FIGURE 2.5: PV generation data

Each sample corresponds to the 30 min interval data for the day. The project setup is shown in table 2.1 and it consists of the BESS of capacity 0.25 MW with a rating of 1 MWhr connected with a 1MW PV station. A fixed load of 500 KW was considered for the full day. The dispatch results for the full day is explained in three parts based on the conditions as mentioned above. The battery was considered to be fully charged initially.

2.3.1. Dispatch Results During Condition 1:

During early morning when the PV generation is low, the $Pbatt(t)$ value is positive and since the initial condition of BESS was full charged, the battery discharges by 250 KW as that is the discharging rate limit considered as shown in figure 2.6. Since the SOC calculated is within the limits as shown in figure 2.7 and hence the $dispatch\ (t)$ solution is equal to the maximum discharge value possible. It continues to discharge for about 3 hours after which the $Pbatt(t)$ value is still positive but the battery cannot discharge since the SOC lower limit is hit .So during this mode of operation the battery is discharged and the rest of the power is taken from the grid to satisfy the load if grid connected else the load is curtailed. This condition is explained in flowchart shown in figure 2.2.

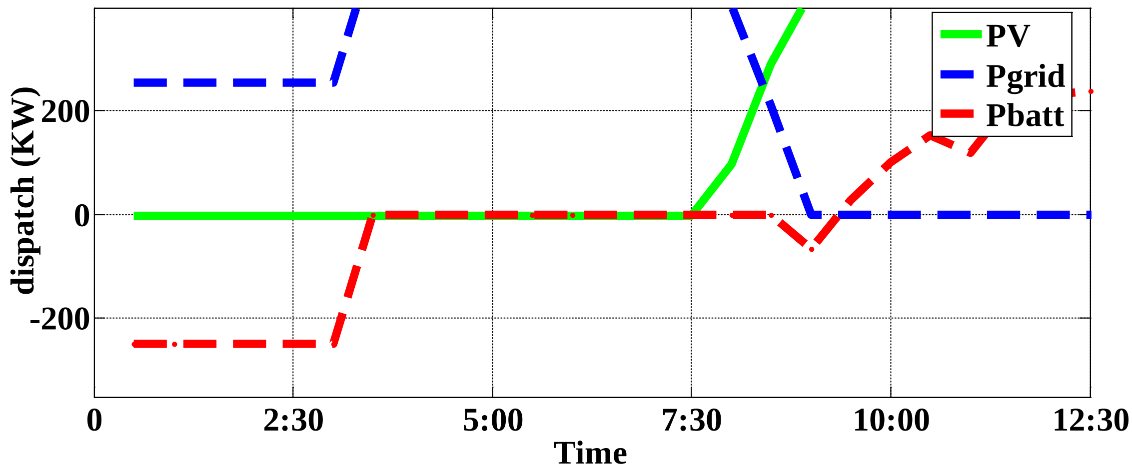

FIGURE 2.6: Rule based dispatch condition 1

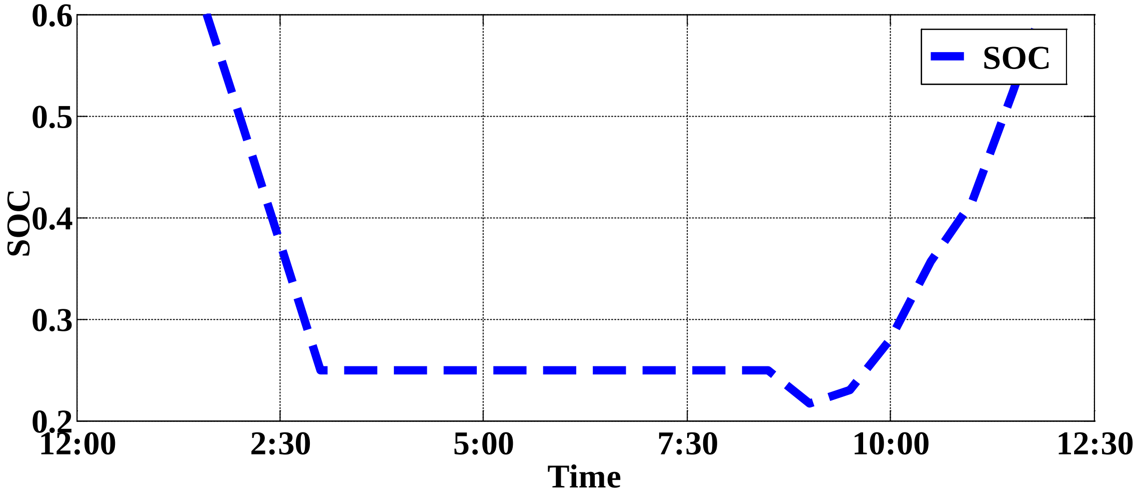

FIGURE 2.7: SOC during condition 1

2.3.2. Dispatch Results During Condition 2:

The battery is fully discharged as shown in figure 2.9 and the battery $Pbatt(t)$ value obtained cannot be discharged as the SOC will go below the lower limit and hence the $dispatch\ (t)$ solution is set to zero and the load has to be satisfied by the grid as the photovoltaic output is zero. As explained in the flow chart When the $Pbatt(t)$ value is positive and SOC has hit the lower limit then based on the sign of the $Pbatt(t)$ variable the $dispatch\ (t)$ value is set to zero as in this case it is positive and the battery cannot be discharged further as explained in the figure 2.3 flowchart above. Also here the battery is not fully utilized as the SOC value has not hit the 0.2 limit. But this will depend on the loading conditions and also it can be modified by adding some new rules.

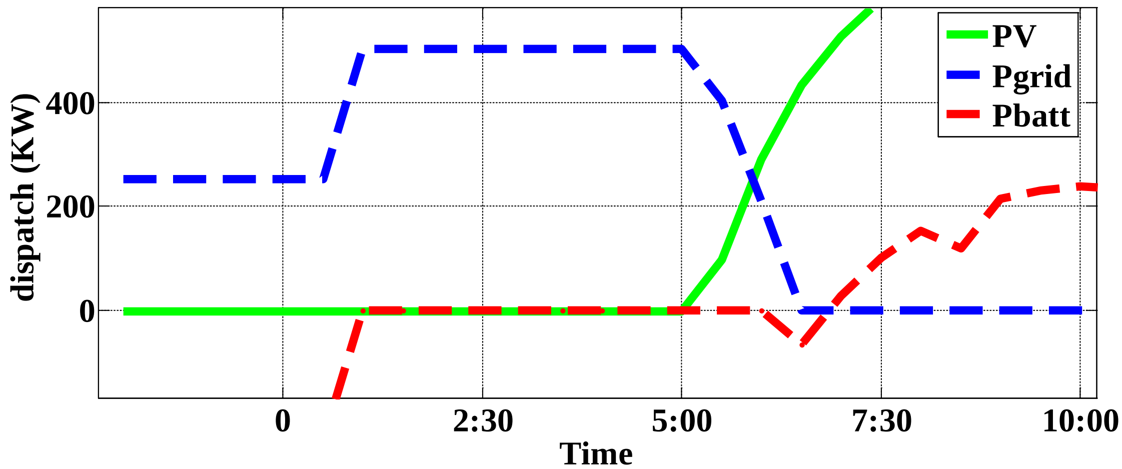

FIGURE 2.8: Rule based dispatch during condition 2

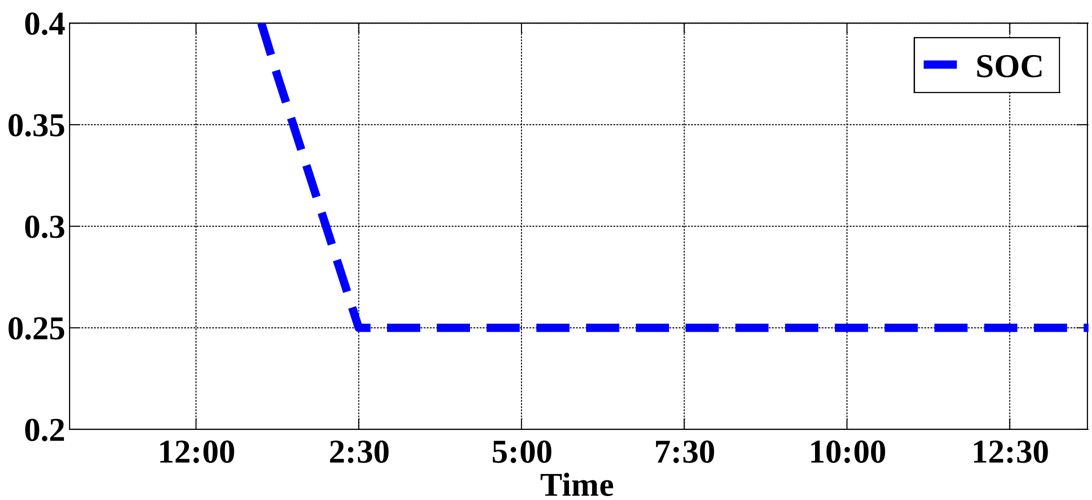

FIGURE 2.9: SOC during condition 2

2.3.3. Dispatch Results During Condition 3:

As the PV generation increases the battery starts to charge till it hits the upper limit of SOC. The total load is satisfied by the PV generation and the power demand from the grid is zero. But once the battery is fully charged, then the excess power is send to grid. This can be shown from figure 2.10 where the positive $Pbatt(t)$ is the charging power and Pgrid value is negative around 3 PM. Similar to the above condition, here also the final battery SOC limit has not hit the upper limit of 1 figure 2.11, as the $Pbatt(t)$ value at that instant is negative and hence based on the flowchart description figure 2.4, since the SOC constraint will be violated the $dispatch\ (t)$ variable is set to zero as can be seen in figure 2.10 around 3 PM and the excess available PV power is send to grid.

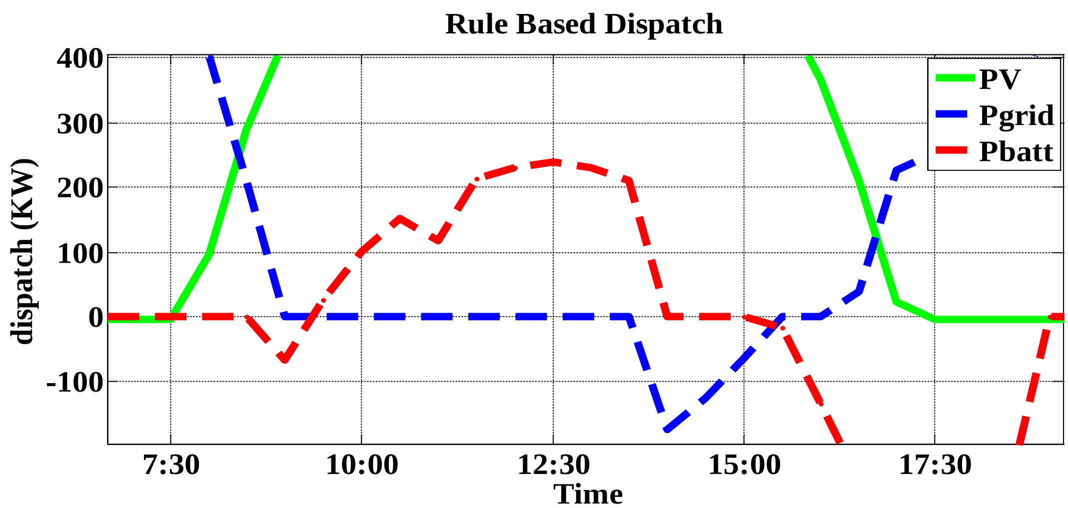

FIGURE 2.10: Rule based dispatch during condition 3

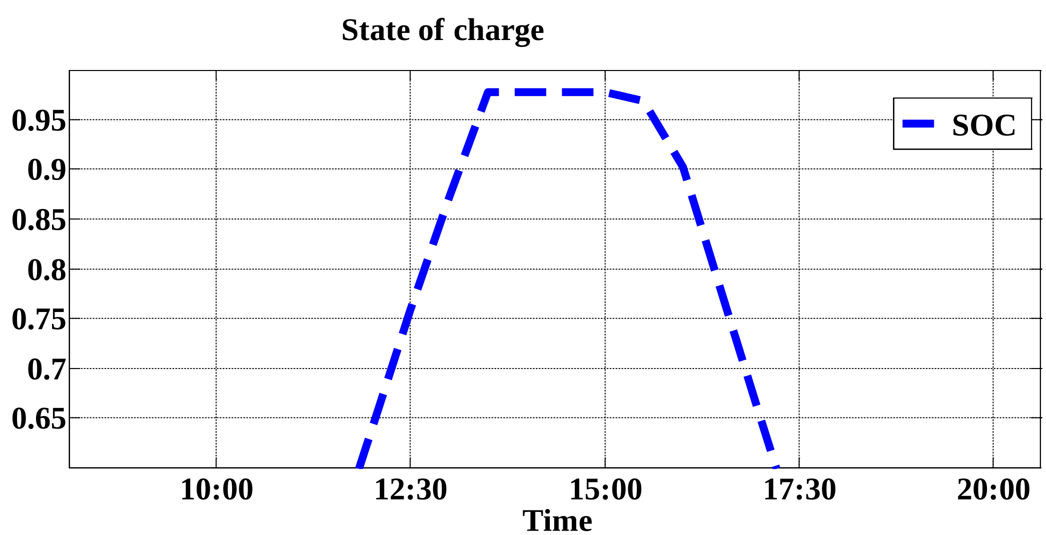

FIGURE 2.11: SOC during condition 3

2.3.4. Complete Rule Based Dispatch Solution:

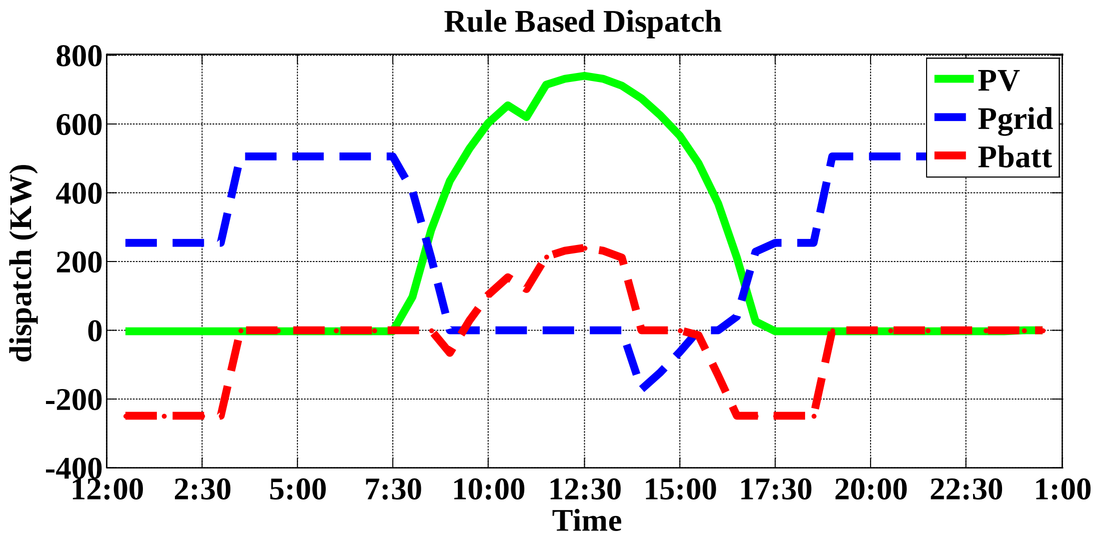

FIGURE 2.12:Full day rule based dispatch

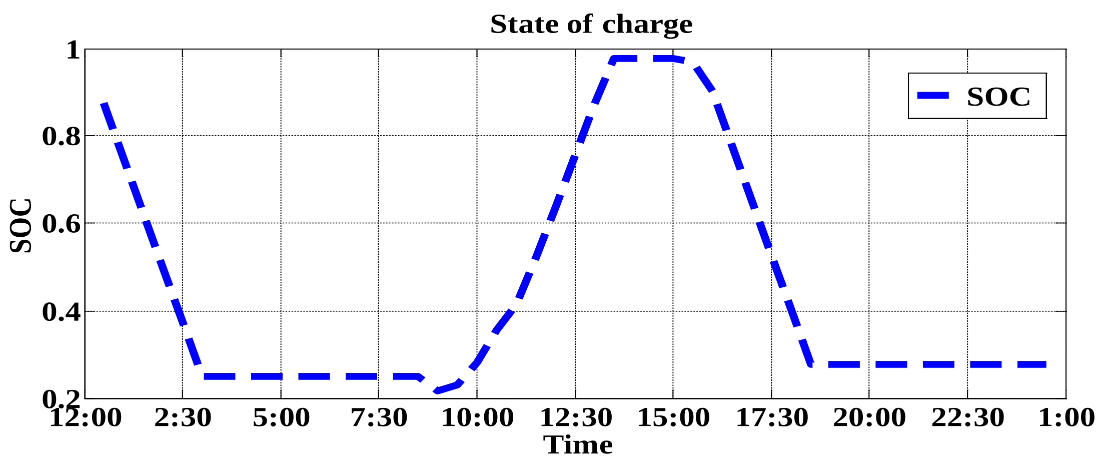

FIGURE 2.13: Full day rule based dispatch-SOC

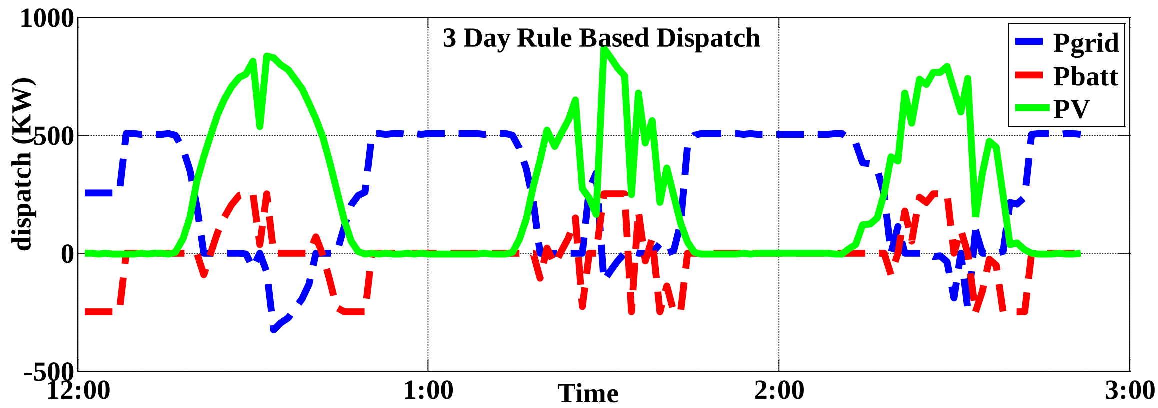

FIGURE 2.14: 3 day rule based dispatch

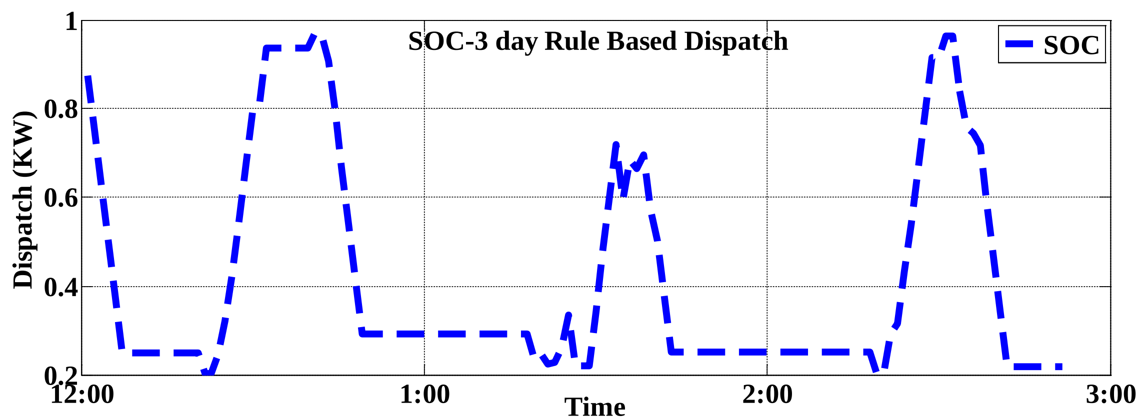

FIGURE 2.15: 3 Day rule based dispatch - SOC

The complete dispatch solution for the full day is shown in figure 2.12 and 2.13. Further a new case was considered where a 3 day profile of PV during sunny and cloudy conditions was considered. The results are shown in figure 2.15 and 2.14 respectively. From the results it can be seen that the battery has not been fully charged means the upper limit was not hit and the same case with the lower limit. But this will depend on the system conditions like load levels, PV generation and battery sizing. Also the constraint limits can be modified based on each user requirement.

