DISTRIBUTED CONTROL OF A SMART GRID NETWORK USING IEC 61499

A Thesis in

Electrical Engineering

by

Srikrishnan Jagannathan

©2015 Srikrishnan Jagannathan

Submitted in Partial Fulfillment
of the Requirements
for the Degree of
Master of Science
August 2015
The thesis of Srikrishnan Jagannathan was reviewed and approved by the following:

Peter Idowu  
Professor of Electrical Engineering  
Thesis Advisor

Scott Van Tonningen  
Senior Lecturer in Electrical Engineering

Wolfram Bettermann  
Lecturer in Physics

Sedig Agili  
Professor of Electrical Engineering  
Graduate Program coordinator

*Signatures are on file in the Graduate School
ABSTRACT

Smart grid automation, in its preliminary and evolving state is achieved through centralized control, where a central control system monitors the entire grid automation and communications. These systems are more commonly known as SCADA systems which are essentially centralized in nature. SCADA systems are implemented using large software programs that are custom developed based on the application and are difficult to reuse for other applications. Modern control systems require flexibility for the purpose of advanced automation. To achieve this high level of flexibility in these systems a new software technology is required that is based on the interaction of distributed objects and that aims at decentralizing control.

The IEC 61499 is a novel method of software development that enables modeling control applications in a distributed manner. The standard presents guidelines for automation protocol in various applications ranging from industrial to smart grid. The IEC 61499 standard defines a distributed model for splitting different parts of an automation process and complex control into functional modules called function blocks (FB). These function blocks can be distributed and controlled across multiple controllers. The function block is a software unit that encapsulates some behavior and is hardware independent. The advantage of using function blocks is that it provides a method of graphical design of the system and also an easy way of distribution of functions in automation processes. Use of FBs makes the device control open and can be reconfigured more easily. End-users can modify the firmware of their devices based on changing technology and still able to use the same software to program the FBs. This enables the devices to adapt to changes in the grid.

The objective of this research is to highlight and demonstrate the benefits of microgrid generation through an innovative economic dispatch application implemented in IEC 61499. The contribution of this research is the use of function block concept to implement an economic
dispatch application considering Levelized Cost of Energy, the availability of Distributed Generators and the load forecast to balance loads. The Levelized Cost of Energy has been calculated based on studies conducted by the Energy Information Administration (EIA). Previous work in this field has been to perform load balancing using Function Blocks with a focus only on the availability, not the economic aspect. The economic modeling of renewable energy systems is implemented in the economic dispatch application in combination with a load forecasting model for each load. A saving in cost of energy production is expected to be achieved through the implementation of different test behaviors. Hence the effectiveness of each behavior is depicted by comparing the power supplied by the utility under the different applications and by comparing the cost of energy production of each of the generating sources for a single day. This cost value can help policy makers in deciding laws for interoperation of different utilities.
# TABLE OF CONTENTS

List of Figures .................................................................................................................. vii

List of Tables .................................................................................................................... ix

Acknowledgements ......................................................................................................... x

Chapter 1 Introduction ........................................................................................................ 1

1.1 Definition and Overview ......................................................................................... 1
1.2 Characteristics ......................................................................................................... 3
  1.2.1 Integrating Renewable Energy Resources ....................................................... 3
  1.2.2 Implementation of DERs - Concept of a Microgrid ....................................... 4
  1.2.3 Economic aspects of a smart grid ................................................................. 5
  1.2.4 Communication ............................................................................................. 5
  1.2.5 Control and Automation .............................................................................. 6
1.3 Research Motivation .............................................................................................. 10
1.4 Contribution ........................................................................................................... 11
1.4 Resources ............................................................................................................... 13

Chapter 2 Essential concepts .......................................................................................... 14

2.1 Theoretical background ......................................................................................... 14
2.2 IEC 61499- Function Block Concept ...................................................................... 15
2.3 General Characteristics ......................................................................................... 16
2.4 Function Block Types ............................................................................................ 19
  2.4.1 Basic function block ..................................................................................... 19
  2.4.2 Composite function block ........................................................................... 19
  2.4.3 Service interface function block types ......................................................... 20
2.5 Function Block Execution ...................................................................................... 20
2.6 Designing control Systems using IEC 61499 ......................................................... 23
  2.6.1 Application Model ....................................................................................... 24
  2.6.2 System Model ............................................................................................. 25
  2.6.3 Distribution Model ....................................................................................... 29
2.7 Exchange format for IEC 61499 entities ................................................................. 31
2.8 Why use Function Blocks ...................................................................................... 33
2.9 Economic Dispatch ............................................................................................... 34
2.10 Load Forecasting ................................................................................................. 35
2.11 Levelized Cost of Energy ...................................................................................... 35
  2.11.1 Factors affecting LCOE ............................................................................. 36
  2.11.2 LCOE calculations ..................................................................................... 39

Chapter 3 Software Tools, System Overview and Application Development .................. 41

3.1 Function Block Development Kit ............................................................................. 41
LIST OF FIGURES

Figure 1-1 Representation of a Smart Grid [2] ................................................................. 1
Figure 1-2 Concept of a Microgrid [8] ............................................................................ 4
Figure 1-3 IEC 61131 Function Block Diagram ............................................................ 7
Figure 1-4 FB diagram with feedback .......................................................................... 8
Figure 1-5 Ladder Diagram showing event based control [14] .................................. 9
Figure 2-1: IEC 61499 Function Block model ................................................................. 17
Figure 2-2 Execution Model [14] .................................................................................. 21
Figure 2-3 Timing of Execution [14] ............................................................................. 23
Figure 2-4 Application Model ....................................................................................... 24
Figure 2-5 System Model .............................................................................................. 26
Figure 2-6 Device Model ............................................................................................... 27
Figure 2-7 Resource Model ........................................................................................... 28
Figure 2-8 Distribution Model ....................................................................................... 29
Figure 2-9 Application Resource Management ............................................................ 30
Figure 3-1 Single Line Diagram ..................................................................................... 42
Figure 3-2 Simulink model of variable load profile ...................................................... 44
Figure 3-3 Circuit breaker dialogue box ....................................................................... 45
Figure 3-4: Stage 2 ......................................................................................................... 57
Figure 3-5 Flowchart for algorithm in Stage 2 .............................................................. 58
Figure 3-6 Reference Architecture for Stage 3 .............................................................. 61
Figure 3-7 Flowchart for Economic Dispatch ............................................................... 66
Figure 3-8 Command prompt ........................................................................................ 73
Figure 3-9 Composite FB receiving time values ........................................................... 75
Figure 3-10 Encapsulated basic FB .............................................................................. 75
LIST OF TABLES

Table 1-1 Load Balancing Schemes Comparison [23] [22]......................................................... 12
Table 2-1 Distribution Characteristics......................................................................................... 31
Table 3-1 Profile for Load 1 ......................................................................................................... 46
Table 3-2 Load profile for Load 2 ............................................................................................... 46
Table 3-3 Profile for Load 3 ......................................................................................................... 47
Table 3-4 Profile for Load 4 ......................................................................................................... 47
Table 3-5 Profile for Load 5 ......................................................................................................... 48
Table 3-6 Profile for Load 6 ......................................................................................................... 48
Table 3-7 LCOE equation parameters for DG1 ................................................................. 51
Table 3-8 Values used in LCOE equation for DG2 ................................................................. 52
Table 3-9 LCOE equation parameters for DG3 ................................................................. 54
Table 3-10 Function Blocks for Stage 2 ............................................................................... 59
Table 3-11 Function Blocks for Stage 3 ............................................................................... 64
Table 4-1 Cost of generation for 24 hours ................................................................................. 85
ACKNOWLEDGEMENTS

I extend sincere gratitude to my family for their support mentally and spiritually.

I would like to specially thank my committee - Dr. Peter Idowu, Dr. Scott Van Tonningen, and Dr. Wolfram Bettermann for giving me the opportunity to work in the exciting field of Smart Grids and for the guidance and mentoring they have given me during the course of my research. I also want to thank Dr. Peter Idowu for letting me work at the PPL Power Lab to improve on my practical knowledge during my research.

Thank you to Jim Christensen, Alois Zotl and other members of the IEC 61499 research group for their timely help at various stages of the project.

My sincere gratitude goes to faculty, staff and colleagues and friends for their moral support and otherwise.
Chapter 1
Introduction

1.1 Definition and Overview

A 'Smart Grid' can be viewed as an electricity grid with decision making abilities. It is an electricity network that can intelligently combine the power generation side and the consumer to ensure delivery of sustainable power in an efficient, economical and reliable manner. The core idea of a smart grid vision is to transform the electric grid from a centralized network controlled by the producer to one that is more decentralized and more consumer-interactive [1].

The Smart Grid technology can use sensors and meters to send and receive data about the grid. This can help in solving the problem of power disturbances in a grid and also increase the efficiency and security of the supply. State-of-the-art microcontrollers implemented with suitable software can enable distributed generation and help in avoiding grid congestions, thereby making progress on the implementation of renewable energy resources. A smart grid is developed with the intention of transforming the traditional grid into a modern system that is in pace with the developments taking place in the generation and distribution technologies.

Figure 1-1 Representation of a Smart Grid [2]
Real-time two-way communications are available in a Smart Grid that allow customers to communicate with the grid. Consumers are able to save energy and sell back energy to the grid. Communication is carried out through Advanced Metering technologies. Installing distributed generating units like residential solar panels and small wind turbines will improve the efficiency of the Smart Grid by generating power close to the loads, avoiding distribution losses, as well as avoiding peak shaving. Peak shaving is the process of reducing the amount of energy purchased from the utility company during peak hours when the rates are highest. It will allow small domestic customers and businesses to sell power to their neighbors or even back into the distribution grid. The same concept can be applied to larger commercial organizations that have renewable power systems that can give the excess power back into the grid during peak demand hours.

With the development of a Smart Grid, it is possible to improve the reliability of the power system and the power quality, and reduce carbon emissions by allowing the integration of renewable-energy sources and engage distributed storage in a cost-effective manner. With the provision of advanced infrastructure it is possible to develop new automated transmission and distribution systems that are capable of implementing various power system protection and control functionalities, such as managing reactive power in order to maintain system voltages, voltage regulation, enabling reduction in cost of operations based on marginal production costs ramping and load following, and reducing transmission loads when congestion costs are high. As a consequence of the development of a smart grid concept a power system network gets equipped with various features that are worth discussing. Section 1.2 presents the characteristics of a smart grid in detail.
1.2 Characteristics

The characteristics of Smart Grids include: (i) Integration of Renewable energy resources; (ii) Implementation of various Distributed Energy Resources (DERs) within a microgrid; (iii) Dispatch of Distributed Generation by considering cost factors such as Fuel cost, Operation and Maintenance cost; (iv) Use of diverse communication protocols for data exchange between different Intelligent Electronic Devices, digital relays, etc. from different vendors in a “plug and play” fashion; and (v) Distributed Control and Automation, for interaction with microgrids. These characteristics have helped transform the traditional power grid to match the ever increasing demand for power in a secure and reliable manner; at the same time making them proactive to emergencies. The characteristics are now explained in sub-sections 1.2.1-1.2.5

1.2.1 Integrating Renewable Energy Resources

A major aspect of implementing a smart grid is the integration of renewable resources and Distributed Generators (DGs) to the main grid. Renewable-energy resources, primarily including wind, hydro, biomass waste, solar and geothermal, can be implemented in either standalone or grid-connected mode for power generation. Greater penetration can be accommodated with increased use of smart grid technologies. Benefit of integrating renewable generation is that it enhances sustainability by reducing greenhouse gas (GHG) emissions, impacts on the environment, and the dependence on fossil fuels [3].
1.2.2 Implementation of DERs - Concept of a Microgrid

There is a sharp increase in the number of distribution systems with interconnected DERs [4], that are reliable and improve power quality [5] [6]. These distribution systems, along with the DERs and controllable loads, form a microgrid. A microgrid is located downstream at the load side of the distribution substation. It has the capability of being synchronized with the main grid [7]. This mode of operation is called as the grid connected mode. When in grid connected mode it usually connects to a grid through a substation transformer. A pictorial representation of a microgrid is shown in Figure 1-2.

![Figure 1-2 Concept of a Microgrid](image)

A microgrid is also capable of autonomous operation, wherein it is islanded from the rest of the grid, and is known as the islanded mode of operation [9].
1.2.3 Economic aspects of a smart grid

A smart grid has decision making capability, considering the monetary factors of operation at the generation side and the load side. Also, various demand response schemes are implemented for peak demand reduction, thus reducing the cost of generation and simultaneously offering customer benefits. One such economic aspect is the Levelized Cost of Energy (LCOE) which is implemented in this project. The LCOE is a conventional index that assesses the price of generated energy to recover the total cost of a plant over its stipulated physical lifetime. This index has been widely used for economical comparison of the types of power plants- both conventional and renewable [10]. Also various Demand Response schemes are implemented that help alleviate the need to operate high cost generating units and also help reduce peak demand [11] [12].

1.2.4 Communication

Digital communication networks form the framework for data exchange in a smart grid. Communication is carried out through various industrial protocols such as Modbus, IEC 61850, DNP3 and Profinet. These protocols set a standard for the design of communication links in substation automation systems. They set a standard for the way the digital data is exchanged between various devices and define a consistent method for connecting IEDs in substation applications. The main advantages of using the communication standards is interoperability, along with reduced installation costs through the usage of Ethernet cables and other communication infrastructure that have largely reduced the wiring. For example, in the IEC 61850 standard the use of the object oriented concept in design along with the use of a standardized Substation Configuration description Language (SCL) makes the task of substation
design very simple. Usage of GOOSE messages make the monitoring and protection fast and very reliable. GOOSE (Generic Object Oriented Substation Event) messages are implemented to carry peer-to-peer communications between different Intelligent Electronic Devices (IEDs).

1.2.5 Control and Automation

Control and Automation systems in smart grid has been classified as Programmable Logic Controllers (PLC) or Distributed Control Systems (DCS). Current DCSs comprise a few large central processors, that provide supervisory control and data acquisition. These processors communicate via local networks with numerous instruments, controllers, sensors and actuators located out in the plant. IEDs positioned at remote locations in the plant typically provide local closed loop control. PLC systems, in contrast, have a Human Machine Interface (HMI) along with the control processors that are provided with a different variety of switches, panels and lights. A large PLC system has multiple PLCs communicating via one or more proprietary high-speed networks. PLCs can handle several input and output (I/O) signals, both analog and digital. Both design approaches have systems with large monolithic software packages, that are difficult to reuse in new applications and are difficult to integrate with each other. Data and functionality of one PLC are application specific and cannot be available for other applications, even though the applications are run on the same machine and written in the same programming language. Significant portion of system development time goes into providing drivers to allow different types of instruments and controllers to communicate and mapping signals between devices. Both kinds of systems are difficult to extend and modify and do not provide the high degree of flexibility that is expected in systems for advanced automation systems.