2.4. Summary

In this chapter, the rule based dispatch algorithm was described. It is based on maximum resource optimization. There is minimum wastage of renewable resource as the total energy available is used completely to satisfy the load or charge the battery at that instant. The battery is not fully utilized as the state of charge has not hit the lower limit and the upper limit as shown in figure 2.15 and 2.13. But these results can be modified by having different rules. But the above algorithm also ensures that the battery is charged only when the PV generation is high and grid power is not utilized to charge the battery. This dispatch algorithm can be used when the microgrid is disconnected from the grid and incase excess power is required to satisfy the load it can be taken from the grid. Also the rules can be modified if the system conditions are known a day ahead by forecasting and then the power send to the grid and battery scheduling can be modified. Also the SOC limits can be modified based on user needs. No battery sizing algorithm was used to determine the battery size. The aim is to utilize the battery irrespective of its size based on the rules. In case the load cannot be satisfied when in off grid mode then loads have to be curtailed based on some demand response techniques which are not in scope of this work.

CHAPTER 3 : DYNAMIC OPTIMIZATION WITH ENERGY BALANCE INCLUDING STORAGE FOR A MICROGRID

Chapter 3 explains the dynamic economic dispatch of the microgrid connected to the grid. The main aim is to find an optimal dispatch to reduce the grid cost. In section 3.1 an overview of the microgrid economic dispatch is described. The Dynamic optimization algorithm explanation is provided in section 3.2 and the formulation of objective function and constraints for the above problem is explained in section 3.3 The simulation results with Fixed Horizon Optimization is explained in section 3.4 Summary is provided in section 3.5

3.1. Overview of the Microgrid Economic Dispatch

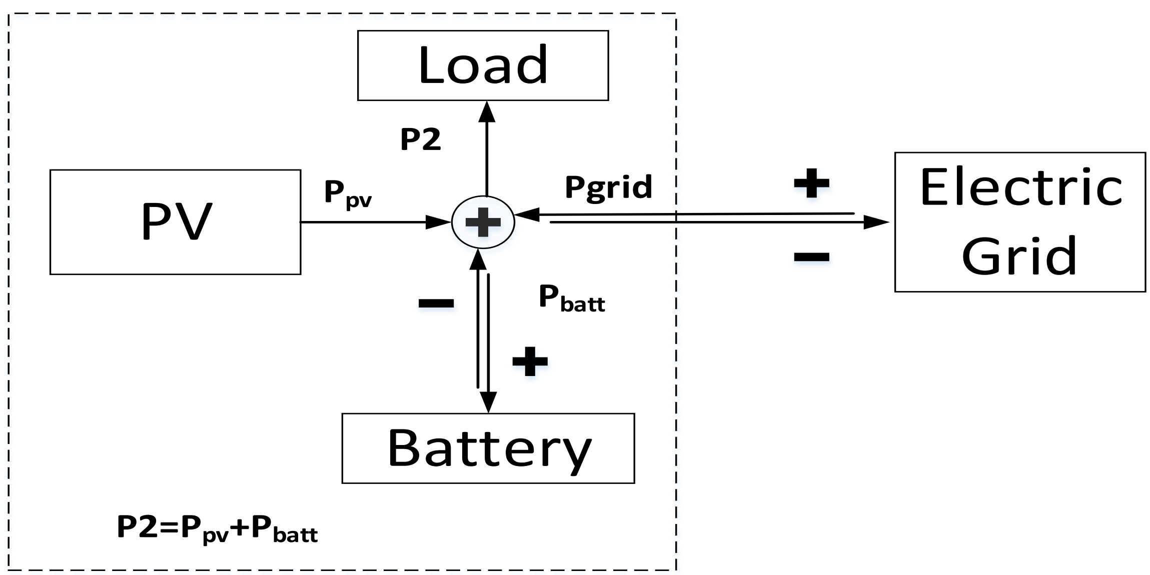

FIGURE 3.1: Microgrid Architecture for dynamic dispatch

Consider the same system as used in the Rule Based Dispatch scheme with a new variable defined as $P2$ as shown in figure 3.1.It represents the total power from the PV plus the battery which will be later explained in the further sections. It is called an economic dispatch since it is based on minimizing the cost of power taken from the grid at the Point of common coupling and obtain the battery dispatch. The optimization problem is formulated for a full day operation of the microgrid. The microgrid economic dispatch has a dynamic formulation due to the presence of energy storage, it has constraints such as power and energy limits which are time dependent and is considered in the microgrid optimization problem. Hence dynamic optimization techniques were used to solve the problem as explained in the introduction chapter. This makes the dispatch solution at every time stage dependent on the dispatch solution of all other time stages. In other words, the dispatch solution in every time step influence the whole schedule of operation. So the schedule of operation is dynamic. For example in a dynamic economic dispatch for a daily operation with small time steps, the dispatch solution at noon will depend on the state of the system at that instant and also the past states of the system. In static economic dispatch, the dispatch solutions in a particular time stage only depend on the criteria of that time stage and are independent from the dispatch solutions in other time stages. In other words, the dispatch solutions in a particular time stage do not influence the dispatch solutions in other time stages. A similar dispatch solution for a microgrid was done in [5],[3] and [2].

3.2. Quadratic Programming

Quadratic programming is used in this thesis for optimization since the objective function which is the grid cost is quadratic with linear constraints. The matlab toolbox was used to solve the above problem. The following description of quadratic programming was taken from reference [11].

$$\frac{1}{2} * x^T * H * x + c^T * x \qquad (3.1)$$

$$A * x \leq b \qquad (3.2)$$

$$Aeq * x = beq \qquad (3.3)$$

$$l \leq x \leq u \qquad (3.4)$$

Where

c is the Vector of linear terms of the quadratic objective function.

H is the Symmetric matrix describing the coefficients of the quadratic terms.

A is the coefficient matrix of inequality constraints.

b is the Vector of inequality right side constraints.

Aeq is the Coefficient matrix of equality constraints.

beq is the vector of equality right side constraints.

l is the Vector of lower bound variables.

u is the Vector of upper bound variables.

The quadratic programming in Matlab uses the interior point convex algorithm which has the following steps:

1. Presolve/postsolve
2. Generate initial point
3. Predictor-corrector
4. Multiple corrections

Each of these are described in [11] in more detail.

3.3. Formulation of Objective Function and Constraints for the above problem:

The general objective function is shown in equation 3.1. It includes the cost function for the grid and battery and also any other source available in the microgrid. Similar problem formulation to solve the above problem can be found in [3], [5]. But in this thesis only the grid cost is considered. Now since the cost function of the grid is quadratic and convex, the above algorithm was used to solve the optimization problem.

$$\sum_{t=1}^{T}\sum_{n=1}^{N}[(F(Pgrid)] \qquad (3.1)$$

$$\big(Pgrid(t) + P2(t) = Pload(t)\big) \qquad (3.2)$$

$$\big(P2(t) + Pbatt(t) = Ppv(t)\big) \qquad (3.3)$$

$$Pgrid^{min}(t) \leq Pgrid(t) \leq Pgrid^{max}(t) \qquad (3.4)$$

$$P2^{min}(t) \leq P2(t) \leq P2^{max}(t) \qquad (3.5)$$

$$Pbatt^{min}(t) \leq Pbatt(t) \leq Pbatt^{max}(t) \qquad (3.6)$$

$$E^{min}(t) \leq Eo + \sum_{t=1}^{N}(Pbatt(t) * dt) \leq E^{max}(t) \qquad (3.7)$$

Where

T is the time horizon of the optimization problem.

The other variable descriptions are the same as in the rule based dispatch algorithm chapter. The PV operating costs are not incorporated here. The battery is modeled with minimum and maximum charging and discharging limits as shown in rule based dispatch algorithm. No battery operating costs are incorporated. The constraints are formulated such that the load demand is to be met by the PV. If the PV output is not enough to satisfy the load demand, the battery discharges to satisfy the load requirement. If the PV output is above the load requirement, the excess generation from the PV is used to charge the battery until full capacity of the battery is reached. The economic dispatch problem is to determine the optimum dispatch of the battery at any given time that minimizes the cost while satisfying the demand and operating limits. $Pgrid(t)$,$P2(t), Pbatt(t)$ are the control variables representing power flows from the grid, PV and battery to the load at any time respectively. The first constraint (3.2) implies that the total power supplied by the PV and battery plus the grid must be equal to the load demand. The constraint (3.3) implies that the sum of the battery charging or discharging power and the power supplied directly to the load from the PV is equal to the PV output at that instant. When the PV generation is low, then based on constraint 3.2, it tries to increase the $P2(t)$ value to match the load and $Pgrid(t)$ value will be decreased. But now since the PV generation is low then the $Pbatt(t)$ value will be decreased i.e. it will have a negative sign to satisfy the constraint (3.3). So the battery discharges without taking too much power from the grid. Now as the PV generation increases and if excess generation is available after satisfying the load then the $Pbatt(t)$ variable is increased to satisfy the constraint (3.3) and hence the battery charges. All the other dispatch variables are constrained by minimum and maximum values as specified by constraints (3.4), (3.5), and (3.6).

The basic algorithm for the matlab code implemented is shown in figure 3.2, which is further explained in the next sections. Initially the PV generation and load data is read from an excel file into matlab. Then the equality and inequality constraints as described above is formulated in matlab code which will be explained in detail in the next sections. Then the objective function is formulated which is the sum of the cost function for each instant of the solution i.e. for the full horizon considered. The structure of the objective function was explained in section 3.2. Initially the c vector was formed which is the coefficients of the linear terms. To obtain this the gradient of the objective function is obtained and then the solution variable which is the x vector is initialized to zero. The H matrix contains the coefficients of the quadratic terms in the objective function and hence the hessian of the function was calculated. Matlab functions are available to calculate the above values. Then the optimization solver was used to solve the above equations to obtain the battery and grid dispatch values required.

3.3.1. Formation of the Bound Constraints in Matlab Code

The optimization problem consists of three bound constraints, two equality constraints and one inequality constraint. Let's consider the bound constraints as shown above first.

$$Pgrid^{min}(t) \leq Pgrid(t) \leq Pgrid^{max}(t) \quad (3.8)$$

$$P2^{min}(t) \leq P2(t) \leq P2^{max}(t) \quad (3.9)$$

$$Pbatt^{min}(t) \leq Pbatt(t) \leq Pbatt^{max}(t) \quad (3.10)$$

Syntax of the Solver in Matlab:

X = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0,options).

The lower bound and upper bound vectors are represented as shown below and then concatenated to form the “lb” and “ub” vectors as required by the solver. Here “N” denotes the total no of samples or time scale considered. Since the solution vector consists of three variables the size becomes 3*N.

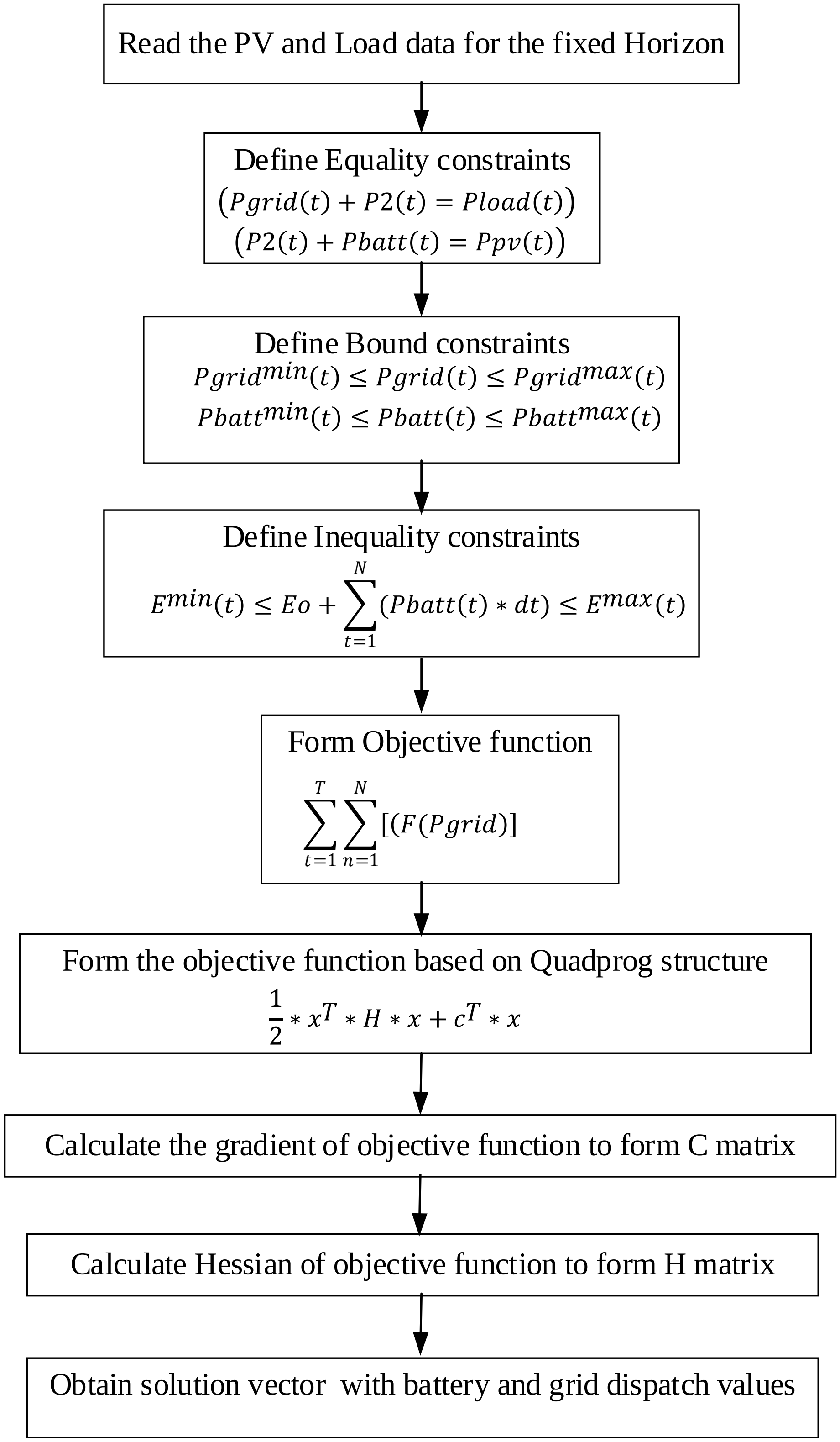

FIGURE 3.2: Flow chart of quadratic programming algorithm

$$\begin{bmatrix} 1 \\ Pgrid^{min} \\ \\ N \end{bmatrix} \leq \begin{bmatrix} \\ Pgrid(t) \\ \\ \\ \end{bmatrix} \leq \begin{bmatrix} 1 \\ Pgrid^{max} \\ \\ N \end{bmatrix} \quad (3.11)$$

$$\begin{bmatrix} N+1 \\ P2^{min} \\ \\ 2*N \end{bmatrix} \leq \begin{bmatrix} \\ P2(t) \\ \\ \\ \end{bmatrix} \leq \begin{bmatrix} N+1 \\ P2^{max} \\ \\ 2*N \end{bmatrix} \quad (3.12)$$

$$\begin{bmatrix} 2*N+1 \\ Pbatt^{min} \\ \\ 3*N \end{bmatrix} \leq \begin{bmatrix} \\ Pbatt(t) \\ \\ \\ \end{bmatrix} \leq \begin{bmatrix} 2*N+1 \\ Pbatt^{max} \\ \\ 3*N \end{bmatrix} \quad (3.13)$$

3.3.2. Formation of the Equality Constraints in Matlab Code

There are two equality constraints in the above problem.

$$\left(Pgrid(t) + P2(t) = Pload(t)\right) \quad (3.14)$$

$$\left(P2(t) + Pbatt(t) = Ppv(t)\right) \quad (3.15)$$

The sum of the solution at each instant should be equal to the right hand side at each instant.

Hence a diagonal identity matrix is considered to implement the above constraint.

The “Aeq” matrix is formed as shown

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & N \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ N & & & \ddots \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 2*N \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ N & & & \ddots \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 3*N \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ N & & & \ddots \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & N \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ N & & & \ddots \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 2*N \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ N & & & \ddots \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 3*N \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ N & & & \ddots \end{bmatrix} \end{bmatrix} (3.16)$$

The “x” matrix is formed as shown below

$$\begin{bmatrix} 1 \\ \\ Pgrid(t) \\ \vdots \\ \vdots \\ N \end{bmatrix}$$

$$\begin{bmatrix} N+1 \\ \\ P2(t) \\ \vdots \\ \vdots \\ 2*N \end{bmatrix} \quad (3.17)$$

$$\begin{bmatrix} 2*N+1 \\ \\ Pbatt(t) \\ \vdots \\ \vdots \\ 3*N \end{bmatrix}$$

The “beq” matrix is formed as shown below

$$\begin{bmatrix} 1 \\ \\ \mathrm{Pload(t)} \\ \vdots \\ \vdots \\ N \\ 1 \\ \\ \mathrm{Ppv(t)} \\ \vdots \\ \vdots \\ N \end{bmatrix} \quad (3.18)$$

3.3.3. Formation of the Inequality Constraints in Matlab Code

Let’s consider the inequality constraint (3.7)

$$E^{min}(t) \leq Eo + \sum_{t=1}^{N} (Pbatt(t) * dt) \leq E^{max}(t) \qquad (3.7)$$

The energy stored in the battery at each instant is dependent on the previous time instant charge stored in the battery and the (charge/discharge) $Pbatt(t)$ solution, so we need to sum the $Pbatt(t)$ solution of previous instants. The above constraint can be divided into two inequality constraints.