However users are now requesting a more open approach to software. As a consequence a new generation of controllers is emerging that can be programmed in a wide range of
programming languages. These soft-controllers are being referred to as Programmable Automation Controllers (PACs). They offer high levels of integration and more flexibility in designing the systems.

Control systems of the future smart grid offer various computer-based control and automation solutions that offer economic, environmental and technical advantages by lowering operational costs, and reducing greenhouse gas emissions, improving reliability respectively [13]. The IEC 61499 standard has been established as reference system architecture for implementation of such distributed automation systems. The standard is built on function block concepts described in the PLC language standard IEC 61131-3. The FB of the IEC 61499 standard makes up for the limitations in a FB of the IEC 61131-3 standard. This can be explained through Figure 1-3.

![Figure 1-3 IEC 61131 Function Block Diagram](image)

The network in the Figure 1-3 is developed using a FB concept defined as a part of the IEC 61131 standard. Each function block consists of a single internal algorithm that is executed when the function block is invoked. Regular execution order is determined by the dependency of one function block on the other function blocks. The order is normally running from left to right.
since function blocks to the right have a dependency on the output values from the function blocks on the left. However with the introduction of a feedback path, as depicted in Figure 1-4, the determination of execution order cannot be carried out from the diagram.

![FB diagram with feedback](image)

**Figure 1-4 FB diagram with feedback**

This is because the execution of both function blocks depends on an output value from the other function block. As the network becomes more complex, it becomes very difficult for a programmer to determine a valid order of execution. This problem can be overcome, by providing additional mechanisms that define the order of execution of function blocks such as manually assigning an execution order for a list of function blocks. However such methods are not portable or consistent across different control systems.

IEC 61131-3 has a crude mechanism for passing execution flow through a chain of function blocks through the use of the EN input and ENO output signals as shown in Fig 1-5. The EN and ENO signals, when used in rungs of a Ladder Diagram, were intended for passing of ‘power flow’ in function blocks. However, it has been identified that the EN and ENO signals do not provide the degree of flexibility required to build complex function block networks. The effect of EN and ENO signals can be considered as a means of passing of events between
function blocks. EN indicates that the function block can be invoked since the data at its input is ready. The objective of the ENO signal is to indicate that the function block has executed and the output data is available to be sent to the function block that is next in the execution order. It is the idea of event passing, included in IEC 61499, that helps in overcoming the limitations of IEC 61131 function block concept.

![Ladder Diagram showing event based control](image)

The FB of an IEC 61499 standard is hence an event-driven module that encapsulates one or several tasks. The objective of implementing this standard is to develop flexible and reconfigurable automation systems [15]. It solves the problem of interoperability. Interconnection of software components in the form of function block instances will be of tremendous advantage to users. The advantages include improved software productivity, reusability of standard solutions and design flexibility by being able to run software and devices from different developers. This approach helps in distributing the control functions and making control decentralized in nature, which helps reduce response time of the control system. Various process control applications have been achieved using a combination of IEC 61499 and IEC 61850. Smart grid is seen as one of the potentially large areas of its application [15].
1.3 Research Motivation

With appropriate strategies to implement the characteristics of a smart grid at a practical level, a traditional power system can be transformed into an intelligent grid that makes it efficient and economically beneficial to operate; it is a win-win situation for the power producers and consumers. The motivation of this research is to implement intelligent smart grid control at the application level in a smart grid, bringing out the characteristics and features discussed in section 1.2. As mentioned in section 1.2.5 traditional control systems such as the PLCs and DCS, lack the flexibility and design and fail to offer the desired levels of integration. The IEC 61499 automation standard is being considered as a novel automation solution to overcome these limitations and is gradually replacing centralized control [16]. The advantage of this standard is that it provides a solution that is flexible and can be used to model control systems by breaking down the control process into basic Function Blocks (FB) instances. It is currently being implemented in a variety of industrial applications [17] [18] and its range of applications is now extending to include smart grids. The functionalities implemented using this standard have been in the fields of protection coordination and fault isolation and restoration [19] [20] [21][19]. The objective of this research is to show that the range of applications can be extended beyond emergency control systems such as fault management systems into applications requiring coordination even under steady state situations. This has been achieved by proposing a novel economic dispatch application using IEC 61499 in this thesis.
1.4 Contribution

The research described in this work proposes an innovative economic dispatch application that is implemented through the Function Block concept of IEC 61499. Previous work in this field has been to perform load balancing using Function Blocks with a focus only on the availability of power generation, without including the economic aspect. The previous work also does not consider the aspect of loads being able to predict their need or, alternatively, the feature of load forecasting [22] [23]. The work carried out in thesis implements an economic dispatch application with factors that have not been included in the previous work. The economic modeling includes studies conducted by the Energy Information Administration (EIA) and National Renewable Energy Laboratory (NREL) to estimate the parameters of the cost function. The inputs to the control system performing the economic dispatch application will include the cost of generation for renewable and conventional energy systems, based on real world economic constraints such as capital cost, fuel rate, heat rate along with the availability of generation and the forecasted load profile. The final goal is to achieve a saving in cost of energy production through the implementation of each behavior. Hence the effectiveness of each behavior is depicted by comparing the power supplied by the utility under the different applications and by comparing the cost of energy production of the various Distributed Generators for a single day.

A comparison of the load balancing models that have been implemented is done in Table 1-1. The application developed in this research is titled as Penn State Harrisburg (PSH) model and is compared with the following load management applications: (i) Future Renewable Electric Energy Development and Management (FREEDM) model, [22]; and (ii) Centralized, Distributed and Hybrid Demand response model [24]. The features of (i), (ii) and the PSH model are depicted in Column 3, 4 and 5 of Table 1-1, respectively.
<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Features</th>
<th>FREEDM Model</th>
<th>Central, Distributed and Hybrid Model</th>
<th>PSH model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load Forecasting</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Microgrids implementation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Function Block Implementation</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Sub-applications on multiple devices</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Load queuing</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>6</td>
<td>Hardware in the loop</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>7</td>
<td>Economics of Generation</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Economic Dispatch</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

On comparison with the other load balancing applications it can be observed that the PSH model includes the features of economics of generation technologies and implementing variable loads that are able to predict their duration of operation in the load, transforming a simple load balancing application into an economic dispatch application. The economics of generation is implemented using the Levelized Cost of Energy. The values considered in this work have been supplied by US DOE [25]. As shown in Table 1-1 the model developed by FREEDM for load balancing using IEC 61499 shows an algorithm for load balancing considering only availability of generation to balance the load [26, 22]. The Central, Distributed and Hybrid model was developed as a simulation model, implemented on OMNet++ network simulator, and focused on demand side management instead of supply management. The contribution of this thesis is to use the function block concept to run and test different behaviors of economic dispatch applications considering Levelized Cost of Energy and availability of Distributed Generators to balance loads.
This also implies that the focus of the application to control the generation rather than the load. The feature of load queuing being associated with demand control, can be implemented in the future when the application is focused on demand side management along with the hardware implementation of the economic dispatch application.

The consideration of a cost model and variable load profile transforms a basic load balancing application into an economic dispatch application. By considering these factors a saving in cost of energy production is expected to be achieved through the implementation of each behavior. Hence the effectiveness of each behavior is depicted by comparing the power supplied by the utility under the different applications and by comparing the cost of energy production of the various generating sources for a single day. This cost value can help policy makers in deciding laws for interoperation of different utilities.

1.4 Resources

Simulink-MATLAB is used for implementing the power system model due to its ease of implementation. It can easily be co-simulated with the software used for FB application development. The software tools or platforms available for implementing Function Block applications are: (i)ISAGraf; (ii)Function Block Development Kit (FBDK); (iii) NxtStudio; and (iv)Framework for Distributed Industrial Automation and Control (4DIAC). 4DIAC and FBDK are open source softwares. FBDK is chosen as the platform for simulation in this project because of the ease of its implementation. NxtStudio and ISAGraf are industrial grade licensed products. NxtStudio, 4DIAC and FBDK have a feature of co-simulation with MATLAB [27] [28]. Previous work has shown that MATLAB-Simulink also can be used to model the IEC 61499 FB models [29]. A co-simulation, or alternatively, a time synchronized simulation between Simulink and FBDK is used to achieve the objective of this research.
Chapter 2

Essential concepts

This chapter describes the concepts implemented in this thesis. It starts with the IEC 61499 standard as a whole- from the basic component to the development of an application using the standard. It describes the architecture and concepts that make up the Function Block Standard along with the purpose of using Function Blocks. Subsequently the concepts of economic dispatch, load forecasting, and the Levelized Cost of Energy are described. The economics consider how conventional and non conventional sources of generation are evaluated using an economic model. This model considers factors such as capital investment, annual operation and maintenance costs, cost of fuel and the capacity factors.

2.1 Theoretical background

The International Electrotechnical Commission (IEC) has developed a specific standard IEC 61499 that defines how function blocks can be used in industrial systems for distributed processes, measurement and control systems. Function blocks are established as a concept for defining reusable software components that are robust, in industrial systems. A function block can provide a software solution to a small issue, such as the opening and closing of a valve, or control of a major unit of a plant, such as a complete line of production. Function blocks can encapsulate industrial algorithms in a manner such that they can be easily understood and applied by people who are not required to have a software background. Each function block has a defined set of input data, which are read by the internal algorithm during its execution. The results of the algorithm are then written to the function block’s outputs. Using this concept, entire applications
can be built from a network of function blocks through the interconnection of function block inputs and outputs.

IEC 61499 standard defines a general model and methodology for describing function blocks in a manner such that it is independent of implementation. System designers can use this methodology to develop distributed control systems. It allows the defining of a system in terms of logically or sequentially connected function blocks that are capable of running on different processing resources.

The primary objective of IEC 61499 is to serve as an architecture and model for distributed systems rather than a programming methodology. IEC 61499 provides a set of models that describe the structure and behavior of distributed industrial process measurement and control systems using the function block concept. It provides terminology and concepts that describe the implementation of a function block oriented distributed control system in a formal and unambiguous manner. This allows systems to be validated, compared and understood, which is the first step towards standard programming methodologies for distributed systems.

2.2 IEC 61499- Function Block Concept

IEC 61499 is built on the concept of a Function Block, consisting of event and data inputs and outputs, encapsulated algorithms, connected together with a data or event interface. Execution of a FB is based on the Execution Control Chart (ECC). The ECC is essentially a state machine processing algorithms and scheduling events. Event inputs and outputs are used to synchronize function blocks and are used to form the schedule in the Execution Control Chart. They are used to schedule the execution of algorithms in an application. A FB remains inactive until it is triggered by an event input. This trigger in the input could be an event output from another FB. Data inputs and outputs act as an interface to other FBs. When function blocks
algorithms and their execution control are expressed in terms of interconnected FBs these are called composite function blocks. An application is defined by a network of function blocks and the execution schedule of their algorithms. [30]

2.3 General Characteristics

Each function block has an instance name and a type name. These names should always be depicted when the function block is shown graphically. Function blocks can have event inputs that receive events from other function blocks via event connections. They have event outputs, which can be used to send out events to other function blocks. A function block can have data inputs that can either be entered by the programmer or be passed in from other function blocks. It can have data outputs to pass the data values, that are produced within the function block, out to other function blocks.

Using the WITH qualifier, events can be associated to data. Represented graphically, this is shown using a small square connector that links the event with the required data. On the occurrence of an input event the input variables associated with it are updated or made available to the FB while other input variables retain their values or are unaffected. Similarly when the output events are triggered their associated data outputs will be made available while the other outputs stay unchanged. The data types supported by IEC 61499 are the same as the data types supported by IEC 61131-3. IEC 61499 does not support global variables. Therefore the encapsulated functionality has access to function block input, output and internal variables only. Function block data is retained between function block invocations.

Figure 2-1 shows the main characteristics of an IEC 61499 function block. The top part of the function block is called the ‘Execution Control’ portion. It contains a definition to map
events on to an encapsulated algorithm or functionality. In other words it defines which functionality defined in the lower part of the FB is to be triggered on the arrival of input events at the ‘Execution Control’ and which of the output events are triggered. The standard defines a way to map the relationship between events arriving at the event inputs, the execution of encapsulated algorithm and the triggering of the associated output events. The lower portion of the function block contains the encapsulated functionality along with internal data variables. Both, the algorithm and the internal data are hidden within the function block. A function block is such a type of software component that, if designed well, there should be no requirement for a user to have a detailed understanding of its internal design. A function block relies on the support of the resource in which it is contained to provide means to map requests to communications and process interfaces and schedule encapsulated functionality.

Figure 2-1: IEC 61499 Function Block model
The concept of FB can be extended to object orientation in other software languages like C++. For example the algorithms written inside a FB can be compared to the methods in an object in an Object Oriented Programming language. However as mentioned before the algorithms are hidden inside a FB. Also the data inside a FB is local to that FB. Alternatively there is no global data present. Thus the FB algorithms do not depend on other FBs. This aspect greatly aids the feature of reusability of a FB for other applications as well.

The function block is the functional unit of IEC 61499 programming. IEC 61499 defines a distributed control system through various reference models such as the application, device, and resource models. An application is a set of related function blocks that must talk to each other in order to fulfill a control task, a device is a control unit having one or more processors and a resource is a processor on which part of a distributed application will run.

The program on the processor is the application. Resources on a device will have interfaces to the communication systems and to the device specific processes. The communication systems will allow data exchange between FBs in remote resources and devices and interface to read and write local device I/Os. ‘Service Interface’ FB is a special kind of FB that provides a link between FB and different interfaces of a resource. For example, a communications SI block can be used to send or read data to FBs in other resources.

The management of a distributed application requires proper coordinated functioning of multiple resources. FB events are used to schedule an algorithm within the block at the correct priority and time. While a system vendor usually supplies a type definition of FB, the programmer can then create multiple instances of the FB block with the capability of each instance running independently.
2.4 Function Block Types

IEC 61499 has the ability to define a function block type, which basically defines the interfaces and behavior of function block instances that can be created from the type. This is analogous to object-oriented software, in which the behavior of object instances is defined by the definition of the associated class of the object. A function block type has a type name, formal definitions of the input and output events of the function block, and definitions of the input and output variables. The type definition additionally includes the internal behaviour of the block. This, however, is defined in various ways for the various types of function blocks.

2.4.1 Basic function block

A basic function block is defined in terms of the algorithms that are called as a response to input events. After the algorithms are executed, the FBs trigger output events to signal the occurrence of certain state changes within the function block. Events are mapped on to algorithms by using a special state transition notation called an Execution Control Chart (ECC).

2.4.2 Composite function block

The composite function blocks internal behaviour is defined through a network of interconnected function block instances. A composite FB therefore includes data and event connections that must be present between the internal function block instances.
2.4.3 Service interface function block types

Service interface function blocks (SIFB) are an interface between the function block domain in the software and external services. An example of a SIFB instance is a FB to communicate with remote device or to read the value of a hardware real-time clock. Service interface function blocks are primarily concerned with data transactions and are hence defined using service sequence diagrams.

2.5 Function Block Execution

This section focuses on the ‘Execution Control’ portion of the Function Block, mentioned in Section 2.3. Figure 2-2 shows the execution model of a FB. Execution Control involves event inputs and outputs that are used to synchronize the function blocks thereby forming the schedule in the Execution Control Chart. They are used to schedule the execution of algorithms in an application. A FB remains inactive until it is triggered by an event input. This trigger in the input could be an event output from another FB. Data inputs and outputs act as an interface to other FBs. The ECC is essentially a state machine processing algorithms and scheduling events. When basic function blocks, algorithms and their execution control are expressed in terms of interconnected FBs these are called composite function blocks. An application is defined by a network of function blocks and the execution schedule of their algorithms. [17]

The event stream initiates the execution of the function block code. The arrival of an input event triggers the execution of one or more algorithms. After the execution of algorithms the function block will enable an output event. Input events come from other function blocks while output events are sent to other FBs. Data inputs are provided from physical devices or from other FBs. When an input event occurs the data values are read and the algorithms are executed. After the output event is sent the output data is made available to the next FB. The execution of
the control algorithm for a particular FB is described by an Execution Control Chart. Each FB has a type name and an instance name that are shown. Each block will have a set of internal variables that are used to hold values retained between algorithm invocations. A resource may provide features to allow the internals of a FB to be accessed. A function block type is defined by a type name, formal definitions for the blocks inputs and output events, and definitions for input and output variables. The type definition also includes the internal behavior of the block and is defined in different ways for different forms of block.

The different parts of a FB are handled by an underlying scheduling function. The scheduling function is provided by the resource. It ensures that each phase of function block execution occurs in the correct order and at the correct priority. There are a number of discrete timing phases, which are required by the FB to execute. Each of these phases may take some time to elapse. Each phase depends on the interaction between FB and underlying scheduling function. There are eight timing points that must occur sequentially for a FB to operate. Termination of each phase is defined by a timing point. [29]

**Figure 2-2 Execution Model [14]**

**Time 1:** Values from external or other FBs are made available at current FB inputs.
**Time 2**: Event associated with the corresponding input values arrives at the event input.

**Time 3**: The FB execution control signals the scheduled function that the data inputs are available and the algorithm is ready to execute.

**Time 4**: After a certain amount of time which depends on the loading and performance characteristics of the resource the scheduled function starts to execute the FBs algorithm.

**Time 5**: The enclosed algorithm is executed using the internally stored values and inputs and subsequently generates new output values which are mapped onto the FBs outputs.

**Time 6**: After the algorithm is executed the algorithm, the scheduled function receives a signal to indicate that the outputs are available.

**Time 7**: The FB’s execution control receives a signal to the scheduling function to generate an output event.

**Time 8**: An output event at the FBs output event interface is then created.

\[
T_{\text{setup}} = T_2 - T_1 = \text{Time difference between the time it takes to make the input values available (i.e. updated by preceding function blocks) and the arrival of an input event.}
\]

\[
T_{\text{start}} = T_4 - T_2 = \text{Time between the receiving of an event input and execution of the function block’s encapsulated algorithm. This duration may depend on the resource loading characteristics } g, \text{ i.e. how many other function blocks are in queue to be executed by the scheduling function.}
\]

\[
T_{\text{algorithm}} = T_6 - T_4 = \text{Time taken for the execution of the function block’s enclosed algorithm.}
\]

\[
T_{\text{finish}} = T_8 - T_6 = \text{Time from post execution of the enclosed algorithm and triggering of the event output.}
\]
2.6 Designing control Systems using IEC 61499

Using the general characteristics of a Function Block along with the various types of Function Blocks described in Sections 2.3-2.5 control systems can be designed through a platform independent approach. The whole design process is basically divided into two steps.

Step 1) Developing an Application Model: This step involves specifying the functionality and composition of each block described in section 2.5. This model does not involve specifics about hardware and communication protocols.

Step 2) Developing a System Model: Here the details of the hardware configuration (device model, resource model) and communication protocols are mentioned. The functions are mapped to the system through this model.
2.6.1 Application Model

An IEC 61499 application is essentially a network of interconnected function blocks, linked by events and data. An application therefore comprises instances of function block and the definitions of their interconnections. Applications can include multiple instances of function blocks of a particular type of function block. As a principle of IEC 61499, all control behavior is defined in terms of function blocks. Consequently global or local variables cannot exist outside of function blocks in an application. This is a very important difference between an application developed for a PLC based using IEC 61131-3 and an application based on IEC 61499. The IEC 61499 application model is shown in Figure 2-4.

Event connections are directed from an output event to an input event. Data connections can be such that one data output can be connected to several data inputs of other function blocks. The reverse however, that is several data outputs connecting to a single data input, is not allowed. This is because function blocks are completely decoupled in IEC 61499 and have no information about the source of input connections. Hence if multiple data inputs were allowed, a function block would not be able to determine the connection from which to take the data. Also incoming data and event connections do not have necessarily originate from the same function block. This
can be considered as a helpful feature in the synchronizing of sampled and processed data that originates from several function blocks with the occurrence of one event.

An application is the entire set of function blocks and their interconnections that is used to solve a particular control and automation problem. A good example can be the application developed with a set of function blocks for the control a production line. The application model focuses only on the overall control functionality and not on any particular hardware. This is in contrast to IEC 61131-3 application development which is more resource focused [29]. IEC 61499 application development can therefore be considered more application oriented since the application is the foundation element in system development. This gives the control engineer a great advantage since he can focus on the functionality. Changes in hardware do not impact the application model much and the modified hardware can continue to support the feature of reusability of general control functions. The application model facilitates the testing and simulation of applications independent of the specific control hardware, thereby enabling control engineers to evaluate only the functionality which is crucial at early stages in the development process.

2.6.2 System Model

The system model defines the behavior at the physical level. At the hardware level a distributed system comprises a set of devices inter-connected through different networks to support a group of simultaneously operating applications. An application requires the running of the control software in a number of devices. Previously, only few distributed applications required a small percentage of software running in remote devices for example, loop and temperature controllers. In such applications the main intelligence was in a central device such as a PLC[29]. With devices such as actuators and smart sensors beginning to provide more
processing capability, software functionality has the capability to become distributed across many more devices, to an extent where there is not one single main controlling device.

*Overall system structure:* This model defines the available control devices and the communication links between those devices. This forms a network of inter-communicating devices. Communication links may be of different types and different devices can be connected to multiple or different communication segments. This enables the formation of communication links as shown in Figure 2-5.

![Figure 2-5 System Model](image)

*Device model:* This model, shown in Figure 2-6, describes the entire structure of the physical control device that supports the execution of an IEC 61499 application of function blocks. The device provides an infrastructure to support multiple resources. An IEC 61499 resource is similar in concept to the resource of a PLC in the sense that it provides a processing and execution environment for the network of FBs. However, IEC 61499 resources do not have to necessarily bound to an execution unit such as a CPU. An IEC 61499 resource is a logical separation within a device. It provides independent control and execution of a networks of FBs. A device has a process interface that enable resources to read and write data to the Input and Output
(I/O) points on the physical device. Also present is a communications interface that allows resources to exchange data with resources in the same or remote devices. When a device may be visible on more than one network the communications interface can handle the access to those devices as well.

<table>
<thead>
<tr>
<th>Communications Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource 1</td>
</tr>
</tbody>
</table>

**Figure 2-6 Device Model**

*Resource model:* A resource is a part of a device and essentially provides services and facilities that support the execution of applications of function blocks that are allocated to it. In order for that to happen, the resource must ensure that events should schedule the appropriate functionality encapsulated within the function blocks at the appropriate priority and time. The resource has the responsibility of retaining the values of variables inside of function blocks in between calls. The resource propagates events and transfers data between function blocks that are present either on the same resource or remote resources that are linked through communication services. Each resource has an interface to the communications system that enables function blocks to exchange data with function blocks in remote resources along with an interface to read and write to local device inputs and outputs. The resource is mainly responsible for mapping of
event flows and data from the local resource to remote resource function blocks through the device communications interfaces.

Figure 2-7 Resource Model

Figure 2-7 depicts the IEC 61499 resource model. Shown within the resource, is an application using function blocks linked via event and data flows. A resource provides a scheduling function that ensures an orderly execution of the functionality encapsulated within function blocks, based on the requirement of the arrival of events at each function block. The FB that provide a link between function blocks and the interfaces of the resource are called Service interface function blocks. A resource can support independent operation. A resource can be started, configured, loaded, and stopped without affecting other resources in the network or in the same device. However, one important point to note is that while managing a distributed application the co-ordination of multiple resources, in which function block network fragments are loaded, is required.
2.6.3 Distribution Model

This model links the application model to the system model. It is concerned with associating parts of an application to devices and their resources and to execute configurations that are device specific, to the assigned application parts.

Mapping of applications: Applications are distributed such that they can run on several resources. A distributed application comprises interconnected function blocks with segments of function blocks running on designated resources. For IEC 61499 the distribution is considered to be the assignment of FBs in an application onto different resources of the devices defined for the system model. This assignment is done through the mapping statement. Figure 2-8 depicts a mapping of a sample application on to devices of a system model.

![Distribution Model Diagram](image)

Figure 2-8 Distribution Model
One of the specific features of IEC 61499 is that the distribution model allows devices to execute more than one application. The device model is defined such that it is possible to load and unload distributed applications from a device without affecting the already running applications on the devices. Since a device can support multiple resources, different applications can be separated within a device by mapping them to different resources (Fig 2-9). There is also a possibility of mapping an application only to a single device, having one or multiple resources within the same device (such as the Application 2 running on Resource 2 in Figure 2-9).

**Figure 2-9 Application Resource Management**

The timing and performance characteristics of applications therefore depend on the resources over which they are distributed and the communications networks through which they are connected. For example, consider two identical applications, running on two different communication networks of controllers. One network has data links running at 1 MBs and the other runs at 1 GBs. Since the communications data rates of the two networks are different, the two applications will clearly have different response and performance times due to network
latencies. However their internal software algorithms will be the same. Function blocks on the other hand only run in a single resource. In this respect, function block performance is unaffected by the characteristics of communications networks. The performance though may to a lesser extent be affected by the characteristics and behaviour of the resource or alternatively the computational power of the device in which the function block has been initiated. Table 2-1 gives a comparative overview of the performance characteristics for a Function Block and an application.

<table>
<thead>
<tr>
<th></th>
<th>Distributable</th>
<th>Timing</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Block</td>
<td>No</td>
<td>Depends on device</td>
<td>Depends on device</td>
</tr>
<tr>
<td>Application</td>
<td>Yes</td>
<td>Depends on device and Communications</td>
<td>Depends on device and Communications</td>
</tr>
</tbody>
</table>

Table 2-1 Distribution Characteristics

2.7 Exchange format for IEC 61499 entities

IEC 61499 has the ability to exchange all models between various platforms and software tools. The ability to exchange function blocks between different systems and products brings significant end-user benefits. This includes major cost savings by being able to reuse function block designs in different hardware and system configurations. IEC 61499 supports this by providing two options: (i) utilising the web-technology Extensible Markup Language (XML) and (ii) a formal textual syntax

XML-exchange format: The ability to exchange software designs between products of different vendors has been a serious drawback of the IEC 61131-3 PLC software standard. This is
because IEC 61131-3 did not include a full definition of a file exchange format[29]. Hence it is not possible to exchange graphical PLC software designs through the usage of IEC 61131-3 languages between different PLC software tools. To overcome this limitation, an XML-based file exchange format was developed by the PLC open organization. The XML format provides a method for storing the structural features of IEC 61499 models such as the function block instances and connections of an application along with the graphical layout information such as the position of function blocks and their bending points of connections, in a vendor independent manner. This exchange format enhances feature of portability in IEC 61499 models since data can be exchanged between tools from different vendors.

_textual syntax for IEC 61499 entities:_ This format allows the IEC 61499 standard to model the entities such as function blocks, system configurations and applications to be described in a text language. These text definitions can be used to define models in an unambiguous manner so that their graphical representations can be created consistently and automatically. Consider an application that is constructed from a set of interconnected function blocks with their appropriate data and event connections as well as parameters. All these aspects of designing the application can be represented using the textual format of IEC 61499. It has been envisioned that there will be a possibility to compile textual definitions and use them as a way to check the validity of a particular model. Textual Syntax defines the semantic aspects of a model. It does not fully describe the graphical details. Consider, a particular model that is developed showing function blocks located at certain positions in a network diagram. The actual positions of these function blocks and the finer graphical details such as font and colour are not described using this syntax. Only the logical connections between function blocks are defined precisely. The textual form is a very useful form of representation that can be used for porting model designs from one platform to another.
2.8 Why use Function Blocks

Now that we have described the Function Block concept and characteristics, this section describes the benefits of using FBs. A function block, conceptually, can be represented as a software unit that is used to represent software as blocks of hardware. In essence a function block is a model of software that regards the enclosed behavior in a way similar to an electronic circuit. The difference is that when improving the software development process, the most important aspect is to add more abstractions and higher level software modeling aspects in order to keep up with the increase in complexity of industrial systems and smart grid.

Object Oriented programming, focused on the development of Objects, has been considered as a solution to devise complex control systems. Function blocks are in many ways similar to objects. They are becoming successful in the industry because they can be used to model the concepts and behavior of the real world. Implementing the IEC 61499 reference architecture for distributed control gives the following benefits:

*Modeling the real world:* When designing an application, it is beneficial to break down complex real-world problems into basic software components. This reduces the complexity of the problem and simplifies the process of application development.