Upper limit of Inequality

$$\sum_{t=1}^{N}(Pbatt(t) * dt) \leq E^{max} - E0 \ (3.19)$$

The "A1" matrix is formed as shown below

$$\begin{bmatrix} 0 & 0 & 0 & N \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ N & & & \ddots \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 2*N \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ N & & & \ddots \end{bmatrix} \begin{bmatrix} 1*dt & 0 & 0 & 3*N \\ 1*dt & 1*dt & 0 & \\ 1*dt & 1*dt & 1*dt & \\ N & & & \ddots \end{bmatrix} \ (3.20)$$

The "b1" matrix is formed as shown below

$$\begin{bmatrix} 1 \\ \\ Emax - Eo \\ \vdots \\ \vdots \\ N \end{bmatrix} \ (3.21)$$

Lower Limit of Inequality

$$E^{min} \leq Eo + \sum_{t=1}^{N}(Pbatt(t) * dt) \ (3.22)$$

Converting the above equation into standard form

$$E^{min} - Eo \leq \sum_{t=1}^{N}(Pbatt(t) * dt)$$

$$-E^{min} + Eo \geq -\sum_{t=1}^{N}(Pbatt(t) * dt)$$

The "A2" matrix is formed as shown below

$$\begin{bmatrix} 0 & 0 & 0 & N \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ N & & & \ddots \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 2*N \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ N & & & \ddots \end{bmatrix} \begin{bmatrix} -1*dt & 0 & 0 & 3*N \\ -1*dt & -1*dt & 0 & \\ -1*dt & -1*dt & -1*dt & \\ N & & & \ddots \end{bmatrix} \quad (3.23)$$

The "b2" matrix is formed as shown below

$$\begin{bmatrix} 1 \\ \\ Eo - Emin \\ \vdots \\ \vdots \\ N \end{bmatrix} \quad (3.24)$$

Concatenating the "beq1" and "beq2" matrices to form the "b" matrix

$$\begin{bmatrix} \begin{bmatrix} 1 \\ \\ Emax - Eo \\ \vdots \\ \vdots \\ N \end{bmatrix} \\ \begin{bmatrix} N+1 \\ \\ Eo - Emin \\ \vdots \\ \vdots \\ 2*N \end{bmatrix} \end{bmatrix} \quad (3.25)$$

The "x" matrix is formed as shown below

$$\begin{bmatrix} \begin{bmatrix} Pgrid(t-3) \\ Pgrid(t-2) \\ Pgrid(t-1) \\ Pgrid(t) \\ \vdots \\ N \end{bmatrix} (N*1) \\ \begin{bmatrix} P2(t-3) \\ P2(t-2) \\ P2(t-1) \\ P2(t) \\ \vdots \\ 2*N \end{bmatrix} (N*1) \\ \begin{bmatrix} Pbatt(t-3) \\ Pbatt(t-2) \\ Pbatt(t-1) \\ Pbatt(t) \\ \vdots \\ 3*N \end{bmatrix} (N*1) \end{bmatrix} \quad (3.26)$$

3.4. Simulation Results with Fixed Horizon Optimization

The system considered is same as mentioned in Chapter 2 for rule based dispatch algorithm. Now here since we are optimizing the grid cost, the microgrid is always connected to the grid. The cost function assumed is shown in table 3.1.

TABLE 3.1: Quadratic cost coefficients

	x^2	x	Constant
Coefficient	0.1	12.6	8

3.4.1. Dispatch Results for Full Day Horizon:

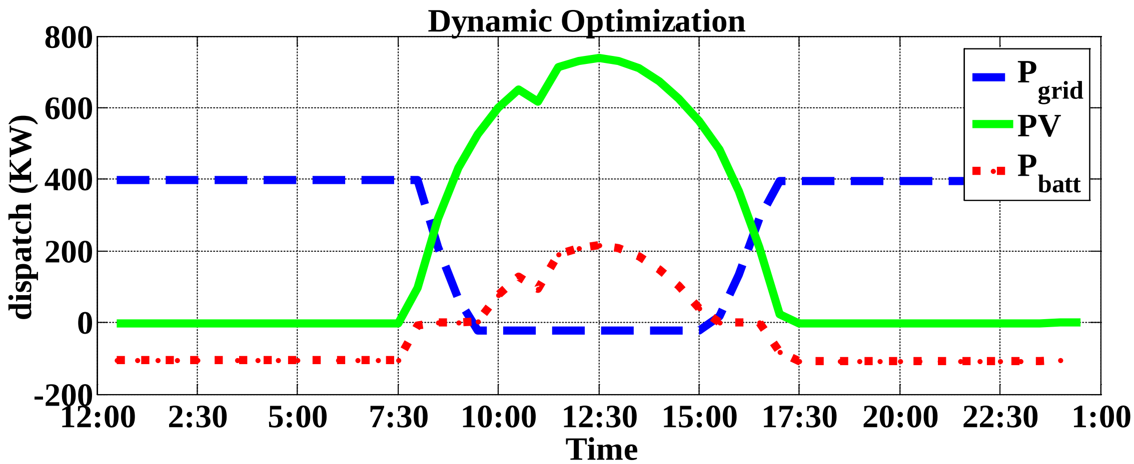

FIGURE 3.3: Simulation result of dynamic optimization.

Figure 3.1 shows the dispatch schedule for a full day. During the early morning and night the load is met by the battery and the grid. The battery dispatch is more uniform as compared to the rule based dispatch schedule. The battery dispatch is 106 KW during the morning time and 108 KW during evening. Now as the PV generation increases it produces more than the load and is able to charge the battery bank and also satisfy the load. But the charging power is optimized such that there is a uniform value of 23 KW power flowing back to the grid so that the overall cost function is minimized. Figure 3.3 shows that the State of charge has hit the lower limit of 0.2 and also the upper limit of 1 and hence the battery is fully utilized. The cost for the above dispatch schedule was 544.5 dollars and the rule based dispatch gave a total cost of 546.5 if we consider that the power is required from grid in off grid mode.

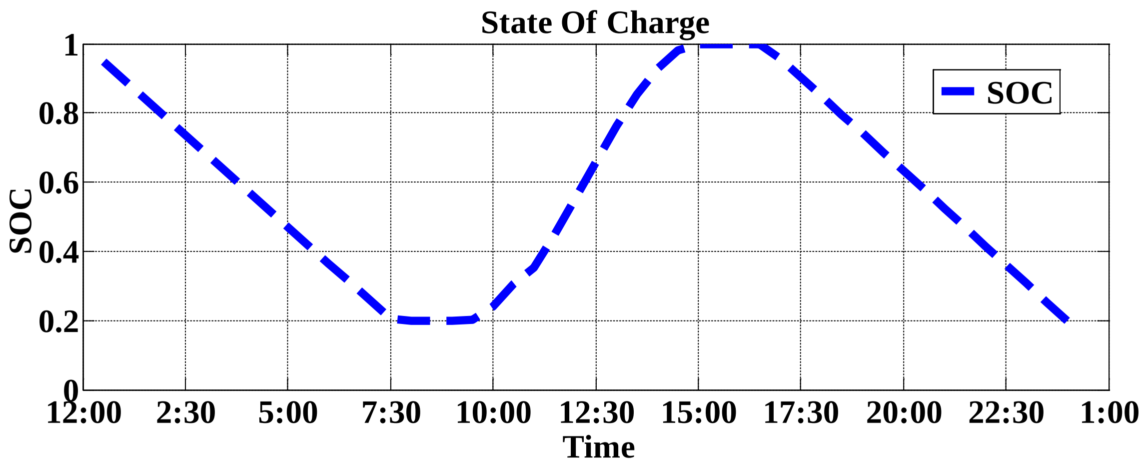

FIGURE 3.4: Dynamic optimization SOC

3.4.2. Dispatch Results for 3 Day Horizon:

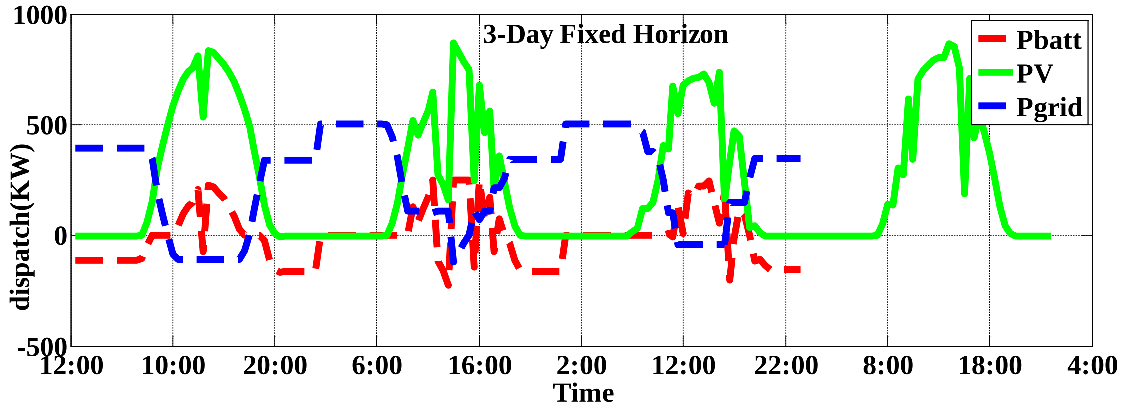

FIGURE 3.5: 3-day dynamic optimization

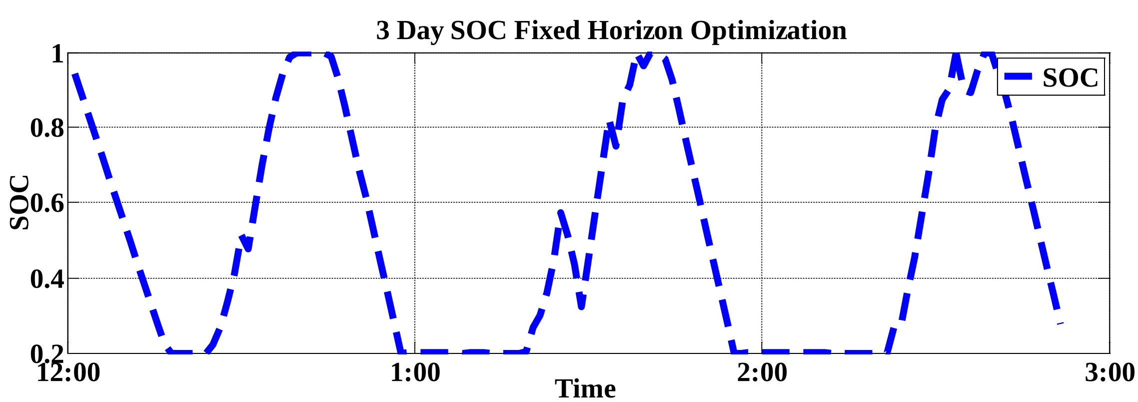

FIGURE 3.6: 3-Day dynamic optimization SOC

Now in this case the same data as shown in the rule based dispatch algorithm was used to obtain the dispatch values for a 3 day horizon. As already seen in the previous case the battery is fully utilized as compared to the Rule Based dispatch results for the same data. But one drawback of the above dispatch scheme is that it sees the fixed horizon only. So it is not able to optimize the battery dispatch based on the next day PV conditions. Hence the above algorithm was modified to include a predictive horizon based dispatch algorithm.

3.5. Summary

In this chapter, the fixed horizon dynamic optimization algorithm was described. It is based on obtaining a battery dispatch such that the power taken from the grid is minimized and hence the cost is reduced for the consumer. There is minimum wastage of renewable resource as the total energy available is used to completely satisfy the load or charge the battery at that instant. The major drawback is that the dispatch schedule is based on the single day data of PV generation and hence the battery state of charge is not optimized for the next day. Hence we go for a predictive horizon based approach in the next chapter.

CHAPTER 4 : PREDICTIVE HORIZON BASED OPTIMIZATION

This chapter explains the predictive horizon based optimization which is based on the concept of Receding horizon control [23]. Description of the predictive horizon based optimization is provided in section 4.1. The optimization algorithm is explained in section 4.2. Simulation results with Predictive Horizon based modification and Fixed horizon is provided in section 4.3 and 4.4. Summary is provided in section 4.5

4.1. Description of the Predictive Horizon Based Optimization

Fixed horizon optimization leads to a dispatch schedule, which begins at the current time and ends at some future time. This fixed horizon solution suffers from a drawback, It does not have information regarding the future PV generation data. This would render the fixed horizon dispatch solutions obsolete or it will not be optimal over a longer period. The above problem is addressed by the idea of receding horizon optimization.

This idea can be summarized as follows:

1. At time k and for the current state, solve an optimization problem over a fixed future interval, say [k; k+N-1], taking into account the current and future constraints as shown in figure 4.1.

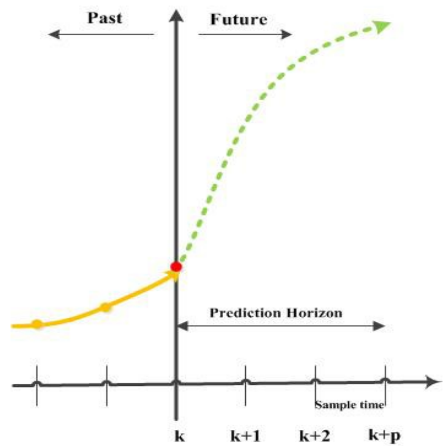

FIGURE 4.1: Solution at first time step [13]

2. Apply only the first step in the resulting optimization solution as shown in figure 4.2.

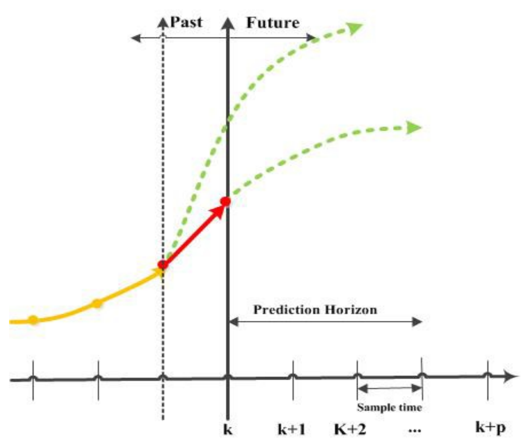

FIGURE 4.2: Solution at next time step [13]

3. Measure the state reached at time k + 1.
4. Repeat the fixed horizon optimization at time k + 1 over the future interval [k + 1; k + N], starting from the current state as shown in figure 4.3.
5. In the absence of disturbances, the state measured at step 3 will be the same as that got from the optimization.

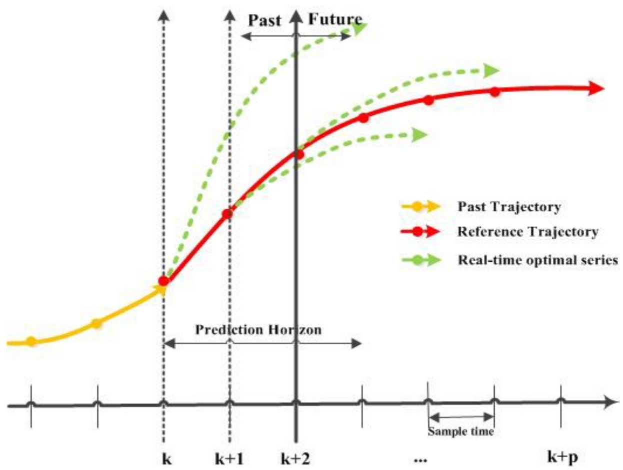

FIGURE 4.3: Overall solution of RHC [13]

4.2. Algorithm Description

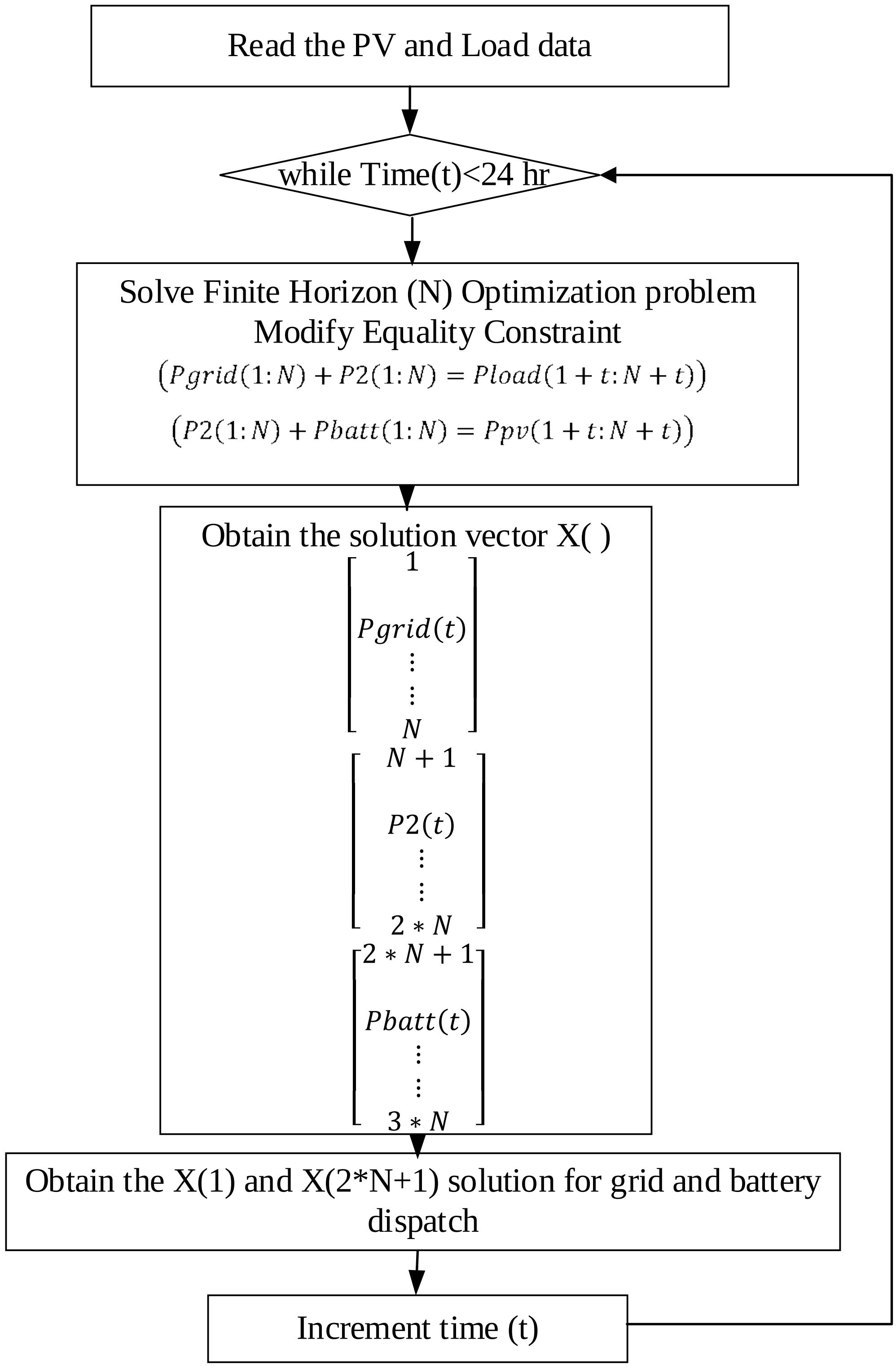

FIGURE 4.4: Algorithm flowchart for predictive horizon optimization

The predictive horizon methodology as described in section 4.1 is implemented in the algorithm as shown in figure 4.4. First the PV generation and load data is read and then the finite horizon optimization problem which was described in the previous chapter is solved. The solution vector is obtained and the structure of the vector was described in the previous chapter. The first solution of the corresponding solution for battery and grid dispatch was selected. Then the time variable is incremented. The main difference is in the way the equality constraint is changed after each solution time step. As already described the constraints are expressed as matrices with fixed size, so by keeping the size constant the data is changed over a horizon value specified by N for each instant of time. This is equivalent to the ‘k’ variable described above in section 4.1 figures.