*Reusability:* Generally, objects are elements that remain the same. In many situations, developers are able to re-use the same object classes in a variety of applications. The same can be applied to function blocks. For example, when an object is created to represent the behavior and characteristics of an ‘employee’, the same object class can be used in different organizations with many employees. An object of ‘employee’ would have details such as name, age, address, job description and salary etc., and the respective functions to input this information. Once a FB is developed and tested it can become a part of a database or library. These FBs can be used either locally or globally.
Reduce complexity: A FB can be used without understanding the internal behavior as long as its function is known. An application can be built by creating and linking FBs and there is no need to understand the internal mechanism of the object. It eases the development and maintenance of various control schemes. The amount of time required to build an application is greatly reduced. There is modularity of code and easy reconfiguration of the system.

Easy reconfiguration: IEC 61499 provides a method of graphical design of the system and also provides an easy way of distribution of functions in an automation process. Use of FBs makes the device control open and can be reconfigured more easily. End users can modify the firmware of their devices based on changing technology and still be able to use the same software to program FBs. This makes the devices capable to adapt to changes in the grid.

2.9 Economic Dispatch

Economic dispatch is defined as the reliable operation of generating units to produce energy at the lowest cost to consumers, while considering the constraints that exist on the system [31]. The classical economic dispatch approach involves allocating generation to loads entirely and in a economical manner. The four major focus areas in the field of economic dispatch are [32]:

(a) Optimal power flow- focuses on an minimizing costs of generation in a system by considering P,Q and voltage constraints in a system

(b) Economic dispatch in relation to Automatic Generation Control(AGC)- Provides Optimum dispatch with Load frequency constraints.

(c) Dynamic dispatch: Considers minimizing operating costs and providing generation based on forecasts of load.
(d) Economic dispatch with non-conventional generation sources: Economic dispatch solution with financial constraints on cost of modeling facilities with renewable energy generation.

At the core, economic dispatch aims at optimizing an objective function that is built considering the different constraints for a system. These constraints can be the voltage and frequency set points, cost of generation, or a load forecast parameter. Each of these constraints in turn can be affected by various parameters. For example, the load forecast variable could be modeled as a stochastic process, or the cost of generation can be affected by parameters such as location, fuel cost and capacity.

2.10 Load Forecasting

Load forecasting is an important aspect in power system planning and operation. Forecasting implies predicting the load ahead of the actual occurrence of the load. The forecasting intervals are basically the time intervals for which the load is forecasted. Based on the forecast intervals the nature of load forecasting can be classified as very short term (few seconds to several minutes), short term (half an hour to a few hours), medium term (ranging from days to weeks) and long term (months to years) [33]. In order to maximize financial benefit a good load forecasting method is required that is capable of reflecting the current and future trends. Based on historical data and statistical techniques short term load forecasting (STLF) can be performed.

2.11 Levelized Cost of Energy

This section describes the economic model used to evaluate the system. For every developing technology a key aspect that researchers and industries look into is the feasibility of
the technology. Hence engineering economics is a key area of focus. The feasibility of the potential solutions that engineers develop is evaluated along with the technical aspects. In the field of power generation the Levelized Cost of Energy (LCOE) is the mathematical quantity that can be used to compare the relative cost of energy production from a generating source.

LCOE can be considered as a measure summarizing the overall competitiveness of the various generating technologies. It is the cost, per kilowatt-hour in dollars, of constructing, operating and maintaining a generating plant over an assumed lifetime of the plant. Essential inputs to calculating LCOE include capital costs, fixed and variable operations and maintenance (O&M) costs, financing costs, plant factors, fuel costs. These factors vary with each plant type. The influence of each of the factors on LCOE varies for different generating technologies. Technologies such as hydro and solar have no fuel costs and hence have relatively small variable O&M costs. In such cases the LCOE is affected by the estimated capital cost and generation capacity. For technologies with significant fuel cost, such as conventional generating units, the capital cost and fuel cost affect LCOE. The availability of various incentives, including state or federal tax credits, can also impact the calculation of LCOE. Each of these factors and their values vary with location and time as technologies evolve and fuel prices change.

The LCOE equation allows a comparison of alternative technologies for different scales of operation, operating time periods or investment [34]. Using this motivation the LCOE has been used as a reference model for the economical evaluation of the potential solutions proposed in this work.

2.11.1 Factors affecting LCOE

*Capital Investment cost:* The major cost components affecting the generating technologies are discussed as follows
i) Construction costs: These are normally the major costs of any project. The civil construction costs are determined by the trends in prices of the country where the project is going to be developed. In countries undergoing economic transition the civil construction costs are generally lower than their developed counterparts. This is due to deployment of local construction materials and local labor. Construction costs are affected by the inherent characteristics of the topography, geological weather conditions and the construction design of the project. Hence the construction costs are site specific. This could potentially lead to variation in investment cost and consequently the LCOE, even for projects having the same capacity.

ii) Cost of power system equipment: The costs of electromechanical equipment follow world market prices for these components as opposed to construction costs. The type of technology used for generation is significant component. For example, in a solar power plant the type of solar module used affects the investment cost since it requires the installation of the necessary equipment. Additionally the investment costs also cover planning costs, licensing, environmental impact analysis, wildlife mitigation, historical and archaeological mitigation, recreation mitigation.

Studies conducted in [35] show that there is a general tendency of increasing investment cost as the capacity increases and that there is also a wide range of cost for projects of the same capacity. For plants with lower installed capacity, for example 5 MW, it is the costs of electromechanical equipment that tend be more significant. However, with an increase in plant capacity, the costs of construction are more influential. The same overall generating capacity can be achieved through a combination of several smaller or large generating units. Plants
implementing multiple smaller generating units have higher costs per kW as compared to plants using fewer larger units. The reason of having higher costs per kW associated with a higher number of generating units is because of a greater flexibility and efficiency achieved by the integration of plants into the electric grid. For plants that have fewer units with larger generating capacity the investment costs tend to be reduced. For example, for a larger generating unit, the hydropower project can be set up to use less volume flow, and consequently smaller hydraulic conduits. The size of the equipments and their related costs are also lower. Results published in [35] have shown the characteristic distribution of investment costs based on geographic areas.

*Operation and Maintenance cost:* Renewable energy technologies, after commissioning, generally require minimal operation and maintenance. A major reason is that plants operating with such technologies do not have to incur the cost of fuel continuously. O&M costs can be calculated as a percentage of the investment cost per kW [35]. In contrast conventional generating units such as fuel base units have a higher Operation and Maintenance cost due to recurring cost of fuel. This again varies with location and market conditions.

*Capacity Factor:* The capacity factor of a power plant is the ratio of its actual power to the output it could produce if it were operating at full nameplate or rated capacity continuously over a given time period. The parameters affecting capacity factors is mainly the availability of the renewable resources. This in turn depends on the location and the type of technology used in implementing the technology. For hydropower, the capacity factor is generally designed during the planning and optimization stages of the project, by considering both the market demand for power and the statistical distribution of flow. Reservoirs can not only be designed to increase the stability of flow for base-load production, but also for supplying highly variable and reliable flow to a power plant. For solar power the main parameter affects that capacity factor is the annual solar radiation at a given location and the type of panel or module used. The type of module affects the efficiency of conversion of light to electricity which affects the actual output of the
plant and hence the capacity factor. A low capacity factor implies low production and higher LCOE. Overall the capacity factors of renewable energy technologies is lower compared to the that of conventional generating units. This is due to the intermittency or unavailability of the renewable energy sources at all times.

*Lifetime:* For renewable energy plants, the civil structures used have very long lifetimes, like dams, tunnels, canals, powerhouses for hydropower or the components in Concentrated Solar Power (CSP) technology for solar are the major cost components. Electrical and mechanical equipment, with comparatively shorter lifetimes contribute less to the cost. Hence it is common to use longer lifetimes for hydropower plants as compared to other electricity generation sources. Typical lifetimes are in the range of 40 to 80 years.

*Discount Rate:* The discount rate can influence the LCOE based on the trend of expenditures and revenues that occur over the lifetime of the investment. Investors generally choose discount rates based on the risk-return characteristics of available investment options. A high discount rate is beneficial for technologies that have low initial capital investment and high running costs. On the other hand, a low discount rate is conducive for Renewable Energy sources such as hydropower and solar power which have comparatively higher capital investment and lower maintenance costs. Typical values for renewable energy technologies are 7% and 10% [35].

### 2.11.2 LCOE calculations

The LCOE equation depends on the capital cost ($/kWh), discount rate, O&M cost ($/kW-yr), Capacity Factor (%), Fuel Cost($/MMBtu), Heat Rate(Btu/kWh) and period (years).

The discount rate may be real or nominal. Using the values of discount rate and lifetime periods the capital recovery factor (CRF) is calculated. A capital recovery factor is the ratio of a constant annuity to the present value of receiving that annuity for a given period of time.
Using an interest rate $i$, the capital recovery factor (CRF) is calculated as shown in [25] and represented in equation 2.1.

$$\text{CRF} = \frac{i(1+i)^n}{(1+i)^n - 1} \quad (2.1)$$

Here $n$ is the plant lifetime in years and $i$ is the discount rate.

Based on definitions in [36] [37] is calculated as shown in equation 2.2:

$$\text{LCOE} = \frac{\text{capital cost} \times \text{CRF} + \text{Fixed O&M}}{8760 \times \text{capacity factor}} + (\text{fuel cost} \times \text{Heat Rate}) + \text{Variable O&M cost} \quad (2.2)$$

where variable O&M costs are in dollars per kilowatt-hour ($/kWh$).

8760 in the denominator represents the number of hours in a year and capacity factor is a fraction between 0 and 1 representing the portion of a year that the power plant is generating power.

The heat rate describes the amount of fuel burned for each unit of electricity produced. Typical values are 9000-10,000 Btu/kWh for utility sized grids. Distributed generation systems of lower capacity have a value between 12,000-13000 Btu/kWh [36]. It also varies with the type of fuel used such as natural gas, coal and fossil fuels.

Fuel cost is expressed in dollars per million British Thermal Units ($/MMBtu$) and heat rate is measured in British Thermal Units per kilowatt-hour (Btu/kWh). Being a based factor it varies with location and time. For renewable generating technologies like hydropower and solar the fuel cost is zero since they do not run on fuel.
Chapter 3

Software Tools, System Overview and Application Development

This chapter describes the Function Block software tool, FBDK, followed by the system overview, an analytical description and the flow chart of the control algorithm used for economic dispatch. Subsequently the list of Function Blocks used in the application development in FBDK are described.

3.1 Function Block Development Kit

FBDK is a software tool that allows the building and testing of function block types, data types, adapter types, functions, resource types, device types, network segment types and system configurations according to the IEC 61499 standard. As discussed in chapter 2 Section 2.4 the main types of a Function Blocks are: Basic and Composite FB and SIFB.

The procedure for creating a Function Block in FBDK is elaborately described in Appendix A. Through the procedure of creating a Function Block in FBDK the concepts can be better understood.

3.2 System Overview

To test the economic dispatch applications, Simulink is used to develop the time domain model of the power system network. The power system configuration and parameters have been extracted from the benchmark system of the IEEE Standard 399-1997 with some modifications to allow for autonomous microgrid operation [9]. Figure 3-1 shows a single line diagram of the
Figure 3-1 Single Line Diagram
system developed. The system consists of a three feeder distribution subsystem operating at 13.8 kV. This subsystem is connected to the main network or grid through a radial line. The large network at the end of the radial line is represented by a 69 kV, 1000MVA short circuit capacity bus. Each of the three feeders is connected to the large network at one end and a Distributed Generator(DG) source and a combination of variable loads at the other. Since the focus is on observing the behavior of the load balancing applications during normal conditions, each of the DG is modeled to operate at steady state. The system includes three DGs- DG1, DG2, DG3, each operating at 13.8kV. DG1 is a conventional unit running on natural gas capable of supplying 4MW. DG2 is modeled as a Photo Voltaic (PV) plant of capacity 4MW. DG3 is a hydro power plant capable of supplying 3 MW.

Thus a microgrid collectively has a DG, two loads and corresponding distribution network components. Each DG has a Levelized Cost of Energy that is calculated using an economic model. Loads L1- L6 are supplied through the radial feeders of the subsystem. Each of the loads varies over time depicting a variable load profile. The simulation is run for twelve seconds, representing a 24 hour day.

3.2.1 Load forecasting development

The load profile is developed using three parameters: a) Power required by the load in kW; b) The start time of the load in seconds; and c) The duration of the load.

For a given load, a varying load profile is simulated in Simulink on which the load balancing applications are tested. The variation in loads is achieved using random number generation for the start time and duration for each of the loads. Each of these loads have values
Figure 3-2 Simulink model of variable load profile
over a certain range- 50% of full load to full load. These loads are connected to circuit breakers whose opening and closing times are controlled through the random number generator in MATLAB. Figure 3-2 shows the simulation model implemented to achieve a varying load profile.

Fig 3-3 shows a screen shot of a circuit breaker controlling a load in microgrid 1. The terms tLA1 and tLA2 are variables storing numbers generated using random number generators. They contain the time values for the switching times of the circuit breakers connected to Load 1.

Figure 3-3 Circuit breaker dialogue box
These values are in seconds. Hence the particular load under consideration has a start time of $t_{A1}$ and a duration of $(t_{A2}-t_{A1})$. The program for the random number generation is shown in the appendix.

The loads profiles for the microgrids are shown from Table 3-1 to Table 3-6. Each load has a threshold value that is calculated as the average of the forecasted load profile. The units of each load are in MVA.

**Table 3-1 Profile for Load 1**

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Time Start</th>
<th>Time Duration</th>
<th>Load: MVA $(P+jQ)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{th1}$</td>
<td>0</td>
<td>$t_{A1}$</td>
<td>3.26+j3.84</td>
</tr>
<tr>
<td></td>
<td>$t_{A1}$</td>
<td>$t_{A2}-t_{A1}$</td>
<td>4+j3</td>
</tr>
<tr>
<td></td>
<td>$t_{A2}$</td>
<td>$t_{A3}-t_{A2}$</td>
<td>3.64+j3.428</td>
</tr>
<tr>
<td></td>
<td>$t_{A3}$</td>
<td>12-$t_{A3}$</td>
<td>3.8-j3.25</td>
</tr>
</tbody>
</table>

**Table 3-2 Load profile for Load 2**

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Time Start</th>
<th>Time Duration</th>
<th>Load: MVA $(P+jQ)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{th2}$</td>
<td>0</td>
<td>$t_{B1}$</td>
<td>3.6+j3.47</td>
</tr>
<tr>
<td></td>
<td>$t_{B1}$</td>
<td>$t_{B2}-t_{B1}$</td>
<td>2.6+j4.27</td>
</tr>
<tr>
<td></td>
<td>$t_{B2}$</td>
<td>$t_{B3}-t_{B2}$</td>
<td>3+j4</td>
</tr>
<tr>
<td></td>
<td>$t_{B3}$</td>
<td>$t_{B4}-t_{B3}$</td>
<td>3.4+j3.67</td>
</tr>
<tr>
<td></td>
<td>$t_{B4}$</td>
<td>12-$t_{B4}$</td>
<td>3.8+j3.25</td>
</tr>
</tbody>
</table>
### Table 3-3 Profile for Load 3

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Time Start (seconds)</th>
<th>Time Duration (seconds)</th>
<th>Load: MVA (P+jQ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pth3= 3.872 MW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>tIC1</td>
<td></td>
<td>3.64+j3.43</td>
</tr>
<tr>
<td>tIC1</td>
<td>tIC2-tIC1</td>
<td></td>
<td>4.2+j2.71</td>
</tr>
<tr>
<td>tIC2</td>
<td>tIC3-tIC2</td>
<td></td>
<td>4+j3</td>
</tr>
<tr>
<td>tIC3</td>
<td>tIC4-tIC3</td>
<td></td>
<td>3.8+j3.25</td>
</tr>
<tr>
<td>tIC4</td>
<td>12-tIC4</td>
<td></td>
<td>3.72+j3.34</td>
</tr>
</tbody>
</table>

### Table 3-4 Profile for Load 4

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Time Start (seconds)</th>
<th>Time Duration (seconds)</th>
<th>Load: MVA (P+jQ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pth4= 3.835MW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>tID1</td>
<td></td>
<td>3.95+j3.065</td>
</tr>
<tr>
<td>tID1</td>
<td>tID2-tID1</td>
<td></td>
<td>3.75+j3.3</td>
</tr>
<tr>
<td>tID2</td>
<td>tID3-tID2</td>
<td></td>
<td>3.64+j3.428</td>
</tr>
<tr>
<td>tID3</td>
<td>12-tID3</td>
<td></td>
<td>4+j3</td>
</tr>
</tbody>
</table>
### Table 3-5 Profile for Load 5

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Time Start</th>
<th>Time Duration</th>
<th>Load: MVA (P+jQ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pth5= 1.75MW</td>
<td>0</td>
<td>tlE1</td>
<td>1.6+j1.92</td>
</tr>
<tr>
<td></td>
<td>tlE1</td>
<td>tlE2-tlE1</td>
<td>2+j1.5</td>
</tr>
<tr>
<td></td>
<td>tlE2</td>
<td>tlE3-tlE2</td>
<td>1.85+j1.68</td>
</tr>
<tr>
<td></td>
<td>tlE3</td>
<td>12-tlE3</td>
<td>1.55+j1.96</td>
</tr>
</tbody>
</table>

### Table 3-6 Profile for Load 6

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Time Start</th>
<th>Time Duration</th>
<th>Load: MVA (P+jQ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pth6= 1.85MW</td>
<td>0</td>
<td>tlf1</td>
<td>1.7+j1.83</td>
</tr>
<tr>
<td></td>
<td>tlf1</td>
<td>tlf2-tlf1</td>
<td>2+j1.5</td>
</tr>
<tr>
<td></td>
<td>tlf2</td>
<td>tlf3-tlf2</td>
<td>2.1+j1.356</td>
</tr>
<tr>
<td></td>
<td>tlf3</td>
<td>12-tlf3</td>
<td>1.6+j1.92</td>
</tr>
</tbody>
</table>
3.3 Application Development

This section describes the Economic Dispatch applications developed in FBDK. The application development has been divided into multiple stages and each stage is associated with a load balancing application exhibiting a different behavior. The effectiveness of each of the applications is depicted by comparing the cost of operation for 24 hours for the entire network. The objective of each of the load balancing applications is to supply the loads using DGs in the network in combination with the utility in a cost effective manner. The criteria for economic dispatch is based on availability of generation and cost of generation, including the constraints that affect both these parameters. The applications are developed using the FB concept on FBDK. The loads are monitored through Smart Metering Devices (SMD). These devices are addressable through IPs assigned to them by the Control Center. The time variation of the loads is generated through random number generation in MATLAB as discussed in Section 3.2.1. This section describes the economic constraints considered in modeling the generating sources that are used in economic dispatch.

3.3.1 Economic Modeling of the generating sources

The economics of each of the generating technologies depend on the Investment cost, capacity factor, fuel cost and heat rate of the fuel. These parameters, which are a part of the LCOE equation, are used to calculate the LCOE for the different generating technologies. The LCOE is then implemented in the economic dispatch application. This section describes the science, reasoning and assumptions behind considering the values for the different parameters of the LCOE equation.
The economic model used in this thesis, based on the LCOE, provides a method to compare the cost of energy across technologies by taking into account the installed system price and associated costs such as fuel cost, fixed and variable operation and maintenance costs along with performance [36]. In order to be able to develop an economic dispatch application of practical significance, the estimates for the parameters on which economic model is based are extracted from the Transparent Cost Database [38]. The Transparent Cost Database collects program cost and performance estimates for U.S Department of Energy Office of Energy Efficiency and Renewable Energy (U.S. DOE EERE) technologies on a public platform where they can be viewed and compared with estimates in other publications. The estimates used in this work are the result of a detailed data analysis conducted by US Energy Information Administration (EIA), a part of the US Department of Energy (DOE) [39] [36]. While implementing new energy technologies, they have to overcome competition from current technology solutions in terms of price. This ensures their large scale market penetration. Therefore EERE solutions are designed with the aim of reducing costs. For multiple technologies within DOE’s Office of Energy Efficiency and Renewable Energy, the road mapping process results in the development of a series of engineering-based cost reduction steps that help achieve the overall cost goals. Thus the economic model used for the generating technologies represents near real-world market conditions and is expected to have practical significance since it is aimed at achieving a cost benefit.

The LCOE equation implemented in this work is dependent on the factors of Capital Cost, Capacity Factor, Fixed Operation and Maintenance Cost, Variable Operation and Maintenance cost, Fuel Cost and Heat Rate. Based on the type of generating technology used, each of the parameters are affected by different factors. The generating technologies implemented in this work are hydro power, conventional fuel generator (Natural Gas Combined Cycle type), solar power.
For the scenario considered in this thesis it has been assumed that the power system network is based in the Pacific region of United States. This assumption significantly affects the investment costs and the fuel cost for conventional generating units. The fuel costs in turn affect the operation and maintenance costs. The location also affects the capacity factor of the renewable energy resources. For each of the given systems it has been considered that the intermittency has been accounted for with sufficient storage capacity in the plant. This is also reflected in the investment and O&M costs.

### 3.3.1.1 Modeling DG1

DG1 is modeled as a conventional generating unit running on natural gas. Being a conventional generating unit it will have the recurring cost of fuel. This cost will depend on the location in which it is set up. Natural gas as a resource does not depend on the weather like solar or hydro. Hence it inherently has a higher capacity factor. The location here essentially affects the price of fuel and in turn the investment cost and the operation and maintenance cost. Being a conventional generating unit the discount rate is higher. For the region under consideration the parameters under consideration are [40] [41]:

**Table 3-7 LCOE equation parameters for DG1**

<table>
<thead>
<tr>
<th></th>
<th>DG1 - Natural Gas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital Investment Cost ($/kW)</td>
<td></td>
</tr>
<tr>
<td>Period (years)</td>
<td></td>
</tr>
<tr>
<td>Discount rate (%)</td>
<td></td>
</tr>
<tr>
<td>Capacity Factor (%)</td>
<td></td>
</tr>
<tr>
<td>O&amp;M costs ($/kW-yr)</td>
<td></td>
</tr>
<tr>
<td>Variable O&amp;M costs ($/kWh)</td>
<td></td>
</tr>
<tr>
<td>Heat Rate (Btu/kWh)</td>
<td></td>
</tr>
<tr>
<td>Fuel Costs ($/MMBtu)</td>
<td></td>
</tr>
<tr>
<td>LCOE (cents/kWh)</td>
<td></td>
</tr>
<tr>
<td>1090</td>
<td>30</td>
</tr>
</tbody>
</table>
The fuel cost is considered for the Pacific region and the heat rate is considered based on studies conducted by EIA [41].

3.3.1.2 Modeling DG2

DG2 is modeled as a solar plant of capacity 5 MVA. The plant economics depend on the Capital investment cost, Capacity Factor, Operation and Maintenance costs and the period of plant operation in years. The capital investment costs for the PV system covers the costs of panels, inverter, and associated hardware costs. It also includes the non hardware costs such as labor and associated non-hardware costs that are region specific. It has been assumed that the PV panel used here is fixed and of thin film type [42]. The type of panel also affects the capacity factor of the DG [43]. The O&M costs are considered based on the study considered in [36] and are mainly concerned with equipment upgrade costs. With these considerations the values used in calculating the LCOE for the PV plant is shown in Table 3-9 [44] [43] [39] [42] [40] [36].

<table>
<thead>
<tr>
<th>Capital Investment Cost ($/kW)</th>
<th>Period (years)</th>
<th>Discount rate (%)</th>
<th>Capacity Factor (%)</th>
<th>O&amp;M costs ($/kW-yr)</th>
<th>Variable O &amp; M costs ($/kWh)</th>
<th>Heat Rate</th>
<th>Fuel Costs</th>
<th>LCOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2600</td>
<td>30</td>
<td>7</td>
<td>25</td>
<td>7.56</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>10.1</td>
</tr>
</tbody>
</table>
3.3.1.3 Modeling DG3

DG3 is modeled as a hydro power plant capable of supplying 3 MW. The data for the costs of the plant were based on historical data analyzed by the Idaho National Engineering and Environmental Laboratory during a study jointly funded by the U.S. Department of Energy (DOE) Office of Energy Efficiency and Renewable Energy and Energy Information Administration [37] [39]. The study was conducted with a purpose of providing the public, policy makers and the DOE with a document containing contemporary estimates of the potential hydroelectricity supply in the USA along with the associated costs [45]. The estimation of the cost of capital investment costs and operating and maintenance costs of hydropower resources were developed in the form of regression curves using historical plant data. Each of the estimates were found to be a function of plant capacity. The development costs were derived considering licensing, construction, and environmental mitigation, recreation mitigation, water quality monitoring costs. While developing the estimates for the Operation and Maintenance costs, the costs of upgrading the turbines and the cost of upgrading the generator are considered. For a hydro power plant, apart from the capacity, the hydraulic head of the turbine and the rotational speed for the generators were important factors [45]. The head is essentially the height from which the water falls. Water sources located at different locations will have different heads. With these considerations the plant factors were estimated for various regions in the United States. For the scenario under consideration it is assumed that the plant is built on a site in the Pacific region that has sufficient storage capacity to supply rated power. Hence the parameters are calculated taking into consideration all these assumptions, and are tabulated in Table 3-9 [40] [37].
Table 3-9 LCOE equation parameters for DG3

<table>
<thead>
<tr>
<th>DG3</th>
<th>Capital Investment Cost ($/kW)</th>
<th>Period (years)</th>
<th>Discount rate (%)</th>
<th>Capacity Factor (%)</th>
<th>Fixed O&amp;M costs ($/kW-yr)</th>
<th>Variable O&amp;M costs ($/kWh)</th>
<th>Heat Rate (Btu/kWh)</th>
<th>Fuel Costs ($/MMBtu)</th>
<th>LCOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2359.3</td>
<td>30</td>
<td>7</td>
<td>57.5</td>
<td>11.3975</td>
<td>0.0036</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.4</td>
</tr>
</tbody>
</table>

3.3.1.4 Determining cost of Large Scale Utility network

This cost is the retail price of electricity provided by the electricity service provider. It is the average annualized retail rate for the electric utility that serves a particular region based on the kind of load, residential or industrial. In the United States, the rate making is decided by a State Governing Authority or by the Federal Energy Regulatory Commission where matters of inter-state power exchange are concerned. Rate deciding attempts to be efficient, competitive and at the same time fair across the different classes of consumers. Hence they have a economical and political dimension to it. At the core rate, making policy is affected by factors such as capital attraction, nature of consumers and demand control. In the USA the majority of the utilities are investor owned. Hence government regulators must assure the private companies a fair return in order to attract more capital. At the same time they have to keep in mind the demand and the nature the of the loads (residential, commercial and industrial) [46]. Also there has to be a consideration for the utilities operating expenses, wages, salaries, tax maintenance. A study conducted by Ventyx Research Inc and Energy Information Administration (EIA) considers the data trends for Utility prices based on location [47]. The study considered the nature of loads, the
number of utilities in a particular region, the price for commercial loads and was conducted in the year 2012. Based on the factors described the cost of utility price is taken to be 14 cents/kWh.

### 3.4 Linking the concepts

In Chapter 2 the concepts of Function Block, Economic Dispatch, Load Forecasting and Levelized Cost of Energy have been explained independently. As discussed in Section 2.8 Economic Dispatch is broadly focused on four major areas. The contribution of this work is to combine these concepts together by proposing an Economic Dispatch solution that combines the features of dynamic dispatch or load forecasting and dispatch using renewable generation sources. Hence the objective function will aim at minimizing the cost of power generation, by considering the cost constraints for renewable energy systems and the feature of load forecasting. The entire application is implemented in the Function Block environment that forms the basis of the IEC 61499 standard thereby proposing an extension in the range of Function Block applications and catering to the need for flexible control systems in a smart grid.

### 3.5 Simulation Test Stages

This section describes the test scenarios considered during the simulation.

#### 3.5.1 Stage 1 Normal grid operation

In this case the loads are supplied power by their respective microgrid DGs and the utility without load balancing consideration. The power supplied to loads is shared by generators based on their impedances.
3.5.2 Stage 2: Single microgrid Economic Dispatch

The application developed in this stage involves interaction between the utility and the DG associated with loads to be balanced. Each load is assigned a threshold value that is calculated as the mean of the load profile. Any increase in load above the threshold limit, in a particular microgrid is shared only between the utility and the DG present in that microgrid [28]. For example, if there is an increase in the load L1, the necessary control action chooses between either the utility or a combination of the utility and DG1 to balance the load. The application is developed on one device with one resource. The control architecture implemented in this stage is shown in Figure 3-4. The criteria for load balancing is based on only the availability of a DG to match power and the power supplied by the Utility network. The flowchart of the algorithm is shown in Figure 3-5.