4.3. Results with Predictive Horizon Based Optimization

The algorithm was run with a horizon of one day and the dispatch schedule obtained was used to calculate the cost. For a full day horizon the dispatch setpoints obtained for the present day was based on the next day data. As shown in table 3.2 the cost for N=48 was higher as compared to the cost calculated in fixed horizon optimization which was 544.5 dollars. But as the horizon length was decreased the battery discharged more and hence the power demanded from the grid reduced and hence the battery dispatch power reduced. This is shown in figure 4.5. Here N signifies the time horizon and the data used is every 30 minute PV generation data hence as explained in chapter 1 there will be 48 samples. The following cases were analyzed. Since the horizon window size is modified based on samples the figure x-axis is not represented in time values.

Case1: N= 48 (full day horizon)

Case2: N= 24 (half day horizon)

Case3: N=12

Case4: N=5

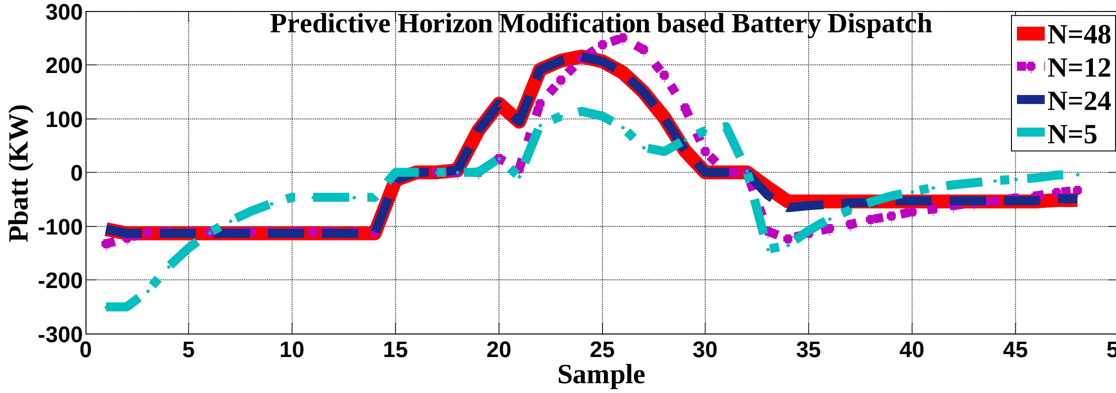

FIGURE 4.5: Predictive horizon battery dispatch for various horizon

Table 3.2: Cost of RHC with various horizon size

	N=48	N=24	N=12	N=5
Cost($)	554.1313	553.825	549.6287	544.8917

As shown in figure 4.4 as the window size is reduced the overall cost for predictive horizon based optimization has reduced. For lower value of window size the battery discharges more and hence the power demanded from the grid is reduced. But as the window size increases as it sees the full next day PV generation schedule it gives a lower dispatch set point. Even though the cost is higher, the advantage is that the state of charge of the battery will be higher at the end of the day. So for the next day the battery power can be used to satisfy the loads and hence grid power required will be reduced. Now if a cost value was added for the conserved state of charge then the overall cost will be lesser for the predictive horizon based approach.

4.4. Comparison with Fixed Horizon and Predictive Horizon Optimization

To show the advantage of using the predictive horizon modification both the algorithms were run for 3 days considering sunny and cloudy weather days. In fixed horizon case the optimization algorithm was run for 3 days and at the end of each day the battery SOC was updated, if SOC was below a particular value then the cost for the next day was increased by 10% which is nothing but the grid demand will be higher for the next day. This is done to bring an economic significance to the reduced battery discharge at the end of the day for the predictive horizon optimization dispatch as shown in figure 4.6 as compared to figure 4.8 were the battery is fully discharged at the end of the day. After running the simulation for one day, the initial battery charge condition is changed to the previous day SOC. Then the optimization was re-run with next day data and the cost coefficients were increased based on the state of charge condition of the battery.

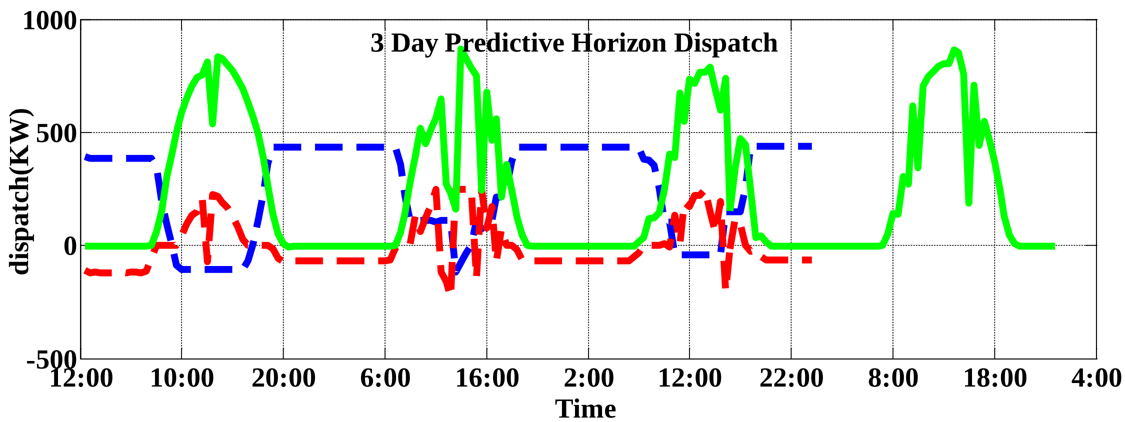

FIGURE 4.6: Predictive horizon 3 day dispatch

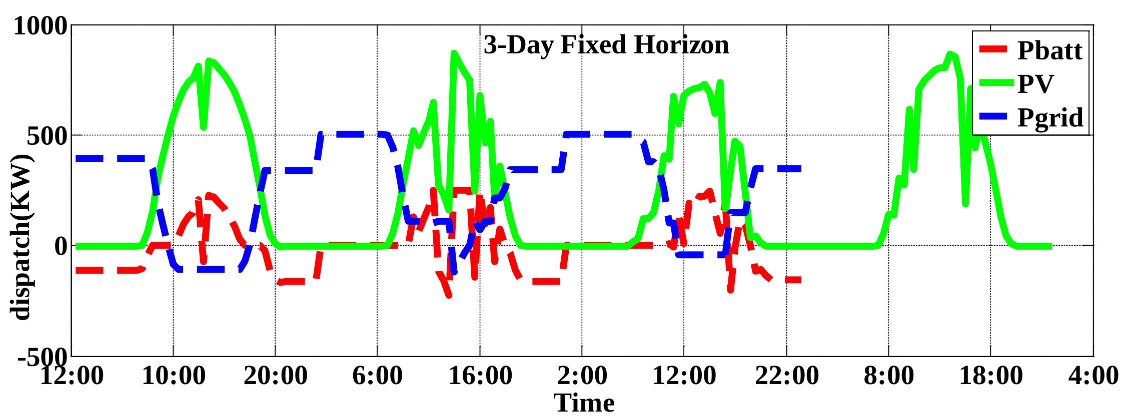

FIGURE 4.7: Fixed horizon 3 day dispatch

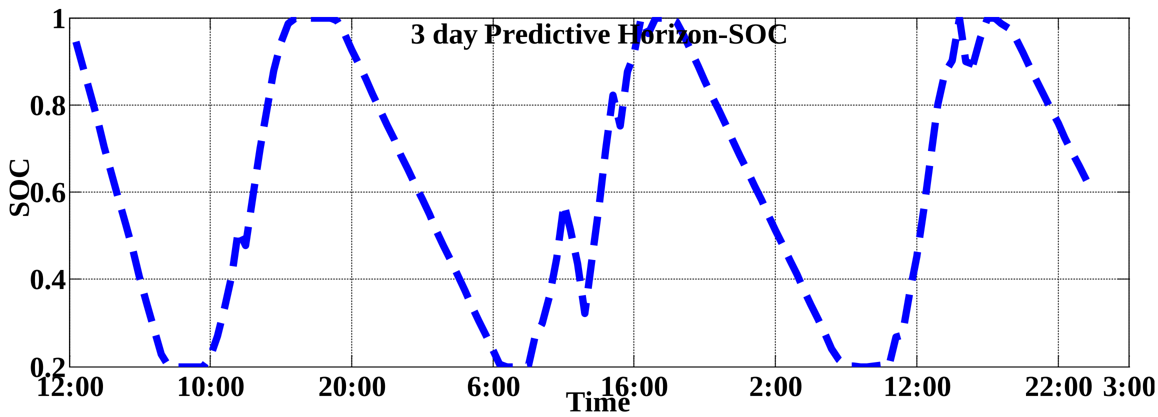

FIGURE 4.8: Predictive horizon 3 day dispatch SOC

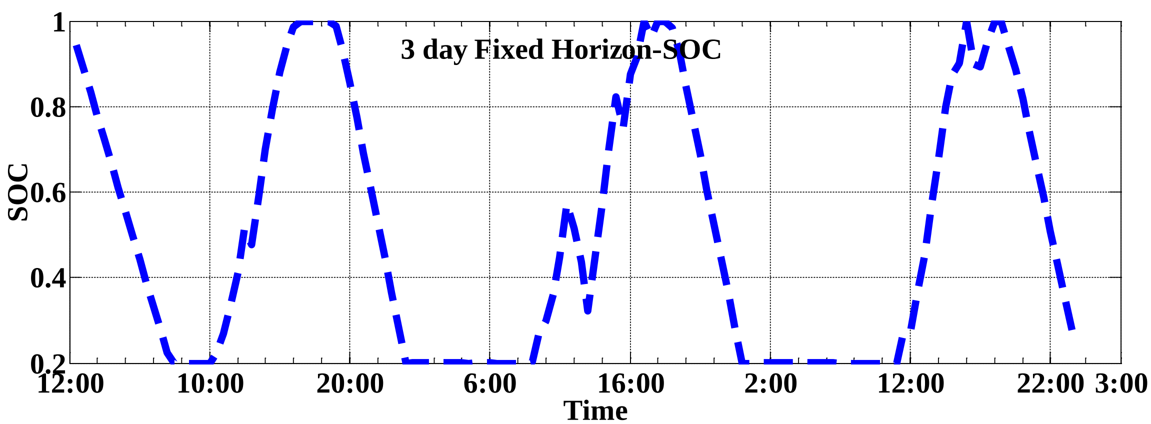

FIGURE 4.9: Fixed horizon 3 day dispatch SOC

Based on the figures 4.8 and 4.9 it is clearly visible that the SOC in the case of fixed horizon dispatch has hit the lower limit by midnight of the first day, but the battery is not drained and the SOC is about 0.6. Hence the battery can take part in the next day dispatch schedule. A cost comparison was done for each day and then the net cost was calculated and shown in table 3.3 and table 3.4.For the First day the fixed horizon optimization gave a lower cost compared to predictive optimization. But for Day-2 since the battery is fully discharged the cost for fixed horizon optimization is higher. Hence for 3 days the total cost is higher for Fixed horizon optimization.

Table 4.3: Cost of 3 days-predictive optimization

Predictive Horizon Control				
	Day-1	Day-2	Day-3	Total
Cost ($)	510.08	556.96	544.67	1611.71

Table 4.4: Cost of 3 days-fixed horizon

Fixed Horizon Optimization				
	Day-1	Day-2	Day-3	Total
Cost ($)	498.62	612.75	601.03	1712.4

4.5. Summary

The predictive optimization dispatch is best when compared for a longer duration of day's .Only then can the reduced battery dispatch values have some significant effect in the total cost. But if there is no large scale variation in the PV generation data then the fixed horizon dispatch values will have lesser cost. But the advantage of this algorithm can be better understood in the next chapter when dispatch set points need to change based on the status of the microgrid. The fixed horizon dispatch will only give a fixed dispatch set points for the whole horizon. So any variations in the PV generation cannot be seen in the dispatch set points.

CHAPTER 5 : SWITCHING BETWEEN DISPATCH ALGORITHMS BASED ON MICROGRID STATUS

In this chapter the voltage at PCC is calculated for the dispatch algorithms described above and the Rule Based Dispatch algorithm and the Predictive Optimization algorithm has been linked to get a new dispatch with minimum voltage deviations. In section 5.1 an overview has been described regarding the system. A flow chart describing the voltage calculation is described in section 5.2. The linking between the dispatch algorithms is described in section 5.3 and the summary is provided in section 5.4

5.1. Overview

The microgrid was considered as attached to a radial distribution system. Now when the microgrid is in off grid mode the dispatch set points can be obtained based on the rule based algorithm and in case of grid connected mode based on the optimization algorithm. So if the microgrid switches between grid connected and islanded mode of operation then the dispatch set points have to be changed accordingly, but it depends on which state the microgrid was running initially. Now in off grid mode since there is no cost optimization, the dispatch results are not optimal but the maximum power available from the renewable source is used completely. When microgrid transfers from the off grid mode to grid connected mode then the rule based dispatch results are not valid anymore and we need to recalculate the optimized active power dispatch values based on the present SOC as initial condition. This is where the linking between the rule based and optimization based algorithm is done so that there can be a seamless transfer. This could be implemented based on a status signal or based on voltage deviation at PCC. For this case it is assumed that the Rule Based Dispatch algorithm can also be used for grid connected mode of operation since there is no curtailing of loads.

5.2. Voltage Calculation at PCC

The system considered was a modified IEEE 9 bus radial distribution feeder. The load connected at last bus was considered to be 500 KW as in previous chapters and was included in the microgrid architecture. So the net grid power demand obtained from the dispatch algorithm becomes the net power demand or net load at that bus. The voltage calculation flow chart is shown in Figure 5.2. Now if a sensor is available to measure the voltage at the PCC, then system power flow data is not required. But since in this work to calculate the voltage at PCC the MATPOWER power flow architecture was used to run the power flow and obtain the voltages. The voltage profile for a single day for the Rule based and Predictive dispatch is shown in figure 5.1.

Where V_{rbd} is the Rule based Dispatch Voltage.

V_{rhc} is the Predictive Optimization Voltage.

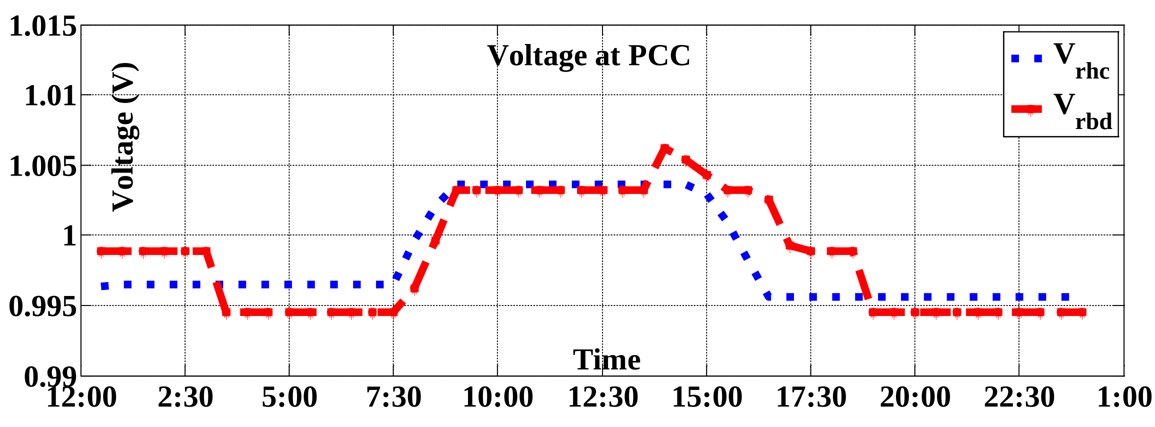

FIGURE 5.1: Voltage calculation at PCC

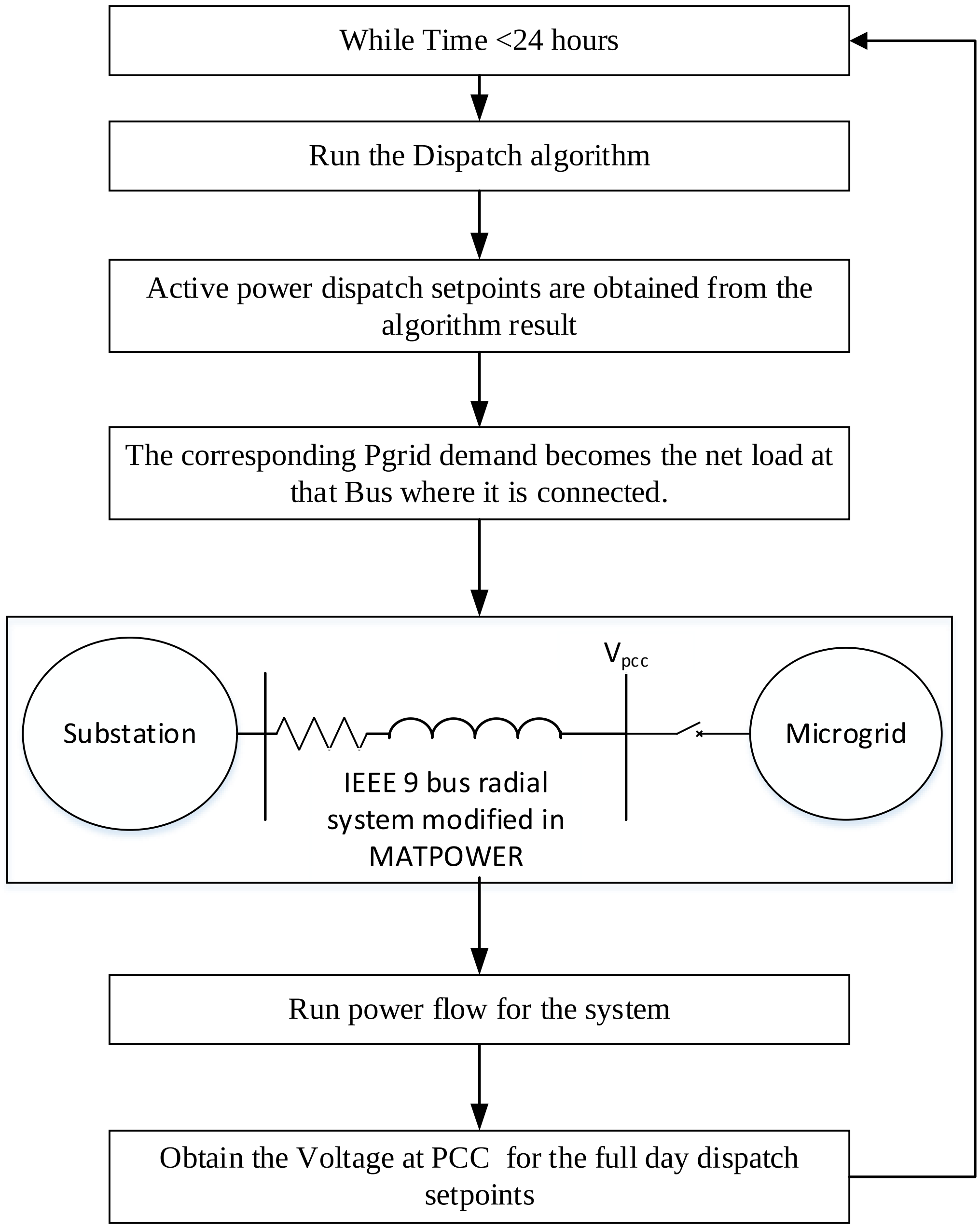

FIGURE 5.2: Voltage calculation at PCC flowchart

5.3. Switching Between Predictive Optimization and Rule Based Dispatch

The microgrid could be operating in off grid or grid connected mode of operation. If this status is available then the dispatch algorithm can be switched between the Rule Based and Predictive Optimization based algorithms. A simple case was taken to show the above operation, where initially the Rule Based algorithm was active and then based on the microgrid status the dispatch values was switched to the Predictive Optimization algorithm values. Two cases were considered for this switching. The first case was based on an external status signal and instantly the dispatch values were changed. In the second case the dispatch values were changed based based on the voltage deviation at that instant.

Case1:

This case is shown in figure 5.3.Initially the system is dispatching based on the Rule Based Dispatch i.e. the system is operating in off grid mode. Then consider that at 7:30 am the status of the microgrid changes to grid connected mode. Now in the figure 5.3 the V_{sw} (green) plot shows the voltage profile during switching and the voltage profile if the dispatch was same as the value obtained from both the algorithms.

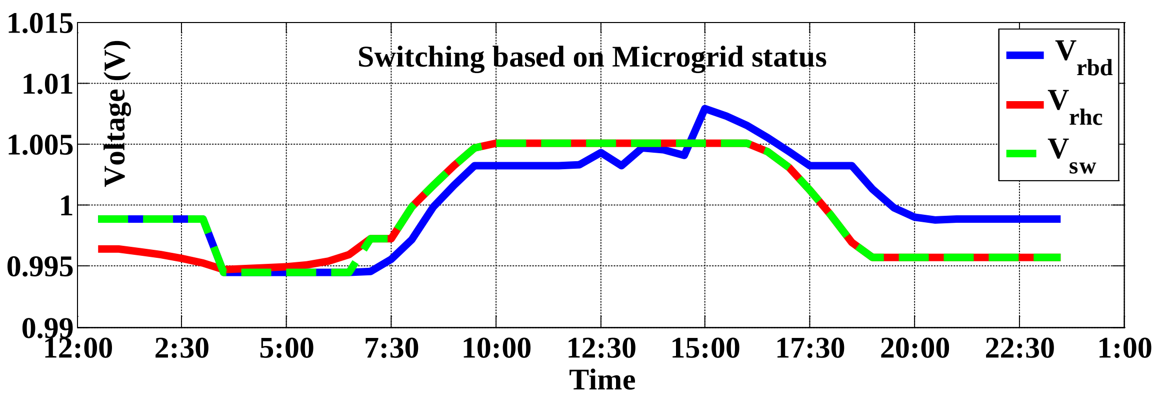

FIGURE 5.3: Switching based on microgrid status

Now as shown in figure 5.4 the voltage waveform (green) moves from the Rule Based to the Predictive Horizon based optimization value which is called V_{rhc} in this figure as mentioned in the beginning .

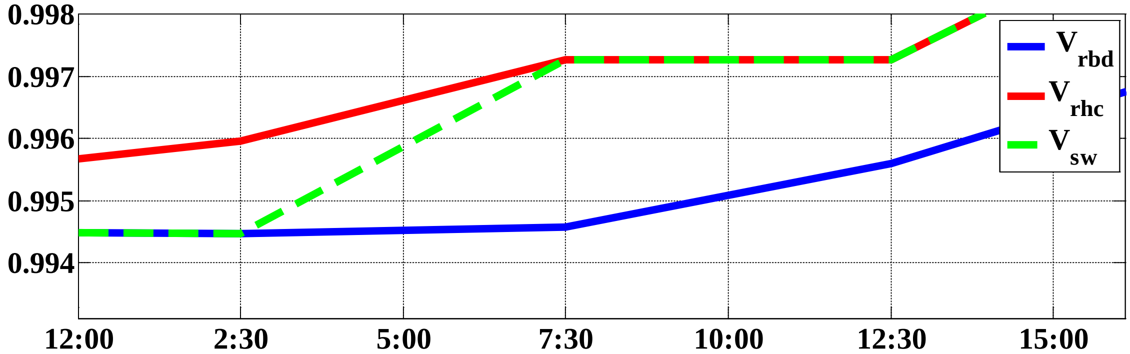

FIGURE 5.4: Voltage during switching based on microgrid status

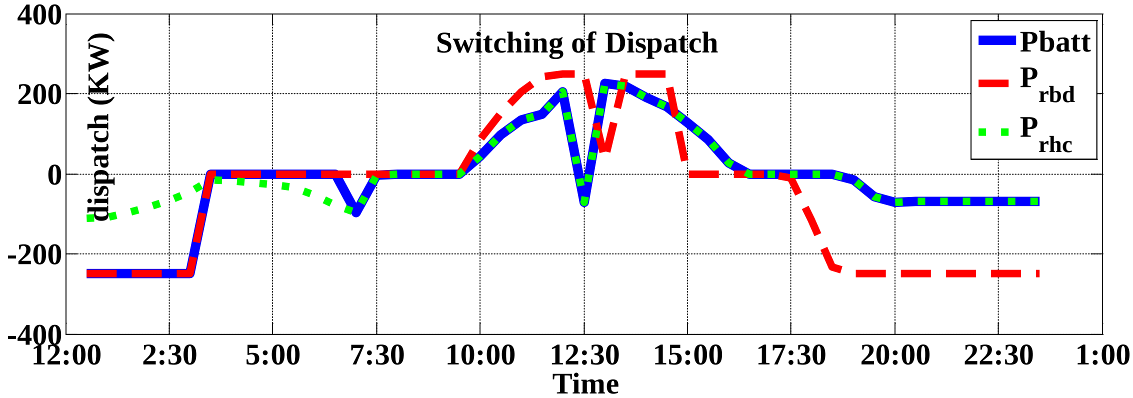

FIGURE 5.5: Dispatch switching based on microgrid status

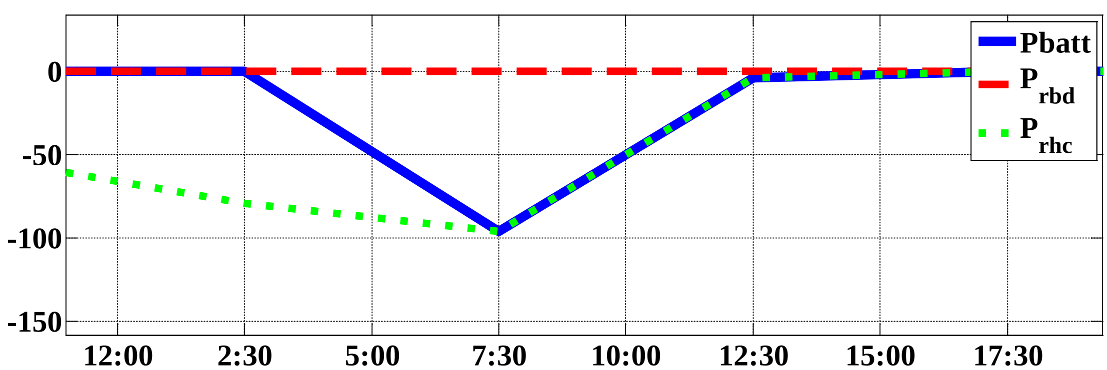

FIGURE 5.6: Dispatch switching from rule based to predictive optimization

The battery dispatch waveforms are shown in figures 5.5 and 5.6. At the instant at 7:30 am the dispatch setpoints change from the rule based dispatch to the predictive optimization dispatch values (P_{rhc}). In figure 5.6 Pbatt (blue) shows the actual battery dispatch.

Case 2:

Now in this case it was considered that the switching between the dispatch algorithms was done based on the voltage deviations at the instant. Here the microgrid is assumed to be running in the grid connected mode initially and then switches to the Rule Based dispatch based on the microgrid status at 15:00. As you can see from the waveform in figure 5.7, the V_{sw} has not switched from the predictive optimization based dispatch at the instant, instead it makes the transfer at 18:30 as shown in figure 5.8 when the deviation of voltage with respect to 1 is less and switches to the rule based dispatch.

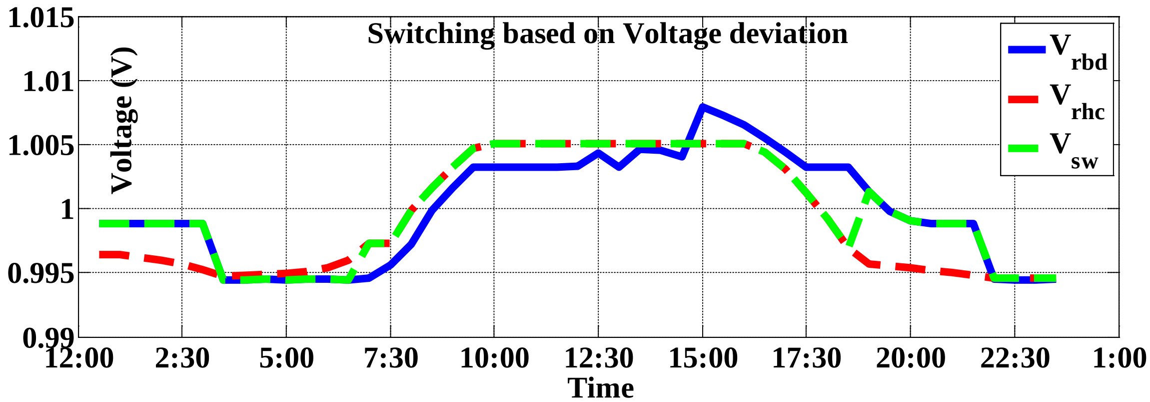

FIGURE 5.7: Switching based on voltage deviation

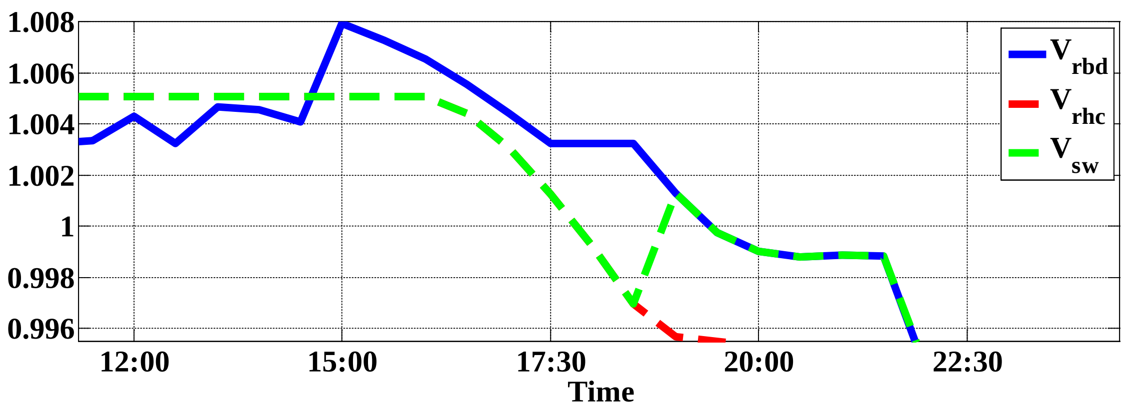

FIGURE 5.8: Voltage switching from predictive to rule based

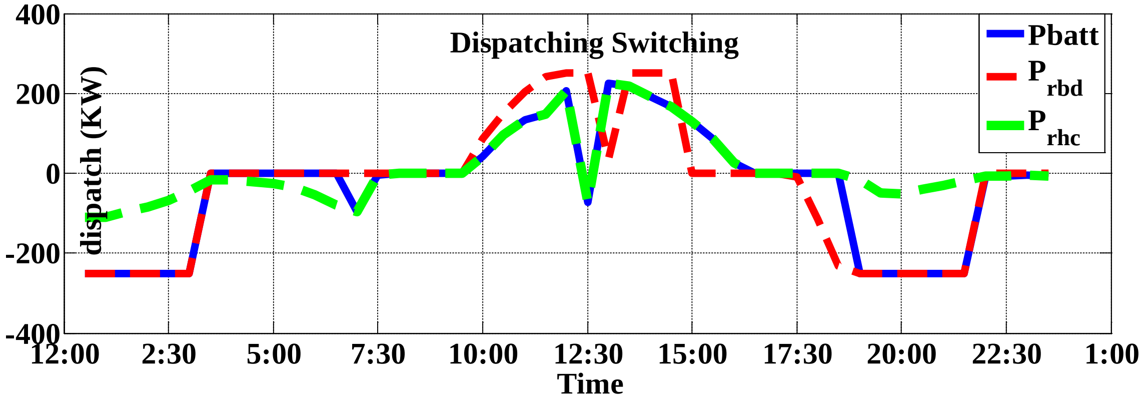

FIGURE 5.9: Dispatch switching based on voltage deviation

The battery dispatch also follows the same path as explained above. The dispatch values follow the Predictive Optimization dispatch till the instant the deviation is less and switches to the Rule Based Dispatch algorithm which is shown in figure 5.9.

5.4. Linking Between Rule Based Dispatch and Predictive Optimization

This algorithm is the extension of the previous case. In the previous case the deviation was only calculated when the switching between the algorithms was considered based on the microgrid status. In this case after first running the Rule Based and Predictive Optimization based algorithms, then the grid demand is obtained and this becomes the load at that bus, which is used to run the power flow for the system and obtain the voltage.at the bus. An assumption made in this case is that the grid power required at the bus obtained from the dispatch algorithm is the net load at that bus. Hence the generation will be equal to the demand in this case. Then the deviation w.r.t to 1 p.u is calculated and based on, which dispatch algorithm results has lesser deviation that battery dispatch is taken as reference. This is done so that the deviation is always within the limits while switching between the algorithms. This process continues all throughout the period of optimization and hence a new final dispatch schedule is obtained for the day which has lesser voltage deviation. The voltage deviations are calculated for the obtained dispatch schedule and then the Voltage deviation index was calculated as done above. The flow chart of the above algorithm is shown below in figure 5.10 and the variables used are explained below.

$V_{rbd}(t)$ is the voltage obtained from rule based dispatch for that instant

$P_{rbd}(t)$ is the battery dispatch obtained from rule based dispatch for that instant

$V_{opt}(t)$ is the voltage obtained from the predictive optimization dispatch for that instant

$P_{opt}(t)$ is the battery dispatch obtained from the predictive optimization for that instant

Rest of variables have been defined in the previous chapters. Initially the PV generation and load data is read from an excel file. Then the initial charge of the battery is initialized. This value is passed to both the algorithms which are defined as two separate functions which returns the Voltage and the battery dispatch. This voltage value is used to calculate the deviation w.r.t 1 p.u and then based on which absolute value difference is smaller, that dispatch value is used and then the charge and state of charge is calculated after which the grid demand is calculated. The battery constraints get calculated within each of the dispatch algorithm functions. Then the net grid demand is used to calculate the voltage based on the flowchart explained in figure 5.2. The power flow is run again in the end to get the new voltage profile for the new dispatch value. The algorithm was run for the system considered above and the voltage profile and battery dispatch was plotted in figures 5.10 and 5.11 and the results are explained below.

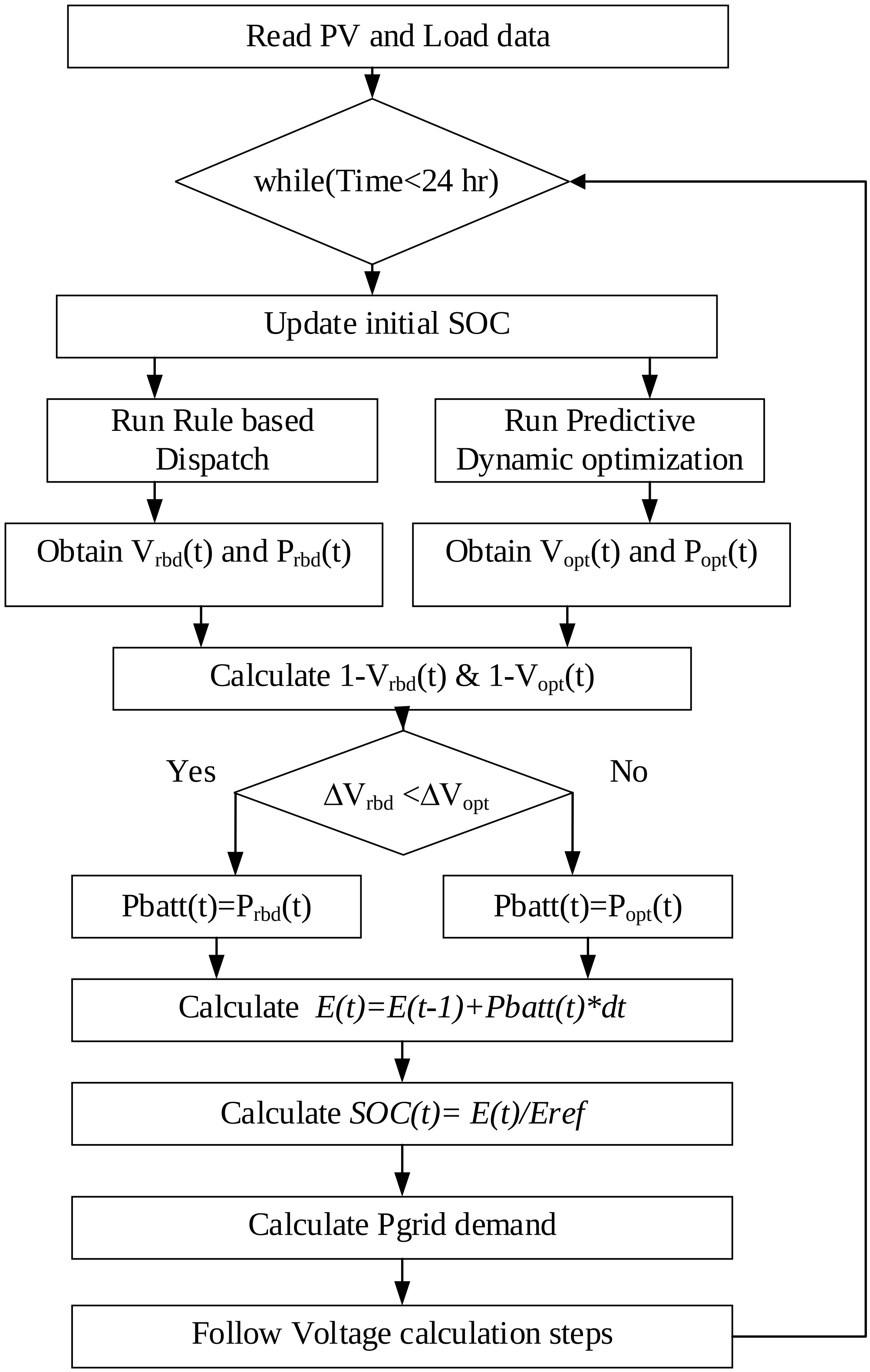

FIGURE 5.10: Linking based on voltage deviation flowchart

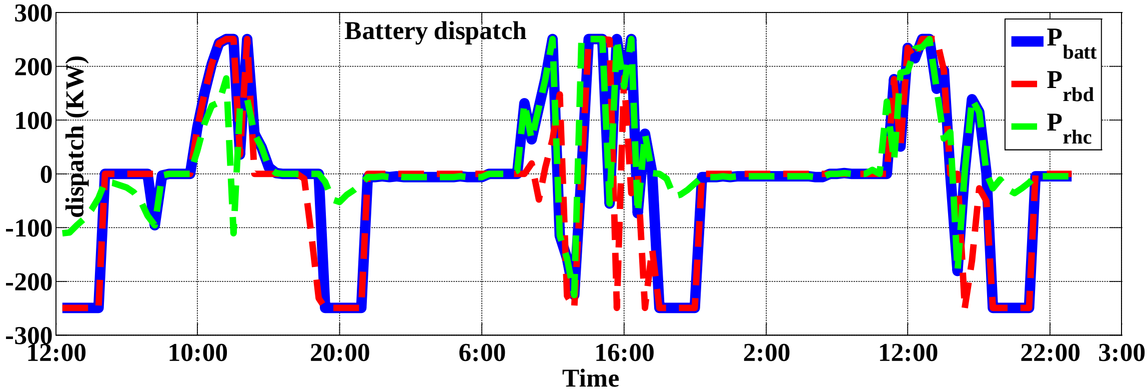

FIGURE 5.11: Battery dispatch based on linked algorithm

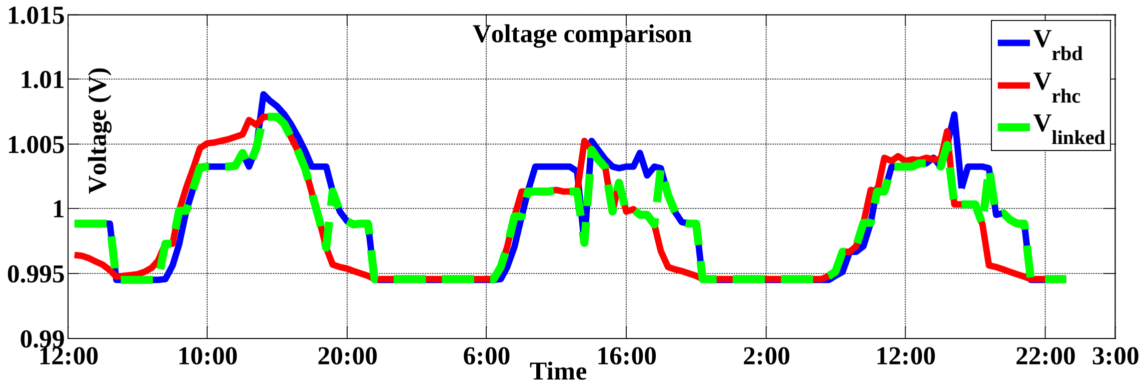

FIGURE 5.12: Final voltage at PCC for linked algorithm

As shown in figure 5.12, the voltage obtained from Rule Based Dispatch is higher initially and hence the deviation w.r.t 1 p.u is smaller and hence the battery dispatch is set to the Rule based dispatch value which is clearly visible in figure 5.11. Here V_{linked} represents the final waveform after switching. So initially the V_{linked} waveform follows the V_{rbd} waveform. Then towards the evening it follows the predictive optimization dispatch. Now every time the SOC is updated and passed to both the algorithms so the final dispatch will be different from the case where it was running in Rule Based Dispatch or in Predictive Dispatch algorithm separately. As this process continues the voltage profile follows the rule based dispatch voltage values and the battery dispatch more or less follows the Predictive Dispatch only during the peak times of the day. The rest of the time it follows the Rule Based dispatch. This algorithm gives minimum voltage deviations when compared with the Rule Based and Predictive Optimization based values. The voltage waveforms for the same 3 day data considered are shown in figure 5.13 when applied to all the algorithms described separately. Finally an index for voltage deviation (V.D.I) was calculated based on the area under the curve of the voltage deviations and an analysis was done if for the same 3 days of data if the Rule Based, Predictive, and Linked algorithms were used what will be the voltage deviation. The results show that the overall voltage deviation is lesser in the case of the linked algorithm when compared with the predictive optimization and the rule based dispatch running separately.

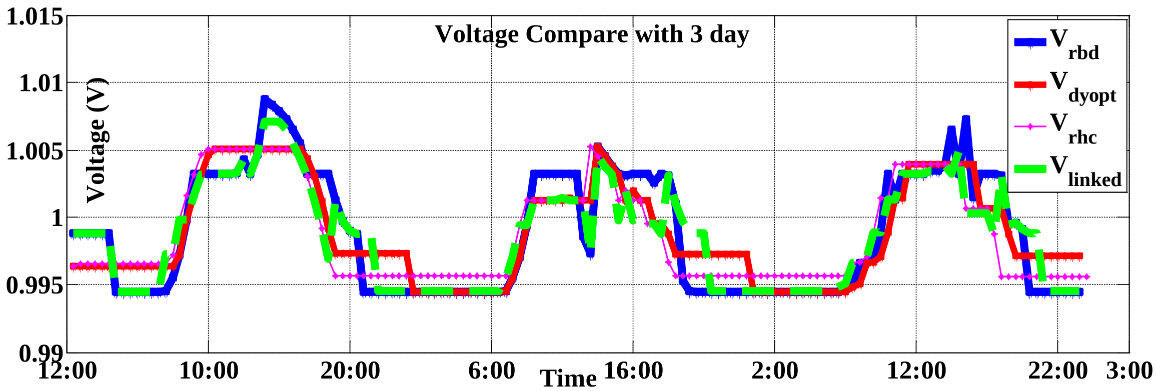

FIGURE 5.13: Comparison of voltage deviation for 3 day data

Table 5.1: 3 days-voltage and cost comparison

3 Day dispatch schedule	Rule Based Dispatch	Fixed Horizon Optimization	Predictive Optimization	Linked Algorithm
VDI	0.59405	0.4934	0.5054	0.49635
COST ($)	1710.8	1712.4	1612.5	1711.2

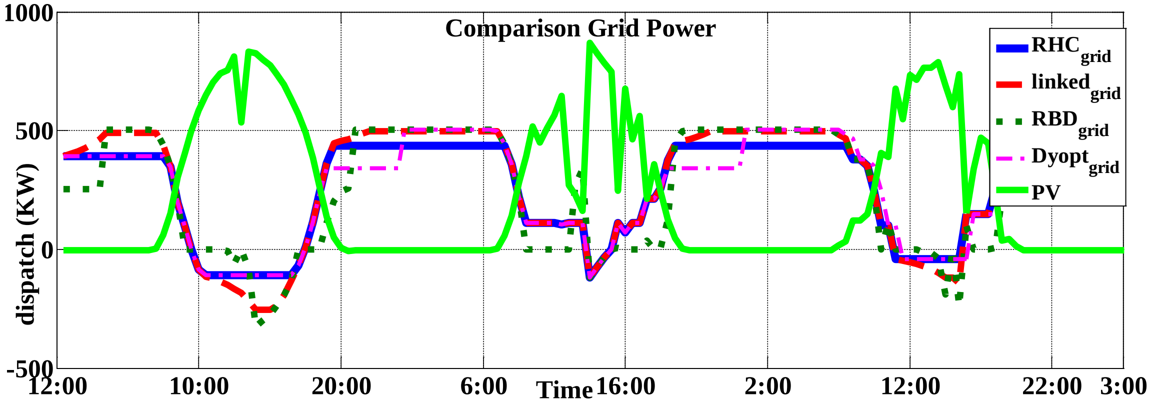

FIGURE 5.14: Comparison of grid power for 3 days

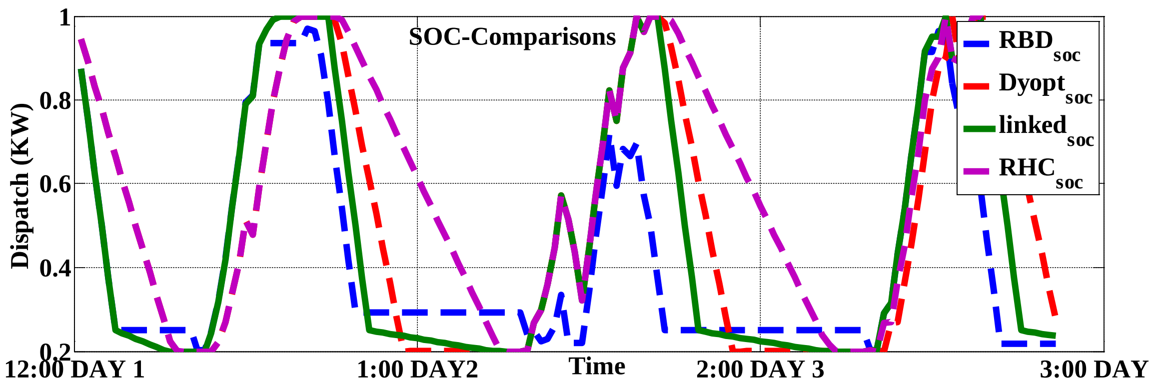

FIGURE 5.15: Comparison of SOC for 3 days

A cost analysis was done similar to the comparison done in chapter 4 between Fixed Horizon and Predictive Horizon Optimization. The power demand from grid is shown in figure 5.14. As explained in the first chapter the positive sign represents power required by the microgrid and the negative sign represents the excess power dispatched back to the grid. Now in the case of Predictive optimization the SOC value does not hit the limit at the end of the day as shown in figure 5.15. But in all the other cases the battery is fully discharged and hence a 10% increase in cost coefficients was applied to quantify the state of charge preserved. This can be seen in figure 5.13 where RHC $_{grid}$ represents the power demand from the Predictive Optimization. Hence the overall cost is less in the case of the

Predictive optimization case. In the above algorithm the battery dispatch can be modified based on the voltage deviation i.e. if the deviation is positive then the battery can be charged and if the deviation is negative then the battery can be discharged to improve the voltage profile and the peaks seen in voltage waveform of above graphs in figure 5.12 can be reduced. This is based on the concept of voltage deviation proportional to the change in the battery dispatch multiplied by a constant. This has not been implemented in this work but can be considered for future works.

5.5. Summary

The Rule Based Dispatch has the highest deviation index when compared to all the algorithms. The Fixed Horizon Optimization and the Linked algorithm has the least voltage deviation index. The main aim of linking the two algorithms was to obtain a framework to switch between maximum resource utilization mode of operation and optimized mode of operation while maintaining PCC voltage within the limits during these modes of operation and at the same time ensure that the battery dispatch is optimal. The linked algorithm also minimizes the voltage deviations at the Point of Common Coupling.

CHAPTER 6 : CONCLUSION AND FUTURE WORKS

6.1. Conclusion

- The Rule based algorithm was developed which can provide the active power reference signals in off grid mode and also if loads cannot be curtailed then in grid connected mode.
- In grid connected mode an optimization algorithm based on dynamic formulation was written to obtain the optimal battery dispatch such that the power taken from grid is minimized.
- It was further modified to include a predictive horizon based approach so that the cost could be minimized based on the future PV generation.
- Then both the Rule based and predictive algorithms were linked to obtain a switching mechanism to transfer the active power reference setpoints when the microgrid transfers from grid connected to off grid mode of operation such that the voltage deviation at the PCC was minimized.

6.2. Future Works

- Analysis of the proposed method for various PV penetration level based on the optimal initial state of charge of the battery.
- Evaluation of the proposed method that links the rule based architecture and optimal dispatch architecture based on the voltage index that minimizes or optimizes the voltage variations.

- Integrated cost and reliability based management of the microgrid for a safe and resilient infrastructure.
- Optimal hybrid active and reactive power management of the microgrid based on battery energy storage inverter.
- Implementation of the proposed architecture on the simulator and evaluating the effect of the controllers.
- Providing a load and renewable energy output prediction algorithm and updating the performance of the rule based and integrated optimization architecture

BIBLIOGRAPHY

[1] S. Teleke, M. E. Baran, S. Bhattacharya, A.Q.Huang,”Rule-based control of battery energy storage for dispatching intermittent renewable sources,”IEEE Trans.Sustain.Energy, vol.1, no. 3, pp 117-124, Oct. 2010.

[2] National Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, “Optimal power flow management for grid connected pv systems with batteries,” IEEE Trans. Sustain. Energy, vol. 2, no. 3, pp. 309–320, Jul. 2011.

[3] H. Tazvinga, X. Xia, J. Zhang “Minimum cost solution of photovoltaic-diesel-battery hybrid power systems for remote consumers” Solar Energy, 96 (2013), pp. 292–299

[4] A. Nottrott, J. Kleissl, and B. Washom, “Storage dispatch optimization for grid-connected combined photovoltaic-battery storage systems,” in Proc. 2012 IEEE Power & Energy Society General Meeting, San Diego, CA, 22–26 July. 2012, pp. 1–7.

[5] L. Xiaoping, D. Ming, H. Jianghong, H. Pingping, and P. Yali, “Dynamic economic dispatch for microgrids including battery energy storage,” in Proc. 2nd IEEE Int Power Electronics for Distributed Generation Systems (PEDG) Symp, 2010, pp. 914–917.

[6] G. W. Chnag, H. J. Lu, H.J. Su, “Short-term Distributed Energy Resource Scheduling for a DC Microgrid” Energy and Power Engineering, 2013, 5, 15-21

[7] L. Bo and M. Shahidehpour, “Short-term scheduling of battery in a grid-connected PV/battery system,” IEEE Trans. Power Syst., vol. 20, no. 2, pp. 1053–1061, May 2005

[8] S. J. Chiang, K. T. Chang, and C. Y. Yen, “Residential photovoltaic energy storage system,” IEEE Trans. Ind. Electron., vol. 45, no. 3, pp. 385–394, Jun. 1998.

[9] C.Wang and M.H.Nehrir, “Power management of a stand-alone wind/ photovoltaic/fuel cell energy system,” *IEEE Trans. Energy Convers.*, vol. 23, no. 3, pp. 957–967, Sep. 2008.

[10] S. Jain and V. Agarwal, “An integrated hybrid power supply for distributed generation application fed by nonconventional energy sources,” *IEEE Trans. Energy Convers.*, vol. 23, no. 2, pp. 622–634, Jun. 2008

[11] Matlab,” Interior-point-convex quadprog algorithm, “available at: http://www.mathworks.com/help/optim/ug/quadratic-programming-algorithms.html#bsqspm.

[12] http://www.eng.newcastle.edu.au/eecs/cdsc/books/cce/Slides/RecedingHorizonControl.pdf

[13] Xiaonan Wang, Ahmet Palazoglu, Nael H. El-Farra, "Operational optimization and Demand response of hybrid renewable energy systems", *Elsevier Applied Energy143 (2015) 324-335.*

[14] Y. Levron, J. M. Guerrero, and Y. Beck, "Optimal power flow in microgrids with energy storage," IEEE Trans. Power Syst., vol. 28, no. 3, pp. 3226–3234, Aug. 2013.

[15] J. C.Vasquez, J.M. Guerrero, J.Miret,M. Castilla, and L.G. deVicuña "Hierarchical control of intelligent microgrids," *IEEE Ind. Electron.Mag.*, vol. 4, pp. 23–29, Dec. 2010.

[16] A. G. Tsikalakis andN.D.Hatziargyriou, "Centralized control for optimizing microgrids operation," *IEEE Trans. Energy Conv.*, vol. 23, no.2, pp. 241–248, Mar. 2008.

[17] A. Hoke, A. Brissette, S. Chandler, A. Pratt, and D. Maksimovic,"Lookahead economic dispatch of microgrids with energy storage using linear programming," in Proc. of the 1st IEEE Conf. on Technologies for Sustainability, 2013, pp. 154–161.

[18] P. M. Corrigan and G. T. Heydt, "Optimized dispatch of a residential solar energy system," in North Amer. Power Symp., Sep. 2007, pp. 183–188.

[19] M. D. Hopkins, A. Pahwa, and T. Easton, "Intelligent dispatch for distributed renewable resources," IEEE Trans. Smart Grid, vol. 3, no. 2, pp. 1047–1054, Jun. 2012.

[20] D. Lu, H. Kanchev, F. Colas, V. Lazarov, and B. Francois, "Energy management and operational planning of a microgrid with a PV-based active generator for smart grid applications," IEEE Trans. Ind. Electron., vol. 58, no. 10, pp. 4583–4592, Oct. 2011.

[21] R. Rigo-Mariani, B. Sareni, X. Roboam, and C. Turpin, "Optimal power dispatching strategies in smart-microgrids with storage",Renewable and Sustainable Energy Reviews, vol. 40pp. 649_658, 2014.

[22] Mahat P, Escribano Jiménez J, Rodriguez Moldes E, et al. A micro-grid battery storage management. IEEE PES general meeting; 2013.

[23] Mattingley JE, Wang Y, Boyd S (2010) Code generation for receding horizon control. In: Proceedings IEEE multi-conference on systems and control, September 2010, pp 985–99

[24] Operations Research Models and Methods Paul A. Jensen and Jonathan F. Bard

[25] Solving Optimization Problems using the Matlab Optimization Toolbox - a Tutorial TU-Ilmenau, Fakultät für Mathematik und Naturwissenschaften Dr. Abebe Geletu December 13, 2007.

[26] J. T. B. A. Kessels, M. Koot, B. de Jager, P. P. J. Van Den Bosch, N. P. I. Aneke, and D. B. Kok, “Energy management for the electric powernet in vehicles with a conventional drivetrain,” IEEE Trans. Control Syst. Technol., vol. 15, no. 3, pp. 494–505, May 2007

[27] J. T. B. A. Kessels, M. W. T. Koot, P. P. J. Van Den Bosch, and D. B. Kok, “Online energy management for hybrid electric vehicles,” IEEE Trans. Veh. Technol., vol. 57, no. 6, pp. 3428–3440, Nov. 2008

[28] Vehicle Power Management: Modeling, Control and Optimization By Xi Zhang, Chris Mi.

[30] Dynamic Optimization Joshua Wilde, revised by Isabel Tecu, Takeshi Suzuki and María José Boccardi August 13, 2013

[31] Economic Dispatch and Introduction to Optimisation by Daniel Kirschen

[32] M. Ceraolo, A. Di Donato, and G. Franceschi, “A general approach to energy optimization of hybrid electric vehicles,” IEEE Trans. Veh. Technol., vol. 57, no. 3, pp. 1433–1441, May 2008

[33] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna, M. Castilla, "Hierarchical Control of Droop-Controlled AC and DC Microgrids—A General Approach Toward Standardization," IEEE Trans. on Industrial Electronics, vol. 58, no. 1, pp. 158 - 172 , Jan 2011.

[34] H. Kanchev, Di Lu, F. Colas, V. Lazarov, B. Francois, "Energy Management and Operational Planning of a Microgrid With a PV-Based Active Generator for Smart Grid Applications," IEEE Trans. On Industrial Electronics, vol. 58, no. 10, pp. 4583 - 4592 , Oct 2011.

[35] Jong-Yul Kim et al., "Cooperative Control Strategy of Energy Storage System and Microsources for Stabilizing the Microgrid during Islanded Operation," IEEE Trans. Power Electronics, vol. 25, no. 12, pp. 3037-3048, Dec. 2010.

[36] D. E. Olivares, C. A. Canizares, and M. Kazerani, “A centralized optimal energy management system for microgrids,” in Proc. Power and Energy Society General Meeting, Jul. 2011.

[37] A. G. Tsikalakis and N. D. Hatziargyriou, “Centralized control for optimizing microgrids operation,” IEEE Trans. Energy Convers., vol. 23, no. 1, pp. 241–248, Mar. 2008.

[38] N. Hatziargyriou, G. Contaxis, M. Matos, J. A. P. Lopes, G. Kariniotakis, D. Mayer, J. Halliday, G. Dutton, P. Dokopoulos, A. Bakirtzis, J. Stefanakis, A. Gigantidou, P. O’Donnell, D. McCoy, M. J. Fernandes, J. M. S. Cotrim, and A. P. Figueira, “Energy management and control of island power systems with increased penetration from renewable sources,” in Proc. IEEE-PES Winter Meeting, Jan. 2002, vol. 1, pp. 335–339.

[39] S. Chakraborty and M. G. Simoes, “PV-microgrid operational cost minimization by neural forecasting and heuristic optimization,” in Proc. IEEE Ind. App. Soc. Annual Meeting, Oct. 2008.

[40] H. Vahedi, R. Noroozian, and S. H. Hosseini, “Optimal management of MicroGrid using differential evolution approach,” in Proc. 7th Int. Conf. European Energy Market (EEM), Jun. 2010

[41] N. D. Hatziargyriou, A. Dimeas, A. G. Tsikalakis, J. A. P. Lopes, G. Karniotakis, and J. Oyarzabal, “Management of microgrids in market environment,” in Proc. Int. Conf. Future Power Systems, Nov. 2005.

[42] T. Logenthiran, D. Srinivasan, and D. Wong, “Multi-agent coordination for DER in MicroGrid,” in Proc. IEEE Int. Conf. Sustainable Energy Technologies (ICSET), Nov. 2008.

[43] J. Oyarzabal, J. Jimeno, J. Ruela, A. Engler, and C. Hardt, “Agent based microgrid management system,” in Proc. Int. Conf. Future Power Systems, Nov. 2005.

[44] L. Xie and M. D. Ilic, “Model predictive dispatch in electric energy systems with intermittent resources,” in Proc. IEEE Int. Conf. Syst. Man Cybern., Singapore, Oct. 2008, pp. 42–47

[45] E. Perez, H. Beltran, N. Aparicio and P. Rodriguez, “Predictive Power ´ Control for PV Plants With Energy Storage”, IEEE Trans. On Sustainable Energy, Vol. PP, n. 99, 2012, pp.1-9

[46] J. Patino, A. Marquez, and J. Espinosa, “An economic MPC ´ approach for a microgrid energy management system,” in IEEE PES Transmission & Distribution Conference and Exposition (T&D-LA), 2014.

[47] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, et al., “A microgrid energy management system based on the rolling horizon strategy,” IEEE Trans. Smart Grid, vol. 4, no. 2, pp. 996–1006, Jun. 2013

[48] A. Parisio and L. Glielmo, “Stochastic model predictive control for economic/environmental operation management of microgrids,” in Proc. IEEE Eur. Control Conf. (ECC), Zürich, Switzerland, 2013.

[49] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, “Demand-side management via distributed energy generation and storage optimization,” IEEE Trans. Smart Grid, 2013

[50] A. Parisio and L. Glielmo, “A mixed integer linear formulation for microgrid economic scheduling,” in Proc. of 2nd Intl. Conf. Smart Grid Communications, Brussels, Belgium, October 17-20, 2011

[51] K. Chandy, S. Low, U. Topcu, and H. Xu, “A simple optimal power flow model with energy storage,” in Decision and Control (CDC), 2010 49th IEEE Conference on, Dec. 2010, pp. 1051 –1057.

APPENDIX A: MATLAB CODES

Rule Based Dispatch Algorithm Code :

```
global inc pv_gen Pq_demand
n=1;
count=0;
tot_loss=0;
inc=1; %initial sample starting from 2 because zero cannot be used as
index.
%dispatch(1)=0;
%batt_dis(1)=0;
%E=zeros(15,0);
Pgen=zeros(282,1);
% E(1)=75;    % initial soc value
% E_f(1)=75;
% Eref=75;    % max soc value
%E(1)=1000;    % initial soc value
%E_f(1)=1000;
%soc(1)=1;
Eref=1000;    % max soc value
E0=1000;
d=0;
dt=30/60;     % time step
%global Gen1 Gen2 Gen3
%m=xlsread('trial.xls','D2:D14');
%t=xlsread('full_day.xls','M3:M33740') % full day net Pload-PVgen
 %pv=xlsread('full_day.xls','K3:K33740'); % full day PV generation
 P_pv = xlsread('final_sc_data1.xlsx','J2:J193') % reading pv data 99
197
 Pload  = xlsread('final_sc_data1.xlsx','K2:K193')% reading load data
99 197
 net_load = xlsread('final_sc_data1.xlsx','I2:I193')% reading netload
data ie Pload-P_pv 99 197
soc_ll=0.20;    %soc lower limit
soc_ul=1;       %soc upper limit
 % value used to get 5  min data increments from the full day excel
data
 m(inc,1)=Pload(n);     % m stores the 5 min load data
 pv_gen(inc,1)=P_pv(n); % this is the PV generation schedule in 5 min
interval

% Calculate the Pload - Ppv
%this becomes the load that must be satisfied by the battery taking
this sign
 % +ve means that value to be discharged from batt
 % -ve means that value to be charged from PV, as PV is excess.

  % if (m(inc)<0)     % charging value
  %if (net battery demand) <0)
  %Use the dispatch to charge the battery
  %else   % discharge value
  %Discharge the battery.
  % end
```

```
    % Pess needs to be between -25MW and +25 MW
    Pess(inc,1) =X_new(inc,1);
    %E(inc,1)=E(inc-1,1)+Pess(inc,1)*dt ;  % calculating the Echarge
value
    E(inc,1)=E0+Pess(inc,1)*dt
    soc(inc,1)= E(inc,1)/Eref ;            % SOC values to get into
condition checks.

    % 1st condition if the soc should be between upper and lower limits

    %2nd condition if the soc should be equal to lower limit or less
than the lower limits

      % 3rd condition if the soc should be equal to upper limit or
greater than the upper limit

    % dispatch variable is the net Pstorage or Pdischarge value,-ve or
+ve
    % +ve dispatch means charging value
    % -ve dispatch means discharging value
    switch d     % switching based on the conditions above based on d
variable
        case 1
            if (P_batt(inc,1)<0)
   % 1st condition satisfied then the available Pes is dispatched
            end
        case 2
            if (P_batt(inc,1)>0)
 % 2nd condition  % means it has hit the lower limit and cannot
discharge anymore, dispatch =0;

            else if (P_batt(inc,1)<0)
  % means it can charge since net Pess is negative so dispatch= Pcharge
                    batt_dis(inc,1)=0;
                end
            end
        case 3                                % 3rd condition
            if (P_batt(inc,1)>0)
  % means battery has hit upper limit so it can only discharge, and no
charging
                dispatch(inc,1)=X_new(inc,1);        %since m(inc)> 0,
it can discharge
                batt_dis(inc,1)=X_new(inc,1);
            else if (P_batt(inc,1)<=0)
 % no charge possible since upper limit hit and since m(inc)<0 it is a
charge value
                    dispatch(inc,1)= 0;             % but since upper
limit no charge

  %Pgen(inc,1)=X_new(inc,1)    % if it hit limit then the available
value can be given to grid.
                end
            end
        case 4
```

```
           % if (m(inc)<0)
             dispatch(inc,1)=0;
            %end
        case 5
            %if ( m(inc)>0)
                dispatch(inc,1)=0;
            %end

    end

    E(inc,1)=E0+ dispatch(inc,1)*dt ;   %final Echarge values
calculated with the dispatch variable
    soc(inc,1)= E(inc,1)/Eref ;                 % final soc value
    Pq_demand(inc,1)=net_load(inc,1)+dispatch(inc,1); % final net
Pg=dispatch+Pload-Pgen
    E0=E(inc,1);



%result=runpf('case_thesis2.m');
 % result=runpf('case_dist_9bus.m');
  result=runpf('case_dist_9bus.m');
   %result=runpf('case_dist.m');

  v(n,1)=(result.bus(10,8)) %10
  loss(n,1)=(sum(get_losses(result)));
 tot_loss= loss(n,1)+ tot_loss;
  %count(n,1)=n;
n=n+1;
inc=inc+1;

 %T =
table(E,soc,dispatch,Pgrid,lambda,P_pvbatt,dispatch,pv_gen,P_batt,P_bat
t1); %writing the required values to table.
 %filename = 'batt_opf_new_ed_23_5_1.xlsx';
%writetable(T,filename);
while(n<144)       % reading values till less than sample size
%m=xlsread('trial.xls','D2:D14');

   % value used to get 5  min data increments from the full day excel
data
 m(inc,1)=Pload(inc);     % n m stores the 5 min data not required here
since it is Pload-Ppv
 pv_gen(inc,1)=P_pv(inc); % n this is the PV generation schedule in 5
min interval



P_batt(inc,1) = m(inc,1)-pv_gen(inc,1); %this becomes the load that
must be satisfied by the battery taking this sign
 % +ve means that value to be discharged from batt
 % -ve means that value to be charged from PV, as PV is excess.

    E(inc,1)=E0+ dispatch(inc,1)*dt ;   %final Echarge values
calculated with the dispatch variable
    soc(inc,1)= E(inc,1)/Eref ;                 % final soc value
```

```
    Pq_demand(inc,1)=net_load(inc,1)+dispatch(inc,1); % final net
Pg=dispatch+Pload-Pgen
    E0=E(inc,1);

%result=runpf('case_thesis2.m');
 % result=runpf('case_dist_9bus.m');
 result=runpf('case_dist_9bus.m');
%  result=runpf('case_dist.m');

  v(n,1)=(result.bus(10,8))  %10
  loss(n,1)=(sum(get_losses(result)));
 tot_loss= loss(n,1)+ tot_loss;
  %count(n,1)=n;
n=n+1;
inc=inc+1;

 %T = %filename = 'batt_opf_new_ed_23_5_1.xlsx';
%writetable(T,filename);
end
% plot(v,'-*b');
 P_pv = pv_gen;         %xlsread('opt_file.xlsx','A2:A282') % reading
pv data
 pgrid =Pq_demand;      %xlsread('rule_dispatch.xlsx','E2:E282') %
reading load data
 pbatt=dispatch;
%xlsread('rule_dispatch.xlsx','C2:C282')
 soc =soc;                  %xlsread('rule_dispatch.xlsx','B2:B282')


subplot(3,1,1);
plot(pgrid,'-b');
hold on;
plot(pbatt,'--r');
hold on;
plot(P_pv,'-g');
% Add labels
xlabel(' sample');
ylabel('dispatch');
title('PV and Pgrid and Pbess');

subplot(3,1,2);
plot(soc,'-b');
xlabel(' sample');
ylabel('SOC');
title('SOC');
% plot(pv_gen,inc);
%end
subplot(3,1,3);
plot(v,'-*b');
xlabel(' sample');
ylabel('Voltage');
title('Voltage');
```

Fixed Horizon Optimization Code :

```
% Dynamic optimization
P_pv =xlsread('30_min_data_new.xlsx','E2:E50'); % reading pv data 99
197
 Pload  =xlsread('30_min_data_new.xlsx','F2:F50');% reading load data
99 197
 plot_pv= xlsread('30_min_data_new.xlsx','E2:E50');

 %Pload  =xlsread('opt_file.xlsx','D2:D290')
 E0 = 1000;  % initial charge in battery
 %Emin = 20 MWhr
 %Emax = 75 MWhr
 %dt=0.0833; % timestep (5/60) 5 min interval for 24 hours
 dt=0.5;
a1=0.1;b1=12.6;c1=8;a2=0.0;b2=0.0;c2=0;% coefficients of cost function
% coefficients of cost fucntion
const=[a1 b1 c1 a2 b2 c2];                % defining the constants to
be used in code
%Pload=[30;30;30;30;30;30;30;30;30;30];
%P_pv=[0;0;40;50;40;40;30;20;10;10];
%P_pv=[-0;0;0;0;0;0;0;0;0;-0];

%N=288 ;        % no of samples
N=48;
x0 = [zeros(size(1:N));-25*ones(size(N+1:N))]; % initial solution
assumption x0 but quadprog does notuse it.

% constraints formation

% Define_constraints;
%bounds LB<=x<=UB

% ---- 2 bound constraints --- 1 equality constraint ----- 1 inequality
constraint
 % formation of matrix as shown in report

% ---equality constraint---
 % formation of matrix as shown in report

%---inequality constraint---
% formation of matrix as shown in report
% formation of matrix as shown in report
% objective function
% Now since my solution consists of Pgrid and Pbatt for the full
horizon,
% it will have 2N variables
X = sym('x',[3*N,1]);  % reates an 2*N by 1 symbolic matrix filled with
automatically generated elements

E_init = sym('E_init');
c_o = sym ({'a1','b1','c1','a2','b2','c2'}.','r'); % creating symbolic
constants for the cost function, the constants are defined above
% the objective function is the sum of Pbatt and Pgrid cost function
```

```
C= (a1.*((X(1:N).^2))) + (b1.*(X(1:N)))+ c1   %(a2.*((X(N+1:2*N).^2)))
+ (b2.*(X(N+1:2*N)))+ c2
%C=((c_o(1)).*((X(1:N))))+c_o(2)
% Now forming the objective equation for the full horizon,so it will be
the
% above sum over the full horizon of solution variables
tot_obj = sum(C)

tot_obj = subs(tot_obj,[c_o],[const']); %substituting the constants
into the equations
% this function creates a function from the above formed equation.
matlabFunction(tot_obj,'vars',{X},'file','obj_test');
%F = double(gradient(tot_obj,X)); %the gradient of the function is
found
fsym = gradient(tot_obj,X); %the gradient of the function is found
f = double(subs(fsym,X,zeros(size(X)))); % now to get the constants,the
X()values are substituted zero.
H = 0.5.*double(hessian(tot_obj,X)); % the hessian of the objective
matrix is found
qpoptions = optimset('Algorithm','interior-point-
convex','Disp','iter');
%qpoptions = optimset('Algorithm','active-set','Disp','iter');
%options = optimset('MaxFunEvals',Inf,'MaxIter',5000,...
 %    'Algorithm','interior-point','Display','iter');
tic
[out,fval3,exitflag,output,lambda] =
quadprog(H,f,A,b,Aeq,beq,LB,UB,x0,qpoptions);
%[out,fval3,exitflag,output,lambda] = linprog(F,A,b,Aeq,beq,LB,UB,x0);
%[out, fval3] = fmincon(@obj_test,x0,A,b,Aeq,beq,LB,UB,[],options);
toc
out
fval3
% exitflag
output
lambda
%plotResults( out, N);
a=(1:N);
b=(N+1:2*N);
pgrid=out(1:N);
P =out(N+1:2*N);
Pbatt=out(2*N+1:3*N);
%soc= ((E0)+(ot*pbatt))/75 ;
soc= ((E0)+(A_new1*out))/1000 ;
%  T = table(out); %writing the required values to table.
%   filename = 'optimization_results_7_6.xlsx';
%  writetable(T,filename);
subplot(2,1,1);
plot(pgrid,'-*b');
hold on;
plot(P,'--*r');
hold on;
plot(P_pv,'-*g');
hold on;
plot(Pbatt,'-*c');
% Add labels
xlabel(' sample');
ylabel('dispatch');
```

```
title('PV and Pgrid and Pbess');

subplot(2,1,2);
plot(soc,'-*b');
xlabel(' sample');
ylabel('SOC');
title('SOC');
```

Predictive Horizon Optimization Code :

```
% Predictive Horizon Optimization

 P_pv =xlsread('final_sc_data1.xlsx','J2:J193'); % reading pv data 99
197
 Pload =xlsread('final_sc_data1.xlsx','K2:K193');% reading load data 99
197
 plot_pv= xlsread('final_sc_data1.xlsx','J2:J193');% 50
 global final_grid i
 %Pload  =xlsread('opt_file.xlsx','D2:D290')
 E0 = 1000;  % initial charge in battery
 i=1;
 days=0;
 s=0;
 while(days<3)
 tot_loss=0;
 %Emin = 20 MWhr
 %Emax = 75 MWhr
 %dt=0.0833; % timestep (5/60) 5 min interval for 24 hours
 dt=0.5;
a1=0.1;b1=12.6;c1=8;a2=0.0;b2=0.0;c2=0;% coefficients of cost function
% coefficients of cost fucntion
const=[a1 b1 c1 a2 b2 c2];                % defining the constants to
be used in code
%Pload=[30;30;30;30;30;30;30;30;30;30];
%P_pv=[0;0;40;50;40;40;30;20;10;10];
%P_pv=[-0;0;0;0;0;0;0;0;0;-0];

%N=288 ;        % no of samples
N=48;
%N=96;
x0 = [zeros(size(1:N));-25*ones(size(N+1:N))]; % initial solution
assumption x0 but quadprog does notuse it.

% constraints formation
     % formation of matrix as shown in report

% Define_constraints;
%bounds LB<=x<=UB
     % formation of matrix as shown in report

% ---- 2 bound constraints --- 1 equality constraint ----- 1 inequality
constraint
     % formation of matrix as shown in report
```

```
% ---equality constraint---
     % formation of matrix as shown in report

%---inequality constraint---
    % formation of matrix as shown in report

ot1 = dt.*sparse(zeros(N));    % formation of matrix as shown in report
-this will form Pgrid side
ot2 = dt.*sparse(tril(ones(N))); % formation of matrix as shown in
report - this will from Pbatt side
A_new1 = [ot1 ot1 ot2];
b_new1 = [(1000-E0).*ones(N,1)];   %upper limit
A_new2 = [ot1 ot1 -ot2];
b_new2 = [(-200+E0).*ones(N,1)]; %lower limit
A = [A_new1;A_new2];
b = [b_new1;b_new2];

% objective function
% Now since my solution consists of Pgrid and Pbatt for the full
horizon,
% it will have 2N variables
X = sym('x',[3*N,1]);  % creates an 2*N by 1 symbolic matrix filled
with automatically generated elements

E_init = sym('E_init');
c_o = sym ({'a1','b1','c1','a2','b2','c2'}.','r'); % creating symbolic
constants for the cost function, the constants are defined above
% the objective function is the sum of Pbatt and Pgrid cost function
C= (a1.*((X(1:N).^2))) + (b1.*(X(1:N)))+ c1   %(a2.*((X(N+1:2*N).^2)))
+ (b2.*(X(N+1:2*N)))+ c2
%C=((c_o(1)).*((X(1:N))))+c_o(2)
% Now forming the objective equation for the full horizon,so it will be
the
% above sum over the full horizon of solution variables
tot_obj = sum(C)

tot_obj = subs(tot_obj,[c_o],[const']); %substituting the constants
into the equations
% this function creates a function from the above formed equation.
matlabFunction(tot_obj,'vars',{X},'file','obj_test');
%F = double(gradient(tot_obj,X)); %the gradient of the function is
found
fsym = gradient(tot_obj,X); %the gradient of the function is found
f = double(subs(fsym,X,zeros(size(X)))); % now to get the constants,the
X()values are substituted zero.
H = 0.5.*double(hessian(tot_obj,X)); % the hessian of the objective
matrix is found
qpoptions = optimset('Algorithm','interior-point-
convex','Disp','iter');
%qpoptions = optimset('Algorithm','active-set','Disp','iter');
%options = optimset('MaxFunEvals',Inf,'MaxIter',5000,...
 %   'Algorithm','interior-point','Display','iter');
tic
```

```
[out,fval3,exitflag,output,lambda] =
quadprog(H,f,A,b,Aeq,beq,LB,UB,x0,qpoptions);
%[out,fval3,exitflag,output,lambda] = linprog(F,A,b,Aeq,beq,LB,UB,x0);
%[out, fval3] = fmincon(@obj_test,x0,A,b,Aeq,beq,LB,UB,[],options);
toc
out
fval3
% exitflag
final_out(i,1)=out(2*N+1);
p(i,1)=out(N+1);
%plot_s=out(97:97+N-1);
%final_out(i,1)=out(193);
final_grid1(i,1)=out(1);
final_grid(i,1)=out(1);
E(i,1)= E0+ (final_out(i,1)*dt);
E0=E(i,1);
%result=runpf('case_dist_9bus.m');
result=runpf('case_dist_9bus_dynopt.m');

loss(i,1)=(sum(get_losses(result)));
 tot_loss= loss(i,1)+ tot_loss;
  v(i,1)=(result.bus(10,8));
i=i+1;

% output
% lambda
%plotResults( out, N);
n=1;
while (n<48)

N=48;
x0 = [zeros(size(1:N));-25*ones(size(N+1:N))]; % initial solution
assumption x0 but quadprog does notuse it.

% constraints formation

% Define_constraints;
%bounds LB<=x<=UB
      % formation of matrix as shown in report

% ---- 2 bound constraints --- 1 equality constraint ----- 1 inequality
constraint
     % formation of matrix as shown in report

% ---equality constraint---
     % formation of matrix as shown in report

%---inequality constraint---
    % formation of matrix as shown in report

% objective function
% Now since my solution consists of Pgrid and Pbatt for the full
horizon,
% it will have 2N variables
```

```
X = sym('x',[3*N,1]);  % reates an 2*N by 1 symbolic matrix filled with
automatically generated elements

E_init = sym('E_init');
c_o = sym ({'a1','b1','c1','a2','b2','c2'}.','r'); % creating symbolic
constants for the cost function, the constants are defined above
% the objective function is the sum of Pbatt and Pgrid cost function
C= (a1.*((X(1:N).^2))) + (b1.*(X(1:N)))+ c1   %(a2.*((X(N+1:2*N).^2)))
+ (b2.*(X(N+1:2*N)))+ c2
% Now forming the objective equation for the full horizon,so it will be
the
% above sum over the full horizon of solution variables
tot_obj = sum(C)

tot_obj = subs(tot_obj,[c_o],[const']); %substituting the constants
into the equations
% this function creates a function from the above formed equation.
matlabFunction(tot_obj,'vars',{X},'file','obj_test');
%F = double(gradient(tot_obj,X)); %the gradient of the function is
found
fsym = gradient(tot_obj,X); %the gradient of the function is found
f = double(subs(fsym,X,zeros(size(X)))); % now to get the constants,the
X()values are substituted zero.
H = 0.5.*double(hessian(tot_obj,X)); % the hessian of the objective
matrix is found
qpoptions = optimset('Algorithm','interior-point-
convex','Disp','iter');
%qpoptions = optimset('Algorithm','active-set','Disp','iter');
%options = optimset('MaxFunEvals',Inf,'MaxIter',5000,...
 %   'Algorithm','interior-point','Display','iter');
tic
[out,fval3,exitflag,output,lambda] =
quadprog(H,f,A,b,Aeq,beq,LB,UB,x0,qpoptions);
%[out,fval3,exitflag,output,lambda] = linprog(F,A,b,Aeq,beq,LB,UB,x0);
%[out, fval3] = fmincon(@obj_test,x0,A,b,Aeq,beq,LB,UB,[],options);
toc
%out
%fval3
% exitflag
final_out(i,1)=out(2*N+1);
%plot_series=out(97:97+N-1)
final_grid1(i,1)=out(1);
final_grid(i,1)=out(1);
p(i,1)=out(N+1);
E(i,1)= E0+ (final_out(i,1)*dt);
E0=E(i,1);
%result=runpf('case_dist_9bus.m');
result=runpf('case_dist_9bus_dynopt.m');
  v(i,1)=(result.bus(10,8))
  loss(i,1)=(sum(get_losses(result)));
 tot_loss= loss(i,1)+ tot_loss;
i=i+1;
n=n+1;
pgrid=final_grid1;
pbatt=final_out;
soc=(1/1000).* E ;
```

```
end
days=days+1;
if (days==1)
s=48;
elseif (days==2)
    s=96;
end
n=0;
 end
%soc= ((E0)+(A_new1*out))/75 ;
%  T = table(out); %writing the required values to table.
%   filename = 'optimization_results_7_6.xlsx';
%  writetable(T,filename);
%plot(v,'-*b');
%hold on
subplot(3,1,1);
 plot(pgrid,'-b');
 hold on;
plot(pbatt,'--r');
hold on;
plot(plot_pv,'-g');
% Add labels
xlabel(' sample');
ylabel('dispatch');
title('PV and Pgrid and Pbess');

subplot(3,1,2);
plot(soc,'-*b');
xlabel(' sample');
ylabel('SOC');
title('SOC');

subplot(3,1,3);
plot(v,'-*b');
xlabel(' sample');
ylabel('Voltage');
title('Voltage');
```

Linking Algorithm Code :

Main function

```
% Combined Algorithm
% function for Rule based Dispatch rbd()
% function for Dynamic Optimization dyopt()
global count Math

count=1;
tot_loss=0;
E0=1000;
dt=30/60;
P_pv = xlsread('final_sc_data1.xlsx','J2:J193'); % reading pv data 99
197
```

```
Pload  = xlsread('final_sc_data1.xlsx','K2:K193');% reading load data
99 197
 net_load = xlsread('final_sc_data1.xlsx','I2:I193');% reading netload
data ie Pload-P_pv 99 197
del_load1=0;
del_load=0;

sw=0;
while(count<144)
   % E0;
    count
 [r1,r2,out_grid1]= %Call Rule based dispatch function
 V_rbd(count,1)=r1;
 P_rbd(count,1)=r2;
 P_out1(count,1)=out_grid1;
[r3,r4,out_grid,p2]= %Call Predictive Optimization function
V_dyopt(count,1)=r3;
P_dyopt(count,1)=r4;
P_out2(count,1)=out_grid;
p_plus(count,1)=p2;

% if rule based voltage deviation less than   predictive optimization
voltage deviation
 % Assign Battery dispatch based on voltage deviation

 Pg_demand(count,1) = net_load(count,1)+P_batt(count,1);%send(count,1)
%net_load(count,1)+P_batt(count,1);send(count,1);
 Math(count,1)= Pg_demand(count,1);
 result=runpf('case_dist_9bus_combined.m');
 loss(count,1)=(sum(get_losses(result)));
 tot_loss= loss(count,1)+ tot_loss;
 voltage(count,1)=(result.bus(10,8))
 E(count,1)= E0+(P_batt(count,1)*dt);
 soc(count,1)=E(count,1)/1000;
 E0=E(count,1);
 pgrid_new(count,1)= abs(Pg_demand(count,1));
p_rhc = abs(out_grid);

count=count+1;
end
 subplot(3,1,1);
plot(P_batt,'-b');
hold on;
plot(P_rbd,'--r');
hold on;
plot(P_dyopt,'-g');
% Add labels
xlabel(' sample');
ylabel('dispatch');
title('P_batt and P_rbd and P_dyopt');

subplot(3,1,2);
plot(V_rbd,'-b');
hold on;
plot(V_dyopt,'--r');
hold on;
```

```
plot(voltage,'--g');
% Add labels
xlabel(' sample');
ylabel('Voltages');
title('V_rbd and V_dyopt and Voltage');
% base=1;
% area(base,voltage)
subplot(3,1,3);
plot(soc,'-b');
xlabel(' sample');
ylabel('SOC');
title('Final-SOC');
```

Rule Based Dispatch Function:

```
function [v1,P_b,out_grid1] = % Rule based function definition
global inc pv_gen Pq_demand
%persistent Pload2 net_load P_pv
n=count;
%count=0;
inc=count; %initial sample starting from 2 because zero cannot be used
as index.
dispatch(1)=0;
batt_dis(1)=0;
%E=zeros(15,0);
Pgen=zeros(282,1);
% E(1)=75;    % initial soc value
% E_f(1)=75;
% Eref=75;    % max soc value
E(1)=1000;    % initial soc value
E_f(1)=1000;
soc(1)=1;
Eref=1000;    % max soc value

d=0;
dt=30/60;     % time step

P_pv2 = xlsread('final_sc_data1.xlsx','J2:J193') ;% reading pv data 99
197
 Pload2  = xlsread('final_sc_data1.xlsx','K2:K193');% reading load data
99 197
 net_load = xlsread('final_sc_data1.xlsx','I2:I193');% reading netload
data ie Pload-P_pv 99 197

%end
soc_ll=0.20;    %soc lower limit
soc_ul=1;       %soc upper limit
% while(n<49)       % reading values till less than sample size

%Pload2(inc)=Pload2(inc)+del_load1;
 %Pload2
m(inc,1)=Pload2(n);     % m stores the 30 min data
```

```
pv_gen(inc,1)=P_pv2(n); % this is the PV generation schedule in 30 min
interval


% Calculate the Pload – Ppv
%this becomes the load that must be satisfied by the battery taking
this sign
 % +ve means that value to be discharged from batt
 % -ve means that value to be charged from PV, as PV is excess.

  % if (m(inc)<0)     % charging value
  %if (net battery demand) <0)
  %Use the dispatch to charge the battery
  %else   % discharge value
  %Discharge the battery.
  % end

    % Pess needs to be between -25MW and +25 MW
    Pess(inc,1) =X_new(inc,1);
    %E(inc,1)=E(inc-1,1)+Pess(inc,1)*dt ;  % calculating the Echarge
value
    E(inc,1)=E0+Pess(inc,1)*dt
    soc(inc,1)= E(inc,1)/Eref ;            % SOC values to get into
condition checks.

    % 1st condition if the soc should be between upper and lower limits

    %2nd condition if the soc should be equal to lower limit or less
than the lower limits

      % 3rd condition if the soc should be equal to upper limit or
greater than the upper limit

    % dispatch variable is the net Pstorage or Pdischarge value,-ve or
+ve
    % +ve dispatch means charging value
    % -ve dispatch means discharging value
    switch d     % switching based on the conditions above based on d
variable
        case 1
            if (P_batt(inc,1)<0)
   % 1st condition satisfied then the available Pes is dispatched
            end
        case 2
            if (P_batt(inc,1)>0)
 % 2nd condition  % means it has hit the lower limit and cannot
discharge anymore, dispatch =0;

            else if (P_batt(inc,1)<0)
  % means it can charge since net Pess is negative so dispatch= Pcharge
                    batt_dis(inc,1)=0;
                end
            end
        case 3                                   % 3rd condition
            if (P_batt(inc,1)>0)
```

```
  % means battery has hit upper limit so it can only discharge, and no
charging
                dispatch(inc,1)=X_new(inc,1);        %since m(inc)> 0,
it can discharge
                batt_dis(inc,1)=X_new(inc,1);
            else if (P_batt(inc,1)<=0)
 % no charge possible since upper limit hit and since m(inc)<0 it is a
charge value
                    dispatch(inc,1)= 0;             % but since upper
limit no charge

  %Pgen(inc,1)=X_new(inc,1)    % if it hit limit then the available
value can be given to grid.
                end
            end
        case 4
           % if (m(inc)<0)
             dispatch(inc,1)=0;
            %end
        case 5
            %if ( m(inc)>0)
                dispatch(inc,1)=0;
            %end

    end
    E(inc,1)=E0+ dispatch(inc,1)*dt ;   %final Echarge values
calculated with the dispatch variable
    soc(inc,1)= E(inc,1)/Eref ;                  % final soc value
%     Pq_demand(inc,1)=m(inc,1)+dispatch(inc,1); % final net
Pg=dispatch+Pload-Pgen
      Pq_demand(inc,1)=net_load(inc,1)+ dispatch(inc,1); % final net
Pg=dispatch+Pload-Pgen

  result=runpf('case_dist_9bus.m');

  v(n,1)=(result.bus(10,8));
  v1=v(n,1);
%n=n+1;
%inc=inc+1;



% plot(v,'-*b');
 P_pv2 = pv_gen;         %xlsread('opt_file.xlsx','A2:A282') % reading
pv data
 pgrid =Pq_demand;      %xlsread('rule_dispatch.xlsx','E2:E282') %
reading load data
 P_b=dispatch(inc,1);
%xlsread('rule_dispatch.xlsx','C2:C282')
 soc =soc;                   %xlsread('rule_dispatch.xlsx','B2:B282')
out_grid1=Pq_demand(inc,1);
 end
```

Predictive Optimization Function:

```
function [v2,P_bb,out_grid,p2]
=dyopt_3_day_linking_new(count,E0,del_load)
persistent nn
nns=0;
P_pv1 =xlsread('final_sc_data1.xlsx','J2:J193'); % reading pv data 99
197
 Pload1  =xlsread('final_sc_data1.xlsx','K2:K193');% reading load data
99 197
 plot_pv= xlsread('final_sc_data1.xlsx','J2:J193');% 50
%end
%
 global final_grid i
 %Pload  =xlsread('opt_file.xlsx','D2:D290')
 %E0 = 1000;  % initial charge in battery
 i=count;
 %Emin = 20 MWhr
 %Emax = 75 MWhr
 %dt=0.0833; % timestep (5/60) 5 min interval for 24 hours
 dt=0.5;
a1=0.1;b1=12.6;c1=8;a2=0.0;b2=0.0;c2=0;% coefficients of cost function
% coefficients of cost fucntion
const=[a1 b1 c1 a2 b2 c2];                % defining the constants to
be used in code

%N=288 ;         % no of samples
N=48;
%N=96;
x0 = [zeros(size(1:N));-25*ones(size(N+1:N))]; % initial solution
assumption x0 but quadprog does notuse it.

% constraints formation
if (count==1)
% Define_constraints;
%bounds LB<=x<=UB

% ---- 2 bound constraints --- 1 equality constraint ----- 1 inequality
constraint
 % formation of matrix as shown in report
% ---equality constraint---
 % formation of matrix as shown in report
%---inequality constraint---
% formation of matrix as shown in report
% objective function
% Now since my solution consists of Pgrid and Pbatt for the full
horizon,
% it will have 2N variables
X = sym('x',[3*N,1]);  % reates an 2*N by 1 symbolic matrix filled with
automatically generated elements

E_init = sym('E_init');
c_o = sym ({'a1','b1','c1','a2','b2','c2'}.','r'); % creating symbolic
constants for the cost function, the constants are defined above
% the objective function is the sum of Pbatt and Pgrid cost function
```

```
C= (a1.*((X(1:N).^2))) + (b1.*(X(1:N)))+ c1  ; %(a2.*((X(N+1:2*N).^2)))
+ (b2.*(X(N+1:2*N)))+ c2
%C=((c_o(1)).*((X(1:N))))+c_o(2)
% Now forming the objective equation for the full horizon,so it will be
the
% above sum over the full horizon of solution variables
tot_obj = sum(C)

tot_obj = subs(tot_obj,[c_o],[const']); %substituting the constants
into the equations
% this function creates a function from the above formed equation.
matlabFunction(tot_obj,'vars',{X},'file','obj_test');
%F = double(gradient(tot_obj,X)); %the gradient of the function is
found
fsym = gradient(tot_obj,X); %the gradient of the function is found
f = double(subs(fsym,X,zeros(size(X)))); % now to get the constants,the
X()values are substituted zero.
H = 0.5.*double(hessian(tot_obj,X)); % the hessian of the objective
matrix is found
qpoptions = optimset('Algorithm','interior-point-
convex','Disp','iter');
%qpoptions = optimset('Algorithm','active-set','Disp','iter');
%options = optimset('MaxFunEvals',Inf,'MaxIter',5000,...
 %    'Algorithm','interior-point','Display','iter');
tic
[out,fval3,exitflag,output,lambda] =
quadprog(H,f,A,b,Aeq,beq,LB,UB,x0,qpoptions);
%[out,fval3,exitflag,output,lambda] = linprog(F,A,b,Aeq,beq,LB,UB,x0);
toc
out
fval3
% exitflag
final_out(i,1)=out(97);
p(i,1)=out(49);
%final_out(i,1)=out(193);
final_grid1(i,1)=out(1);
final_grid(i,1)=out(1);
%E(i,1)= E0+ (final_out(i,1)*dt);
%E0=E(i,1);
result=runpf('case_dist_9bus_dynopt.m');
  v(i,1)=(result.bus(10,8));
%i=i+1;

v2=v(i,1);
P_bb=final_out(i,1);
 out_grid= final_grid1(i,1);
p2=p(i,1)
else
    if isempty (nn)
        nn=1;
    else
        nn=nn+1;
    end
    nns=nn;
% output
% lambda
```

```
%plotResults( out, N);

 %while (i<49)
    % Pload1(i)= Pload1(i) + del_load ;
 N=48;
 Pload1
 x0 = [zeros(size(1:N));-25*ones(size(N+1:N))]; % initial solution
assumption x0 but quadprog does notuse it.
%
% % constraints formation
%
% % Define_constraints;
% %bounds LB<=x<=UB
%
% % ---- 2 bound constraints --- 1 equality constraint ----- 1
inequality constraint
%  % formation of matrix as shown in report
% % ---equality constraint---
% % formation of matrix as shown in report
% %---inequality constraint---
% % formation of matrix as shown in report
% % objective function
% % Now since my solution consists of Pgrid and Pbatt for the full
horizon,
% % it will have 2N variables
 X = sym('x',[3*N,1]);  % reates an 2*N by 1 symbolic matrix filled
with automatically generated elements
%
 E_init = sym('E_init');
 c_o = sym ({'a1','b1','c1','a2','b2','c2'}.','r'); % creating symbolic
constants for the cost function, the constants are defined above
% % the objective function is the sum of Pbatt and Pgrid cost function
 C= (a1.*((X(1:N).^2))) + (b1.*(X(1:N)))+ c1  ;
%(a2.*((X(N+1:2*N).^2))) + (b2.*(X(N+1:2*N)))+ c2
% %C=((c_o(1)).*((X(1:N))))+c_o(2)
% % Now forming the objective equation for the full horizon,so it will
be the
% % above sum over the full horizon of solution variables
 tot_obj = sum(C);
%
 tot_obj = subs(tot_obj,[c_o],[const']); %substituting the constants
into the equations
% % this function creates a function from the above formed equation.
 matlabFunction(tot_obj,'vars',{X},'file','obj_test');
%F = double(gradient(tot_obj,X)); %the gradient of the function is
found
 fsym = gradient(tot_obj,X); %the gradient of the function is found
 f = double(subs(fsym,X,zeros(size(X)))); % now to get the
constants,the X()values are substituted zero.
 H = 0.5.*double(hessian(tot_obj,X)); % the hessian of the objective
matrix is found
 qpoptions = optimset('Algorithm','interior-point-
convex','Disp','iter');
% %qpoptions = optimset('Algorithm','active-set','Disp','iter');
% %options = optimset('MaxFunEvals',Inf,'MaxIter',5000,...
%  %   'Algorithm','interior-point','Display','iter');
 tic
```

```
 [out,fval3,exitflag,output,lambda] =
quadprog(H,f,A,b,Aeq,beq,LB,UB,x0,qpoptions);
% %[out,fval3,exitflag,output,lambda] =
linprog(F,A,b,Aeq,beq,LB,UB,x0);
% %[out, fval3] = fmincon(@obj_test,x0,A,b,Aeq,beq,LB,UB,[],options);
 toc
% %out
% %fval3
% % exitflag
 final_out(i,1)=out(97);
 p(i,1)=out(49);
 final_grid1(i,1)=out(1);
 final_grid(i,1)=out(1);
 %E(i,1)= E0+ (final_out(i,1)*dt);
 %E0=E(i,1);
 result=runpf('case_dist_9bus_dynopt.m');
 v(i,1)=(result.bus(10,8))
 %i=i+1;

 pgrid=final_grid1;
 pbatt=final_out;
% soc=(1/1000).* E ;
% end
% %soc= ((E0)+(A_new1*out))/75 ;
%nn=nn+1
 v2= v(i,1);
 P_bb=final_out(i,1);
 out_grid= final_grid1(i,1);
 p2=p(i,1)
end
end
```

APPENDIX B: SYSTEM DATA

9 Bus Test System Data:

Line No.	**From Bus**	**To bus**	**R**	**X**	**P**	**Q**
1	**0**	**1**	0.1233	0.4127	1840	460
2	**1**	**2**	0.014	0.6057	980	340
3	**2**	**3**	0.7463	1.205	1790	446
4	**3**	**4**	0.6984	0.6084	1598	1840
5	**4**	**5**	1.9831	1.7276	1610	600
6	**5**	**6**	0.9053	0.7886	780	110
7	**6**	**7**	2.0552	1.164	1150	60
8	**7**	**8**	4.7953	2.716	980	130
9	**8**	**9**	5.3434	3.0264	0.500	200