Each grid has two loads associated with it. The power consumed by each of the loads is \( P(1), P(2), P(3), P(4), P(5) \) and \( P(6) \). The utility is connected to the grids through CBs labeled CB(1), CB(2) and CB(3) respectively for grid 1, 2 and 3. Each load has a pre-determined threshold value (\( P_{th1}, P_{th2}, .. P_{th6} \)). The individual threshold values are calculated as the mean of the varying load profiles exhibited by each of the loads. The attributes of a load are received at the CCC. The value of the power consumed in a grid is compared with the threshold for that particular grid. If the power consumed exceeds the threshold for even one of the loads in the microgrid then the CB between the utility and the particular microgrid containing that load is closed. This ensures that the necessary power is supplied to the load in an efficient way and the load is hence balanced. The described algorithm is executed on one device.
Figure 3-4: Stage 2

The flowchart in the Figure 3-5 gives a diagrammatic explanation of the control logic implemented.
Figure 3-5 Flowchart for algorithm in Stage 2

START

Grid 1: Load 1: Input Power, Time start, Load duration

If Power Input1 > Threshold 1 and Time < Time start + duration
  Yes
  Close CB for Grid 1
  No
  Open CB for Grid 1

Grid 2: Load 2: Input Power, Time start, Load duration

If Power Input2 > Threshold 2 and Time < Time start + duration
  Yes
  Close CB for Grid 2
  No
  Open CB for Grid 2

If Power Input2 > Threshold 3 and Time < Time start + duration
  Yes
  Close CB for Grid 2
  No
  Send value to Grid CBs

Grid 3: Load 3: Input Power, Time start, Load duration

If Power Input3 > Threshold 5 and Time < Time start + duration
  Yes
  Close CB for Grid 3
  No
  Open CB for Grid 3

If Power Input3 > Threshold 6 and Time < Time start + duration
  Yes
  Close CB for Grid 3
  No
The list of Function Blocks used in this case with their description is given in Table 3-10.

Table 3-10 Function Blocks for Stage 2

<table>
<thead>
<tr>
<th>Event</th>
<th>Req</th>
<th>Cnf</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>INT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNF</td>
</tr>
<tr>
<td>EVENT</td>
<td></td>
<td></td>
<td>EVENT</td>
</tr>
</tbody>
</table>

This FB acts as the interface in accepting data from the SMD in Simulink. The input to this block is the IP address of the Metering device and the guard condition which has a Boolean value of 1. The FB execution gets triggered only when it receives an event input trigger at INIT. The values LREAL, TIME are the data types of the values received from the SMD. They are essentially the Power value, start time and duration of load respectively.

This FB essentially executes the comparison of the loads with their threshold for a single grid. The output of this FB is a signal that is communicated to the CB connecting the grid to the concerned microgrid. If the Loads are above the threshold the output CB_SEL is a value of 1 and if less than the threshold the output has a value of 0.
The behavior exhibited in Stage 2 is such that, in the event of any load going above threshold, the only source of secondary generation to balance the load is the Utility. However the behavior can be improved by considering the presence of other secondary sources of generation that are capable of supplying the excess demand and at the same time are cheaper than the Utility. The improved behavior has the potential of reducing the cost of operation further and has been implemented by the inter microgrid economic dispatch application developed in Stage 3.

### 3.5.3 Stage 3- Inter microgrid Economic Dispatch

The application developed in this stage is an Economic Dispatch application and involves lateral interaction between the microgrids. Any increase in load in a particular microgrid can be shared by the utility and the other microgrids in the network. The application has been divided into multiple sub-applications and implemented on multiple devices. The control architecture is shown in Figure 3-6. The criteria for Economic Dispatch is based on availability of power, the cost of generation and the forecasted load. This Stage is an example of a distributed application, i.e., a single application distributed into multiple sub-applications on multiple devices and hence can be considered as an application being implemented on remote devices.
3.5.3.1 Mathematical Formulation

This section describes the mathematical formulation of the economic dispatch application developed. As described in Chapter 2, at the core of every economic dispatch application is an objective function that aims at supplying loads in a cost effective manner. The objective function used in this work is formulated as:

\[
\text{Objective Function} = \min \sum_{i=1}^{n} c_i x_i
\]

where \( c_i \) are the cost coefficients and \( x_i \) are the decision variables representing the power outputs of the generators.
\[ [P_{\text{set-point}}, C_{\text{min}}] = f(P_{\text{load}}, \text{Time}_{\text{start}}, \text{Time}_{\text{duration}}, \text{LCOE}, P_{Gi}, P_{th}) \]  \hspace{1cm} (3.1)

Subject to:

\[ L_{\text{min},i} \leq P_{Gi} \leq P_{Gi,\text{max}} \]  \hspace{1cm} (3.2)

\[ \sum P_{Gi} = \sum P_{\text{load}}; \text{ i represents the } i^{th} \text{ generating source} \]  \hspace{1cm} (3.3)

\[ P_{Gi,\text{max}} = \min(P_{Gi,\text{rated}}, P_{th}) \]  \hspace{1cm} (3.4)

\[ P_{th} = \sum_{j=1}^{K} \text{mean}(P_{\text{load},j}); \quad K = \text{number of loads in a micro grid} \]  \hspace{1cm} (3.5)

\[ P_{Gi,\text{rated}} = \text{rated capacity of generator } i; \]  \hspace{1cm} (3.6)

\[ L_{\text{min}} = \min(P_{\text{load}}) \]  \hspace{1cm} (3.7)

Where:

\[ C_{\text{min}} = \min \sum_{i=1}^{N} \text{LCOE} \times E_{Gi}; \]  \hspace{1cm} (3.8)

\[ N = \text{number of generating sources} \]

\[ E_{Gi} = \text{Energy output of generator based on forecasted } P_{\text{load}} \]

\[ E_{Gi} = P_{Gi} \times 1 \text{ hr} \]  \hspace{1cm} (3.9)

\[ P_{\text{load}} = \text{load demand (MW)} \]

\[ \text{Time}_{\text{start}} = \text{Start time of load} \]

\[ \text{Time}_{\text{duration}} = \text{duration of load} \]

\[ \text{LCOE} = \text{Levelized cost of energy ($/\text{MWh})} \]

\[ \text{LCOE} = \frac{\text{Capital Investment cost ($/\text{kW})} \times \text{Capital recovery factor} + \text{fixed O&M cost ($/\text{kW-yr})}}{8760 \times \text{capacity factor}} + \left\{ \frac{\text{fuel cost ($/\text{MMBtu})} \times \text{heat rate (Btu/kWh)}}{\text{variable O&M cost ($/\text{kw-h})}} \right\} \]  \hspace{1cm} (3.9)
Figure 3-7 shows a block diagram of the economic dispatch application implemented in this work.

The proposition here is to minimize the cost of inter-operation of microgrids. Based on the economic, load and availability constraints of the system, which go as input to the control system, the output of the system is the minimum cost of operation and the appropriate control signals for the circuit breakers to the microgrid DGs for supplying loads. Here the inputs to the system are the forecasted load demand ($P_{\text{Load}}$), load start time and duration ($\text{Time}_{\text{Start}}$, $\text{Time}_{\text{duration}}$), and the Levelized Cost of Energy. The load start time and duration are generated by a random number generator following uniform distribution to create a load profile that acts as the forecast to the control system. The cost function for the renewable energy systems is based on the Levelized Cost of Energy which considers the Investment cost of equipment, Capital cost, O&M
Thus the economic dispatch solution combines dynamic dispatch and load forecasting to provide a cost effective solution.

Table 3-11 shows the essential Function Blocks used in this Stage including the Function Blocks included in Stage 2.

**Table 3-11 Function Blocks for Stage 3**

<table>
<thead>
<tr>
<th>Function Blocks for Stage 3</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>This FB is present in device 4. It takes as input the CB signals from the first three devices 2 at a time. The CB will close if even one of the inputs goes to 1.</td>
<td>This FB performs the comparison algorithm. The inputs CostX, CostY and CostUTIL are the LCOEs of the DGs from which the concerned grid might take power. After performing the Power Comparison Algorithm the FB sends out three outputs for three CBs. For example if we consider grid 1 as the grid demanding power. Then CB_A_B is the CB connecting grid 1 and grid 2. CB_B_C is the CB connecting grid 1 and grid 3. CB_UTIL is the CB connecting grid 1 and the Utility. 1 implies CB close and 0 implies CB open. The other inputs are the instantaneous load values, their duration and threshold limits.</td>
</tr>
</tbody>
</table>
Similar to Stage 2 each load has a pre-determined threshold value (Pth1, Pth2, .. Pth6). Each Microgrid is controlled by a Local Control Center (LCC) and is responsible for performing the economic dispatch application in respective grids. Hence there are three local control centers in total. These Local Control Centers then communicate their decisions, to open or close the breakers, to a Central Control Center which then sends the trip signal to the appropriate CB. Each of the three grids is connected to each other through a circuit breaker. The CB between grid 1 and grid 2 is CB\textsubscript{grid1-grid2}, CB between grid 2 and grid 3 is CB\textsubscript{grid2-grid3}, and CB between grid 1 and grid 3 is CB\textsubscript{grid1-grid3}. In addition to these CBs there are also three CBs connecting each of the grids to the main utility. These are labeled CB\textsubscript{Utility1}, CB\textsubscript{Utility2} and CB\textsubscript{Utility3} respectively for grid 1, 2 and 3. These CBs are controlled by a Central Command Center (CCC) after receiving signals from the LCC. Each load has a particular threshold value. When a particular load in a microgrid exceeds the threshold value that grid looks for generating sources that offer power at the lowest price. If any of the microgrids has excess capacity then it supplies it to the grid in deficit. If no DG is able to supply the required amount of power entirely then the power is supplied through a combination of DGs and the main utility. If the load(s) cannot be supplied by the microgrid DGs at all, only then is the deficit supplied by the main grid entirely. Based on which DG is scheduled to supply the power the corresponding breaker is closed thus ensuring the necessary amount of power supply to the load. The flow chart for the economic dispatch is shown in Figure 3-7.
Figure 3-7 Flowchart for Economic Dispatch
Figure 3.7 Flowchart for Economic Dispatch
Figure 3.7 Flowchart for Economic Dispatch
Figure 3-7 Flowchart for Economic Dispatch
Figure 3-7 Flowchart for Economic Dispatch
Figure 3-7 Flowchart for Economic Dispatch
3.6 Co-simulation between MATLAB and FBDK

The entire execution of the application has been executed through a co-simulation model between MATLAB and FBDK. In order to achieve successful co-simulation it must be ensured that both the software tools are able to exchange data in a language easily understood by both. The logic is quite analogous to real world communication between two people. In order for two people to successfully convey their ideas both of them have to ensure that they present their ideas in a way understood by the other. This is possible when they both communicate in the same language (say English). Similarly before sending data from MATLAB into Function Block Development Kit, it has to be encoded in a manner understood by the latter. That is to say the data and variables have to be encoded into data types supported and understood by Function Block Development Kit. This has been the underlying principle of co-simulation followed in this work.

The co-simulation has been carried out through a simple approach using the class files used to create the Function Blocks in FBDK. Co-simulation has been performed using the fact that FBDK is a Java based program. Hence each Function Block consists of class file. These Java class files can be put into a single package file format called Java Archive (Jar) file. A JAR file can be viewed as a collection or library of Java class files. A unique feature of MATLAB is that it allows the addition of java library files when they are compressed into a jar file. The jar files once created can be added into MATLAB through the command window. The addition of the jar files implies the addition of the class files into MATLAB. Hence instances or objects of these classes can be included in programs in MATLAB and Simulink.

Co-simulation is done using TCP/IP. It can be viewed as a communication between a SERVER and a CLIENT. Since MATLAB is the side initiating the communication it is the
SERVER and FBDK is the CLIENT. Hence class files of SERVER FBs are included in MATLAB.

The procedure to include the class file of a SERVER Function Block exchanging Time values is as follows:

In the folder/package where the file is stored the command prompt is opened. The screenshot of the Command prompt window looks as shown in Figure 3-8.

![Figure 3-8 Command prompt](image)

The java program is then compiled and run typing the following commands in the command window:

```
javac –cp fbrt.jar –d . TIME_SENDER.java
java –cp fbrt.jar;. time.server.TIME_SENDER
```

Here the name of the java file is TIME_SENDER. On executing the above two commands the file TIME_SENDER gets compiled.

The JAR file is then created by typing the following command:

```
jar cvfm time.jar manifest.txt time fbrt.jar.
```
The result of this command is to create a JAR file named time.jar. This file depends on the files fbrt.jar and manifest.txt. Hence before executing the above commands it should be ensured that the file fbrt.jar is present along with the file manifest.txt. The contents of manifest.txt file are:

*Main-Class: time.server.TIME_SENDER*

*Class-Path: fbrt.jar*

Once the jar file has been created the essential jar files are added into MATLAB by typing the following commands into the command window:

```
>> javaaddpath J:\FBDK_JAVA\TRIAL1\src\time\server\time.jar
>> javaaddpath J:\FBDK_JAVA\TRIAL1\src\time\server\fbrt.jar
```

These two commands ensure the addition of the jar files into MATLAB. Every time the MATLAB session is started the javaaddpath commands have to be typed to include the JAR files in the session.

To create instances of the class TIME_SENDER in MATLAB the following command needs to be typed:

```matlab
Time1 = TIME_SENDER();
```

This line creates an object Time1 of class TIME_SENDER. Subsequently the associated variables and methods associated with this object can be accessed.

The TIME_SENDER is a Service Interface Function Block(SIFB) created in MATLAB that sends time values to FBDK. It is the FB initiating the process of communication. Hence TIME_SENDER can be viewed as a “virtual FB” that is created at the MATLAB end to enable co-simulation. At the receiving end that is FBDK the SIFB is essentially the FB accepting the request. In this application it has been named CLT_0_1_TIME. It has been essentially created as a composite FB and looks like the figure below:
The composite FB encapsulates a CLIENT FB that exchanges time values. The basic FB that is present within looks like as shown in Figure 3-10. The data value RD_1 is of data type TIME in this case.

3.7 Time Synchronization between Simulink and FBDK

In order to successfully achieve co-simulation between Simulink and FBDK time synchronization needs to be achieved between the two platforms. Alternatively, the time step in Simulink must be matched with the time step in FBDK. The method implemented in this research
is a modification of the method described in [15]. The MGRID_PWR_CMP block is the Function block that performs the time comparison and consequently the synchronization. The FB is shown in Figure 3-11. The inputs labeled STRT_A, DURN_A, STRT_B, DURN_B, and time are used to perform the time synchronization. This enables the execution of a particular event at a specified instant of time.

![Figure 3-11 Time comparison FB](image)

The input labeled time receives its input from the SERVER FB that is created in MATLAB using the procedure described in Section 3.5.4.

The Execution Control Chart for the time comparison FB is shown in Figure 3-12.
Figure 3-12 ECC for time comparison

The box titled REQ attached to the box titled CNF is the name of the algorithm performing the time comparison and power comparison. Another box titled REQ on the left is a state. When the REQ state gets active the algorithm associated with REQ gets executed and triggers the CNF event. Figure 3-13 shows the transition between the states START and REQ. The transition takes place when event variable REQ goes true.

Figure 3-13 Guard condition

The part of the application that receives data and synchronizes FBDK with MATLAB is shown in Figure 3-14.

Figure 3-14 Part of application performing synchronization
The data from MATLAB is received using CLIENT_0_3_LREAL_TIME. The values received are those of the load demand, load start time and load duration. These are the essential parameters that are required for the implementation of the economic dispatch application. They are then sent to the FB labeled MGRID_PWR_CMP. The time synchronization and load comparison is performed using this FB. This part of the application is common to both Stage 2 and Stage 3.
Chapter 4

Simulation Results, Conclusions and Future Work

As discussed in Chapter 3 the application development is divided into 3 stages. Each stage has a different load balancing behavior run for the same variation in load. The time variation for each load is developed using a random number generator in MATLAB. Once the random numbers are generated they are stored in the MATLAB workspace. These numbers are stored in a data file in MATLAB and loaded during every test stage. Thus this logic ensures a randomly varying load profile and also creates a valid environment for comparing the effectiveness of each behavior. The effectiveness of each application is decided by comparing the power supplied by the DGs in each of the microgrids and Utility. Based on the power generated the total cost of operation for one day is calculated and compared. Section 4.1 discusses the power generated by the DGs and the Utility and Section 4.2 describes the calculation for cost of production of power generated by the DGs and Utility.

4.1 Simulation Results for power outputs

The power supplied by each of the DGs and the utility for each of the stages is shown in Figures 4-1 to Figure 4-3.

Stage 1

Under this stage the Utility and the DGs are supplying the loads based on the generator impedances. This Stage represents the typical operation of a power network without any form control. It forms the basis for the development for the subsequent stages. The power supplied by each of the generating units for Stage 1 is shown in Figure 4-1.
Figure 4-1 Power supplied for 24 hours: Stage 1

Stage 2

Fig 4.2 shows the power supplied by the Utility and the DGs under the single microgrid application in Stage 2.

Figure 4-2 Power supplied for 24 hours: Stage 2 – Single microgrid dispatch
On comparison with the power supplied in Stage 1 there is a reduction in power supplied by the utility and a corresponding increase in the power supplied by DG3. This result is based on the comparison of the availability and Levelized Cost of Energy of DG3 and the Utility. For the model under consideration since cost of DG3 is less than the cost of Electricity of Utility network, DG3 is the primary choice of supplying the excess demand. The reduction in Utility power can be noticed in the time window starting at 0.5 seconds to about 6 seconds. In Stage 1 the power supplied by the utility during this time window is approximately 1.25MW. In Stage 2, for the same time window the power supplied by the utility reduces to approximately 1MW. This reduction in power supplied by the utility needs to be compensated in order for the load to continually receive power. This reduction is compensated by DG3. An increase in generation can be observed for DG3 by the same amount. The powers generated by DG1 and DG2 are almost unchanged indicating that the loads associated with these DGs are under the threshold. The benefit of the application under this stage is that it reduces the cost of generation as compared to the cost of generation in Stage 1 and that it is straightforward to implement.

**Stage 3**

The load balancing application in Stage 3 is more efficient than the application in Stage 2 in terms of cost of operation. This is because the economic dispatch application also considers economics of generating sources in adjacent micro grids. The application developed under this stage considers availability and LCOE as the factors when calculating the power generation of each of the sources. This cost factor assists policy makers in deciding the cost of selling power. Fig 4.3 shows the power generated by the DGs and the Utility network. The difference between the power outputs in this stage and the power outputs in Stage 2 can be noticed for the utility and the other DGs as well. During this time of zero power for the utility the DGs compensate by increasing their generation.
A comparison of the average power supplied by each of the generating units is shown from Figure 4-4 to Figure 4-7.
Figure 4-5 Average power supplied by DG1

Figure 4-6 Average power supplied by DG2
On observing the average power supplied by generating units in each of the Stages it can be concluded that economic dispatch application aims at choosing that generating unit that is capable of supplying the loads at the lowest price. A decrease in power generated by one or more generating units is compensated by the other more economically feasible generating units.

4.2 Daily cost of generation for different Stages

Having discussed the power outputs for each of the Stages this section tabulates the total cost of generation of each of the sources for a single day. The cost of energy generation forms a good basis for the effectiveness of the application developed in each Stage.
Table 4-1 Cost of generation for 24 hours

<table>
<thead>
<tr>
<th></th>
<th>Utility ($/day)</th>
<th>DG1 ($/day)</th>
<th>DG2 ($/day)</th>
<th>DG3 ($/day)</th>
<th>Total Cost ($/day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1</td>
<td>03741</td>
<td>04441</td>
<td>08630</td>
<td>01938</td>
<td>18750</td>
</tr>
<tr>
<td>Stage 2</td>
<td>03071</td>
<td>04340</td>
<td>08471</td>
<td>02133</td>
<td>18015</td>
</tr>
<tr>
<td>Stage 3</td>
<td>02060</td>
<td>04674</td>
<td>08388</td>
<td>02170</td>
<td>17292</td>
</tr>
</tbody>
</table>

Table 4-1 shows that the values in the total cost column reduces with the development of successive stages. The total cost is the sum of the cost of Energy of the individual generating units and the Utility network. This indicates the effectiveness of each load balancing application. Each of applications are developed as a global solution. Hence depending on the location the individual cost of generation (DG1, DG2, DG3 or Utility) will vary. However the overall goal of achieving a reduced total cost is achieved. A reduction in power supplied by the one generating unit is compensated by an increase in the generation of the other DGs or the large utility network. This aspect is also reflected in the cost of energy production for these DGs. For the case scenario considered the Levelized Cost of generation for the DGs calculates to be lower than the cost of electricity from the Utility network. Hence there is a reduction in power supplied by the Utility network. This in turn reflects in the cost of electricity distributed by Utility network for a single day. Also based on the Levelized Cost of Energy of the individual generating units there will be a increase or decrease in the cost of operation based on the economic dispatch algorithm. This is evident from the graphical representation of the costs shown in Figure 4-8 (a)-(d).
Figure 4-8 Cost of generation for 24 hours

(a) Total cost of operation for Utility network

Figure 4-8 (b) Total cost of energy production for DG1
Figure 4-8 Cost of generation for 24 hours

Figure 4-8 (c) Total cost of energy production for DG2

Figure 4-8 (d) Total cost of energy production for DG3
The Total Cost of operation for 24 hours for the entire network is shown in Figure 4-9. A gradual decrease can be observed in the total cost of energy production for a day.

![Total Cost](image)

**Figure 4-9 Total Cost of operation for 24 hours**

The effectiveness of each scheme can be better compared by calculating the percentage saving achieved by load balancing applications in each of the stages when compared with the normal operation in Stage 1. This is shown in Figure 4-10.

![Percent Saving](image)

**Figure 4-10 Percentage Saving**
4.3 Interpretation of Results

On observing the results in each of the stages it can be pointed out that with the development of each stage there is a reduction in the cost of generation. The economic dispatch application uniquely combines the features of load forecasting, economic modeling of renewable energy systems using the concept of Levelized cost of Energy. The benefit of using LCOE for financial calculations is that, the input variables of the LCOE equation are modeled using data available thus more accurately reflecting the cost uncertainty associated with a particular generating technology. The LCOE can be affected by considering income tax rates, tax incentives. Hence the LCOE equation can be extended further to model next to real world market conditions. The economic dispatch application developed here is a global solution. Therefore depending upon the location, capacity and generating technology the results will vary. The load profile for the region under consideration also affects the final output. Hence if the economic dispatch is to be implemented in a different region with a different set of DGs and another set of variable load profiles the results will vary, however the end goal of achieving a cost benefit is achieved. In essence the only inputs to the control application would be the load value, start time, load duration of the load profile for a given region, the capacities of the generators and the LCOE of the generating technologies used. Thus the control application has been designed to be flexible to implement.

The control method using IEC 61499 is also worth discussing. The ease of implementation and the flexibility that is achieved using this control architecture is the primary reason for choosing IEC 61499. An engineer can build control applications in IEC 61499 by just knowing the task or algorithm that a FB is designated to execute without having to have knowledge of the internal process that runs the FB. This is in contrast to the large software packages that are used to develop Smart Grid control applications. Also the IEC 61499 standard
sets guidelines for inter-operation between devices manufactured by different vendors. This increases the flexibility of the control systems that are used.

**4.4 Conclusions**

Economic dispatch can be performed to guide policy makers, regulators and investors in making sound financial decisions for optimal power system operation. It can help different Utility operators decide on policies relating to inter-operation of power system networks. The LCOE index provides a platform for assessing different renewable energy resources by providing an economic ranking system that considers the initial capital cost, O&M cost and the environmental impacts of the renewable sources of generation. Thus in locations where renewable energy technologies are more expensive than conventional generating technologies the federal and state governments can provide subsidies and incentives to help bring down the cost of these technologies.

Policy makers consistently make an effort to create market conditions that are conducive for the inter-operation of utility networks. This can be achieved by developing and implementing smart and intelligent control applications that give the regulators an economic benefit. This project combines both the aspects of intelligent control and an economic benefit by implementing the Function Block Control to develop load balancing applications that prove to be economically beneficial. In conclusion the ease and flexibility the control application is an unique feature that has been implemented in this thesis.
4.5 Future work and developments

As mentioned in Chapter 1 the focus of this project is the control of distributed generation of the grid. Future work can include the control of load demand of the grid or also called demand side management by including load queuing. This extends the area of applications of IEC 61499 into Demand Response. Also the model considered a single DG within each microgrid. A future extension would be to modify the load balancing behavior to accommodate multiple DGs within the same microgrid.
Appendix

A Creating a basic FB

Start the FBEditor.

Create a new basic FB type in the editor by clicking - New... => FB Type =>

Basic menu item. The window that appears is shown in Figure A-1:

![Figure A-0-1 FB Editor](image)

Edit the Function Blocks sub- properties such as name, properties, ECC and Algorithms from the Navigation port.

A basic FB to add two numbers is depicted in Fig A-2 and the process to create a basic FB to add two numbers is as follows:
The properties tab enables to edit the name and it looks as shown in Fig A-3:

This dialog allows the editing of the name and comment fields for the basic FB. A field for editing can be opened by double-clicking on it.

Editing the function block's elements: Edit the FB type's properties by typing the package name as fb.rt.<pckgname> in the field Compiler Info header. Here <pckgname> is the name of the subdirectory in src folder that is being used.

Edit the FB type's interface: An interface element can be edited by double-clicking on its type name, e.g., EVENT or REAL, or via the Edit item of the corresponding pop-up menu. When clicking the interface of the FB the following operations can be performed:

Moving Elements: An interface element such as an event or a variable can be moved up or down in the list of elements by clicking on the Up/Down buttons on the toolbar.
Associating Elements: An event is associated with a variable by performing Alt+Drag action between the event's e.g. REQ, and the variable, e.g., UINT. Using this action on an already existing association deletes the association.

Deleting Elements: An interface element is deleted by clicking the Delete option in the pop-up menu obtained by right-clicking on the specific element's type name. If the element is already in use, for e.g., in an ECC transition, function block diagram in the body of a function block, or a part of an algorithm it cannot be deleted.

Editing the ECC:

Add a new State to ECC by clicking the New Action option in the ECC popup menu. This menu is obtained by right clicking on an empty area of the workspace).

Deleting a State: An ECC state is deleted by using the Delete option in the popup menu or by pressing Alt+Del. However this will also delete all transitions to or from the state.

The EC State Dialog: This dialog box is used to add a new state to an ECC or edit an existing state. It is invoked by clicking the "Add State" option in the popup menu of an ECC, or by double-clicking an existing EC state or clicking the "Edit" item of its popup menu.

![Edit EC State](image)

**Figure A-0-4 EC State Editor**

After pressing the OK button or the Enter key while entering data, if no errors are detected, the changes made will be made and the box will close. If there is any error on pressing the OK button or the Enter key, an error message appears at the top of the dialog, for instance:
A state can be made the initial state of the ECC by selecting the required state in the ECC and pressing the \(\uparrow\) Up button till it becomes the first state in the ECC.

Editing ECC Transitions: Adding a Transition to ECC: Add a transition between any two states of an ECC using drag-and-drop editing in the same manner as connections in a function block diagram (FBD). The initial transition condition is unconditional (1), which may lead to a self-cycle or a guard-only cycle, so an EC Transition Dialog is opened to check for this possibility and correct the transition condition if necessary and it looks like the box shown in Fig A-6:
Deleting a Transition: A transition can be deleted by using the Delete item of its popup menu.

Transition Priorities: At runtime, each ECC contains exactly one active step. Transitions out of an active step are evaluated in the order in which they are declared in the XML document, subject to the following rules:

Upon the occurrence of an input event, only those transitions out of the active step that are associated with the event will be evaluated. Upon completion of the actions associated with the active step, those transitions out of the step that are not associated with an event will be evaluated. To assist in determining the order of transition evaluation, the order in which the transition has been declared is shown in curly brackets \{\} preceding the transition condition. The order in which a selected transition is declared (and hence its priority of evaluation at runtime) can be changed by using the Up/Down buttons on the menu bar.

The final ECC looks like this:

![Execution Control Chart](image-url)
After the REQ event arrives at the input the ECC reaches a state REQ. This is indicated by the box at the bottom left corner. The state REQ has an action associated with it. In this case it is the algorithm titled REQ. This is adjacent to the box titled CNF. The box titled CNF is an output event that gets executed when the algorithm titled REQ is executed.

Editing EC State Actions:

Adding an Action to an EC State: Add an action to an EC state with the New item of the state's popup menu.

Deleting an Action: Delete an action by using the Delete item of its popup menu or by using the Alt+Del keychord.

Moving an Action: If an EC state has more than one action, a selected action up or down with the Up/Down buttons on the main menu bar.

The EC Action Dialog: Use this dialog both for adding a new action to an ECC state or for editing an existing action. Invoke it by clicking the New Action popup menu item of an EC state, or by double-clicking an existing EC action, or by clicking the Edit item of an existing action's popup menu. The dialog contains a drop-down list for selecting the algorithm (if any) to be performed by the action, and another for selecting the output event (if any) to be issued upon completion of the algorithm. It looks like the figure shown below:
Algorithms: Adding an Algorithm to a Basic Function Block Type: Add an algorithm to a basic FB type with the New item of the popup menu of the Algorithms node in the Navigation tree.

Deleting an Algorithm: delete an algorithm by first deleting all references to the algorithm in EC actions, then using the Delete item of the algorithm's node in the Navigation tree.

Editing an algorithm: Add the dialog both for adding a new algorithm to a basic FB type or for editing an existing algorithm. Invoke it by double-clicking an existing algorithm in the Navigation tree or clicking the Edit item of its popup menu, or by selecting the New Algorithm item of the Algorithms popup menu in the Navigation tree. The dialog contains: Text fields for entering a name and comment for the algorithm. A set of radio buttons for selecting the language in which the algorithm is programmed. A text area for editing algorithms in the ST or Java languages (algorithms in the LD or FBD languages are edited in the Worksheet area). Buttons and text fields for performing find/replace operations in the text area.

![Figure A-0-9 Algorithm Dialogue Box](image)
Algorithms can be edited in Structured Text or Java. If an algorithm is written in Java or ST the algorithm is generated in the other language automatically on compiling.

After editing is finished the work can be saved with the Save As... =>XML dialog.

![Save ADD Dialogue Box](image)

Test the FB type by clicking Run button. In the test window, set or clear the values of the inputs, click on event input buttons and observe the resulting event and variable outputs.
B Program for Economic Dispatch inside FB

if(cb.value==1) // if one or both the loads in a micro grid exceed their respective thresholds.
{

// start with checking if loads in neighboring grids are below their thresholds.

if(LA.value<PthA.value && LB.value<PthB.value && LC.value<PthC.value && LD.value<PthD.value && L0_1.value>Pth0_1.value && L0_2.value>Pth0_2.value)
{

if((LA.value+LB.value+L0_1.value - Pth0_1.value+L0_2.value)<=(PthA.value+PthB.value)&&(LC.value+LD.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)>(PthC.value+PthD.value) )
{

// if only one of the DGs is capable of supplying the load entirely take power only from that DG

CB_A_B.value=1;
CB_B_C.value=2;
CB_UTIL.value=2;
}
elseif((LC.value+LD.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)<=(PthC.value+PthD.value)&&(LA.value+LB.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)>(PthA.value+PthB.value))
{

CB_A_B.value=2;
CB_B_C.value=1;
CB_UTIL.value=2;
}
101

```c
elseif((LC.value+LD.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)<=(PthC.value+PthD.value) && (LA.value+LB.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)<=(PthA.value+PthB.value))
{
    // if both DGs are capable of supplying the load then compare the cost of energy production and
    // take power from the cheapest source

    if((((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostX.value)]<=(((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostY.value) &&
((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostY.value)<= (((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostUTIL.value))
    {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=2;
    }

} elseif(((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostY.value)<=(((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostX.value) &&
((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostY.value)<= (((L0_1.value-Pth0_1.value)*T0_1.value+(L0_2.value-Pth0_2.value)*T0_2.value)*CostUTIL.value))
    {
        CB_A_B.value=2;
        CB_B_C.value=1;
    }
```

CB_UTIL.value=2;

}
else
{
  // if Utility is the cheapest source of generation then take in power from the Utility network
  CB_A_B.value=2;
  CB_B_C.value=2;
  CB_UTIL.value=1;
}

elseif((LA.value+LB.value+L0_1.value-Pth0_1.value+L0_2.value-
Pth0_2.value)>(PthA.value+PthB.value)&&(LC.value+LD.value+L0_1.value-
Pth0_1.value+L0_2.value-Pth0_2.value)>(PthC.value+PthD.value))
{
  if((LA.value+LB.value+LC.value+LD.value+L0_1.value-Pth0_1.value+L0_2.value-
Pth0_2.value)<=(PthA.value+PthB.value+PthC.value+PthD.value))
  {
    // if neither of the DGs can supply the loads completely then take in power from a combination
    of the 2 cheapest source of generation
    CB_A_B.value=1;
    CB_B_C.value=1;
    CB_UTIL.value=2;
  }
  else
  {
    CB_A_B.value=1;
    CB_B_C.value=1;
  }
CB_UTIL.value=1; // take power from all the sources of generation in the event that the load can not be supplied by the DGs alone
}
}
}
if(LA.value<PthA.value&&LB.value<PthB.value&&(LC.value>PthC.value||LD.value>PthD.value) && L0_1.value>Pth0_1.value && L0_2.value>Pth0_2.value)
{
if((LA.value+LB.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)<=(PthA.value+PthB.value))
{
CB_A_B.value=1; // if only one of the DGs are capable of supplying the load demand; accept generation from that DG
CB_B_C.value=2;
CB_UTIL.value=2;
}
else
{
CB_A_B.value=1;
CB_B_C.value=2;
CB_UTIL.value=1; // Both Utility and DG are supplying the excess load
}
}
}
if(LC.value<PthC.value&&LD.value<PthD.value&&(LA.value>PthA.value||LB.value>PthB.value) && L0_1.value>Pth0_1.value && L0_2.value>Pth0_2.value)
if((LC.value+LD.value+L0_1.value-Pth0_1.value+L0_2.value-Pth0_2.value)\leq(PthC.value+PthD.value))
{
    CB_A_B.value=2;
    CB_B_C.value=1;
    CB_UTIL.value=2;
}
else
{
    CB_A_B.value=2;
    CB_B_C.value=1;
    CB_UTIL.value=1;
}
}

---------------------

if(LA.value<PthA.value \&\& LB.value<PthB.value \&\& LC.value<PthC.value \&\& LD.value<PthD.value \&\& L0_1.value>Pth0_1.value \&\& L0_2.value<=Pth0_2.value)
{
    // this subsection performs the economic dispatch if only one of the forecasted load exceeds the threshold value.
    if((LA.value+LB.value+L0_1.value-Pth0_1.value)\leq(PthA.value+PthB.value) \&\& (LC.value+LD.value+L0_1.value-Pth0_1.value)>(PthC.value+PthD.value))
    {
    // this subsection performs the economic dispatch if only one of the forecasted load exceeds the threshold value.
{ 
    CB_A_B.value=1;
    CB_B_C.value=2;
    CB_UTIL.value=2;
}

elseif((LC.value+LD.value+L0_1.value-
        Pth0_1.value)<=(PthC.value+PthD.value) 
        &&
        (LA.value+LB.value+L0_1.value-
        Pth0_1.value)>(PthA.value+PthB.value))
    {
        CB_A_B.value=2;
        CB_B_C.value=1;
        CB_UTIL.value=2;
    }

else if ((LC.value+LD.value+L0_1.value-
        Pth0_1.value)<=(PthC.value+PthD.value) 
        &&
        (LA.value+LB.value+L0_1.value-
        Pth0_1.value)<=(PthA.value+PthB.value))
    {
        if(((L0_1.value-Pth0_1.value)*T0_1.value)*CostX.value)<=(((L0_1.value-
        Pth0_1.value)*T0_1.value)*CostY.value)
        &&
        (((L0_1.value-
        Pth0_1.value)*T0_1.value)*CostX.value)<=(((L0_1.value-
        Pth0_1.value)*T0_1.value)*CostUTIL.value))
            {
                CB_A_B.value=1;
                CB_B_C.value=2;
                CB_UTIL.value=2;
            }
    }

elseif((L_0_1.value-Pth_0_1.value)*T_0_1.value)*CostY.value<=(((L_0_1.value-Pth_0_1.value)*T_0_1.value)*CostX.value) && (((L_0_1.value-Pth_0_1.value)*T_0_1.value)*CostY.value)<=(((L_0_1.value-Pth_0_1.value)*T_0_1.value)*CostUTIL.value))
{
    CB_A_B.value=2;
    CB_B_C.value=1;
    CB_UTIL.value=2;
}
else
{
    CB_A_B.value=2;
    CB_B_C.value=2;
    CB_UTIL.value=1;
}
elseif((LA.value+LB.value+L_0_1.value-Pth_0_1.value)>(PthA.value+PthB.value) && (LC.value+LD.value+L_0_1.value-Pth_0_1.value)>(PthC.value+PthD.value))
{
    if((LA.value+LB.value+LC.value+LD.value+L_0_1.value-Pth_0_1.value)<=((PthA.value+PthB.value+PthC.value+PthD.value))
    {
        CB_A_B.value=1;
        CB_B_C.value=1;
        CB_UTIL.value=2;
    }
}
} else {
    CB_A_B.value=1;
    CB_B_C.value=1;
    CB_UTIL.value=1;
}

if(LA.value<PthA.value && LB.value<PthB.value && (LC.value>PthC.value || LD.value>PthD.value) && L0_1.value>Pth0_1.value && L0_2.value>=Pth0_2.value) {
    if((LA.value+LB.value+L0_1.value-Pth0_1.value)<=(PthA.value+PthB.value)) {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=2;
    } else {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=1;
    }
} else {
    CB_A_B.value=1;
    CB_B_C.value=2;
    CB_UTIL.value=1;
}
if(LC.value<PthC.value && LD.value<PthD.value && (LA.value>PthA.value || LB.value>PthB.value) && L0_1.value>Pth0_1.value && L0_2.value<=Pth0_2.value)
{
if((LC.value+LD.value+L0_1.value-Pth0_1.value)<=((PthC.value+PthD.value))
{
CB_A_B.value=2;
CB_B_C.value=1;
CB_UTIL.value=2;
}
else
{
CB_A_B.value=2;
CB_B_C.value=1;
CB_UTIL.value=1;
}
}

if(LA.value<PthA.value && LB.value<PthB.value && LC.value<PthC.value && LD.value<PthD.value && L0_1.value<Pth0_1.value && L0_2.value>Pth0_2.value)
{
if((LA.value+LB.value+L0_2.value-Pth0_2.value)<=(PthA.value+PthB.value) && (LC.value+LD.value+L0_2.value-Pth0_2.value)>(PthC.value+PthD.value))
{

}
CB_A_B.value=1;
CB_B_C.value=2;
CB_UTIL.value=2;
}
}
else if((LC.value+LD.value+L0_2.value-Pth0_2.value)<=(PthC.value+PthD.value) &&
(LA.value+LB.value+L0_2.value-Pth0_2.value)>(PthA.value+PthB.value))
{
CB_A_B.value=2;
CB_B_C.value=1;
CB_UTIL.value=2;
}
else if ((LC.value+LD.value+L0_2.value-Pth0_2.value)<=(PthC.value+PthD.value) &&
(LA.value+LB.value+L0_2.value-Pth0_2.value)>(PthA.value+PthB.value))
{
if((((L0_2.value-Pth0_2.value)*T0_2.value)*CostX.value)<=(((L0_2.value-Pth0_2.value)*T0_2.value)*CostY.value) &&
((L0_2.value-Pth0_2.value)*T0_2.value)*CostX.value)<=(((L0_2.value-Pth0_2.value)*T0_2.value)*CostUTIL.value))
{
CB_A_B.value=1;
CB_B_C.value=2;
CB_UTIL.value=2;
}
elseif(((L0_2.value-Pth0_2.value)*T0_2.value)*CostY.value)<=(((L0_2.value-Pth0_2.value)*T0_2.value)*CostX.value) &&
}
Pth0_2.value)*T0_2.value)*CostY.value)\leq(((L0_2.value-
Pth0_2.value)*T0_2.value)*CostUTIL.value))

{  
  CB_A_B.value=2;
  CB_B_C.value=1;
  CB_UTIL.value=2;
}
else
{
  CB_A_B.value=2;
  CB_B_C.value=2;
  CB_UTIL.value=1;
}

elseif((LA.value+LB.value+L0_2.value-Pth0_2.value)\geq(PthA.value+PthB.value)\&\&
(LC.value+LD.value+L0_2.value-Pth0_2.value)\geq(PthC.value+PthD.value))
{
  if((LA.value+LB.value+LC.value+LD.value+L0_2.value-
Pth0_2.value)\leq(PthA.value+PthB.value+PthC.value+PthD.value))
  {
    CB_A_B.value=1;
    CB_B_C.value=1;
    CB_UTIL.value=2;
  }
}
else
{
    CB_A_B.value=1;
    CB_B_C.value=1;
    CB_UTIL.value=1;
}

if(LA.value<PthA.value&&LB.value<PthB.value&&(LC.value>PthC.value||LD.value>PthD.value) && L0_1.value<=Pth0_1.value && L0_2.value>Pth0_2.value)
{
    if((LA.value+LB.value+L0_2.value-Pth0_2.value)\=(PthA.value+PthB.value))
    {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=2;
    }
else
    {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=1;
    }
}

if(LA.value<PthA.value&&LB.value<PthB.value&&LC.value>PthC.value&&LD.value>PthD.value) && L0_1.value<=Pth0_1.value && L0_2.value>Pth0_2.value)
{
    if((LA.value+LB.value+L0_2.value-Pth0_2.value)\=(PthA.value+PthB.value))
    {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=2;
    }
else
    {
        CB_A_B.value=1;
        CB_B_C.value=2;
        CB_UTIL.value=1;
    }
}
if (LC.value < PthC.value && LD.value < PthD.value && (LA.value > PthA.value || LB.value > PthB.value) && L0_1.value <= Pth0_1.value && L0_2.value > Pth0_2.value) {
    if ((LC.value + LD.value + L0_2.value - Pth0_2.value) <= (PthC.value + PthD.value)) {
        CB_A_B.value = 2;
        CB_B_C.value = 1;
        CB_UTIL.value = 2;
    } else {
        CB_A_B.value = 2;
        CB_B_C.value = 1;
        CB_UTIL.value = 1;
    }
} else {
    CB_A_B.value = 2;
    CB_B_C.value = 2;
}

else if (cb.value == 0) {
    // if none of the loads exceed their respective threshold values then they are supplied by the associated DGs in their grid itself
    CB_A_B.value = 2;
    CB_B_C.value = 2;
}
CB_UTIL.value=2;
}

References


[17] V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-


[28] "4DIAC for Smart Grid Applications," in 16th IEEE International Conference on Emerging


[38] "TransparentCostDatabase,"[Online].Available:


