**0Revision Sheet**

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| Rev. 0.0 | 11/02/2017 | Application guideline template |
| Rev. 0.1 | 11/29/2017 | First version |
| Rev. 0.2 | 12/13/2017 | Added data logging and protection coordination |
| Rev. 0.3 | 12/14/2017 | Added communication |
| Rev. 0.4 | 12/15/2017 | Integrate all parts together, first complete version |
| Rev. 0.5 | 01/22/2018 | Updates figures |
| Rev. 0.6 | 10/17/2018 | Updates new function blocks and Instruction for utilizing the microgrid controllers with a new feeder |
| Rev. 0.7 | 04/01/2019 | Modification on the main body |
| Rev. 0.8 | 06/28/2019 | Add the Planned/unplanned islanding, topology identification, fault and abnormal state control;  Revise the new MG implementation |
| Rev. 0.9 | 06/28/2019 | Modified for DynaMiC\_Basic on GitHub |
| Rev. 1.0 | 08/06/2020 | Modified for DynaMiC\_NET\_MG\_Basic  on GitHub |
| Rev. 1.1 | 10/1/2020 | Finalize the application for DynaMiC\_Basic and DynaMiC\_NET\_MG\_Basic |

COPYRIGHT NOTICE:

Copyright <2020> <The University of Tennessee at Knoxville and the University of Tennessee Research Foundation>

REDISTRIBUTIONS:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the redistribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

DISCLAIMER:

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**TABLE OF CONTENTS**

Page #

[1. INTRODUCTION 2](#_Toc52529281)

[1.1 Purpose 2](#_Toc52529282)

[1.2 Organization of the guideline 2](#_Toc52529283)

[1.3 References 3](#_Toc52529284)

[1.4 Authorized Use Permission 3](#_Toc52529285)

[1.5 Acronyms and Abbreviations 4](#_Toc52529286)

[2. System function block summary 0](#_Toc52529287)

[2.1 System Configuration 0](#_Toc52529288)

[2.2 User Access Levels 2](#_Toc52529289)

[3. GETTING STARTED 4](#_Toc52529290)

[3.1 Download, Installation and Requirement 4](#_Toc52529291)

[4. MICROGRID CENTRAL CONTROLLER 6](#_Toc52529292)

[4.1 Finite State Machine 7](#_Toc52529293)

[4.1.1 Functionality description 7](#_Toc52529294)

[4.1.2 Function Block diagram 7](#_Toc52529295)

[4.1.3 Algorithm flow chart 10](#_Toc52529296)

[4.1.4 Function inputs and outputs 10](#_Toc52529297)

[4.2 Model Management 11](#_Toc52529298)

[4.2.1 Functionality description 11](#_Toc52529299)

[4.2.2 Function Block diagram 11](#_Toc52529300)

[4.2.3 Function inputs and outputs 11](#_Toc52529301)

[4.3 State Estimation 12](#_Toc52529302)

[4.3.1 Functionality description 12](#_Toc52529303)

[4.3.2 Function Block diagram 12](#_Toc52529304)

[4.3.3 Algorithm flow chart 12](#_Toc52529305)

[4.3.4 Function inputs and outputs 13](#_Toc52529306)

[4.4 Data logging and event recording 13](#_Toc52529307)

[4.4.1 Functionality description 13](#_Toc52529308)

[4.4.2 Function Block diagram 14](#_Toc52529309)

[4.4.3 Function inputs and outputs 14](#_Toc52529310)

[4.5 PV Forecast 14](#_Toc52529311)

[4.5.1 Functionality description 14](#_Toc52529312)

[4.5.2 Function block diagram 14](#_Toc52529313)

[4.5.3 Algorithm flow chart 15](#_Toc52529314)

[4.5.4 Function inputs and outputs 15](#_Toc52529315)

[4.6 Load Forecast 15](#_Toc52529316)

[4.6.1 Functionality description 15](#_Toc52529317)

[4.6.2 Function block diagram 16](#_Toc52529318)

[4.6.3 Algorithm flow chart 16](#_Toc52529319)

[4.6.4 Function inputs and outputs 16](#_Toc52529320)

[4.7 Energy Management 17](#_Toc52529321)

[4.7.1 Functionality description 17](#_Toc52529322)

[4.7.2 Function block diagram 17](#_Toc52529323)

[4.7.3 Algorithm flow chart 17](#_Toc52529324)

[4.7.4 Function inputs and outputs 18](#_Toc52529325)

[4.8 Communication 18](#_Toc52529326)

[4.8.1 Functionality description 18](#_Toc52529327)

[4.8.2 Function block diagram 18](#_Toc52529328)

[4.8.3 Algorithm flow chart 19](#_Toc52529329)

[4.8.4 Function inputs and outputs 20](#_Toc52529330)

[4.9 Black Start 21](#_Toc52529331)

[4.9.1 Functionality description 21](#_Toc52529332)

[4.9.2 Function Block diagram 21](#_Toc52529333)

[4.9.3 Algorithm flow chart 22](#_Toc52529334)

[4.9.4 Function inputs and outputs 22](#_Toc52529335)

[4.10 Planned/Unplanned Islanding 23](#_Toc52529336)

[4.10.1 Functionality description 23](#_Toc52529337)

[4.10.2 Function block diagram 23](#_Toc52529338)

[4.10.3 Algorithm flow chart 23](#_Toc52529339)

[4.10.4 Function inputs and outputs 24](#_Toc52529340)

[4.11 Fault and abnormal state control 25](#_Toc52529341)

[4.12 Resynchronization 26](#_Toc52529342)

[4.13 Topology Identification 27](#_Toc52529343)

[4.14 Islanded state control 28](#_Toc52529344)

[5. MICROGRID LOCAL CONTROLLER 31](#_Toc52529345)

[5.1 Active power control 31](#_Toc52529346)

[5.1.1 Functionality description 31](#_Toc52529347)

[5.1.2 Function Block diagram 31](#_Toc52529348)

[5.1.3 Algorithm flow chart 31](#_Toc52529349)

[5.1.4 Function inputs and outputs 32](#_Toc52529350)

[5.2 Reactive power control 33](#_Toc52529351)

[5.2.1 Functionality description 33](#_Toc52529352)

[5.2.2 Function Block diagram 33](#_Toc52529353)

[5.2.3 Algorithm flow chart 33](#_Toc52529354)

[5.2.4 Function inputs and outputs 34](#_Toc52529355)

[5.3 Resynchronization 34](#_Toc52529356)

[5.3.1 Functionality description 34](#_Toc52529357)

[5.3.2 Function Block diagram 34](#_Toc52529358)

[5.3.3 Algorithm flow chart 34](#_Toc52529359)

[5.3.4 Function inputs and outputs 35](#_Toc52529360)

[6. INSTRUCTION FOR UTILIZING THE MICROGRID CONTROLLER WITH A NEW FEEDER 38](#_Toc52529361)

[6.1 Microgrid topology acquiring 38](#_Toc52529362)

[6.2 Create the inputs for the model management 44](#_Toc52529363)

[6.3 Create the communication channels 45](#_Toc52529364)

[6.4 Set up the Opal-RT model for controller test 48](#_Toc52529365)

[6.5 Initialize the microgrid controllers 50](#_Toc52529366)

[6.6 Microgrid controller operations 50](#_Toc52529367)

[7. Report and feedback 53](#_Toc52529368)

[7.1 Report an error 53](#_Toc52529369)

[7.2 Feedback 53](#_Toc52529370)

**1.0 INTRODUCTION**

# INTRODUCTION

This chapter presents an overview and brief description of the layout of this Application Guideline document and a list of references and abbreviations used herein.

## 1.1 Purpose

The purpose of this Application Guideline document is to define the functionality and operations guide for the software that has been developed by the University of Tennessee (UTK) for the ARPA-E project entitled “A Smart and Flexible Microgrid with a Low-cost Scalable Open-source Controller” (Award number: DE-AR0000665). This document serves as the sole reference for the scope of the system functionality and operation guide. Changes will be updated in future open source releases.

The controller developed under this project us refered to as the Dynamic Microgrid Controller (DynaMiC). There are four versions of the software for the DynaMiC as listed in Table 1‑1 .

Table 1‑1 Controller versions for DynaMiC

|  |  |  |
| --- | --- | --- |
| **Version Number** | **Version Name** | **Descriptions** |
| 1 | DynaMiC\_Basic | Single microgrid and no dynamic boundary |
| 2 | DynaMiC\_Advanced | Single microgrid with dynamic boundary |
| 3 | DynaMiC\_NET\_MG\_Basic | Multiple microgrids and no dynamic boundary |
| 4 | DynaMiC\_NET\_MG\_Advanced | Multiple microgrids with dynamic boundary |

The DynaMiC\_Basic and DynaMiC\_NET\_MG\_Basic versions are released as open source projects.

The DynaMiC\_Advanced and DynaMiC\_NET\_MG\_Advanced are reserved as proprietary by UTK and licensed under separate commercial agreement.

This Application Guideline is written for the DynaMiC\_NET\_MG\_Basic version.

## 1.2 Organization of the guideline

The application guideline consists of seven chapters: Introduction, System function block summary, Getting started with the controller, Microgrid Central controller, Microgrid local controller, Instruction for utilizing the microgrid controller, and Report and feedback.

The Introduction chapter explains in general terms the controller and the purpose for this guideline.

The ‘System function block summary’ chapter provides a general overview of the MGCC controller. The summary outlines the uses of the controller software requirements, controller configuration, and brief function introduction of each function block. Note that certain of the functions pertaining to dynamic boundary control are proprietary and not released in the open source project.

The ‘Getting started’ chapter provides download and installation instructions for the controller software.

The ‘Microgrid central controller’ and ‘Microgrid local controller’ chapters provide detailed description of each function block - including the block diagram and algorithm flow chart.

The ‘Instruction for utilizing the microgrid controller’ section explains the procedure to apply this controller to a new feeder and provides an example in Opal-RT simulation environment.

The report and feedback chapter describes how to join the open source software development, where to download the latest version, and how to feedback any information to us.

## 1.3 References

References that were used in preparation of this document.

1. <http://curent.utk.edu/>
2. <https://bitbucket.org/microgrid/profile/repositories>
3. <https://github.com/GeniusMicrogrid/DynaMic_Basic>
4. Schneider, K. P., Chen, Y., Chassin, D. P., Pratt, R. G., Engel, D. W., & Thompson, S. E. (2008). Modern grid initiative distribution taxonomy final report (No. PNNL-18035). Pacific Northwest National Lab. (PNNL), Richland, WA (United States).

## 1.4 Authorized Use Permission

The DynaMiC\_Basic and DynaMiC\_NET\_MG\_Basic projects are offered as open source. Thus, developers are welcomed to join and download the software source code through bitbucket. They can be found in the GitHub link in section 1.3.

Certain advanced developed functions related to dynamic boundary management are included in this document for completeness but are maintained as proprietary software and are available for licensing by contacting the sponsor at:

Leskovjan, Andreana aleskovj@utk.edu

Krishnamurthy, Maha mkrishn1@utk.edu

## 1.5 Acronyms and Abbreviations

This section provides a list of the acronyms and abbreviations used in this document and the meaning of each.

|  |  |
| --- | --- |
| NI | National Instrument |
| MG | Microgrid |
| MGCC | Microgrid Central Controller |
| FSM | Finite State machine |
| PV | Photovoltaic |
| ARPA-E | Advanced Research Projects Agency–Energy |
| UTK | The University of Tennessee |
| EPB | Electric Power Board of Chattanooga |
| DAQ | Data acquisition |
| DNP | Distributed Network Protocol |
| IntelliRupter | A smart switch |
| PQ | Power Quality - Active and reactive power |
| ISD | Islanded |
| FTP | File Transfer Protocol |
| Dtms | A file type generated by NI real-time application which can be read by excel |
| SCADA | Supervisory control and data acquisition |
| BESS | Battery energy storage system |
| PCC | Point of common coupling |
| Vi | Virtual instrument |

**2.0 SYSTEM FUNCTION BLOCK SUMMARY**

# System function block summary

This chapter provides a general overview of the controller software. The summary outlines the uses of the system’s hardware and software requirements and system configuration.

## 2.1 System Configuration

The overall software configuration is shown in the Figure 1. There are seventeen different function blocks in MGCC. Each function block operates and can be tested individually. Due to the benefits of the LabVIEW programming, these function blocks operate in parallel. Therefore, the enable and disenable actions are simplified. Following is a summary for each function block.

1. ***Finite State Machine***: processes operational and feedback information from each function block and manages the enable and disable function signals to control the function blocks.
2. ***Model management***: manages the feeder information, including topology, configuration, Y-bus, PQ information, and other presetting information received via FTP from the NI CompactRIO. For detailed data format, please refer to the detailed block description.
3. ***Protection coordinator***: protects the feeder from attack or event through controlling the protective relays.
4. ***Black Start***: controls the restoration of the MG from a total or partial shutdown. The detailed procedure is shown in the block description.
5. ***PV forecasting***: forecasts available PV generation based on historical generation and irradiance data and near-term weather forecast data.
6. ***Data logging and event recorder***: records operational data into the FTP via .tdms data format. The detailed recording list is shown in the block descriptions. The event recorder processes relevant data associated with specified operational events.
7. ***Load forecasting***: forecasts the electrical demand of the system based on historical load profile and current load conditions.
8. ***Energy management***: generates scheduling/dispatch strategies to reduce the electric bill in grid connected mode, and to provide energy surety to the critical load under islanded mode.
9. ***State estimation***: estimates the feeder state based on feedback signals from local controllers.
10. ***Communication***: transfers data between the MGCC and LCs via the DNP3 communication protocol, and interacts with the Supervisory Control and Data Acquistion (SCADA) system(s).
11. ***PQ balance***: provides a strategy of balancing the electrical power and load within the microgrid during islanding operation mode as well as the transition between islanding and grid-connected modes.
12. ***Topology identification***: identifies the current microgrid topology based on feedback from IntelliRupters, BESS, and PV. It provides the topology input to the PQ balance, planned islanding, and FSM.
13. ***Planned islanding***: provides a strategy of the suitable load section combination and BESS power so that the active and reactive power flow through the PCC point can be minimized.
14. ***Unplanned islanding***: provides an ISD state check function and signal to jump from grid connected state to unplanned ISD and then to the ISD state.
15. ***Fault and abnormal state detection***: detects and classifies the fault and abnormal signals from each function block. Based on the fault and abnormal signals, provide flags to the FSM.
16. ***Islanded state control***: opens the uncontrolled IntelliRupters caused by permanent faultin the MG.
17. ***Resynchronization***: reconnects the islanded MG back to the main grid.

As it can be observed from the **Figure 2‑1**, the function blocks have single/bi-directional data transfer between FSM and themselves. The communication among MGCC and LC depends on DNP3 protocol. In DynaMiC\_NET\_MG\_Basic version, the enabled functions in the LabVIEW project are: Black start, communication, data logging, finite state machine, model management, resynchronization, state estimation, planned islanding, load forecasting, PV forecasting, and energy management.

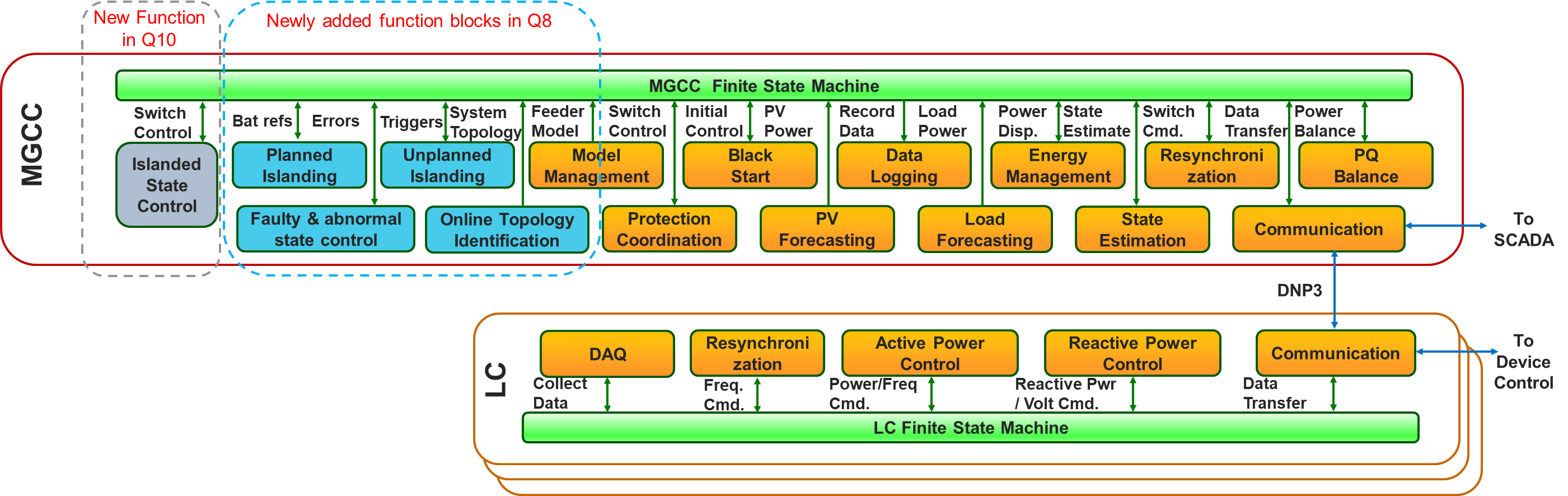


Figure 2‑1 MGCC function block overview

## 2.2 User Access Levels

This controller is open source software. Detailed setting and program information is available to users.

As shown in Table 2‑1 , the open source, proprietary, partial proprietary, and unavailable functions in DynaMiC\_Basic and DynaMiC\_NET\_MG\_Basic versions are given. In this guideline, DynaMiC\_NET\_MG\_Basic is the version discussed.

Table 2‑1 The open source, proprietary, partial proprietary, and unavailable functions in different versions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version name** | **Open source functions** | **Proprietary functions** | **Partial proprietary functions** | **Unavailable functions** |
| DynaMiC\_Basic | * Black start * Communication * Data logging * Finite state machine * Model management * State estimation * Load forecasting * PV forecasting * Islanded state control * Fault and abnormal state detection * Unplanned islanding * Protection coordinator | * PQ balance * Topology identification | * Resynchronization * Planned islanding * Energy management. |  |
| DynaMiC\_NET\_MG\_Basic | * Black start * Communication * Data logging * Finite state machine * Model management * State estimation * Load forecasting * PV forecasting | * PQ balance * Topology identification | * Resynchronization * Planned islanding * Energy management. | * Islanded state control * Fault and abnormal state detection * Unplanned islanding * Protection coordinator |

**3.0 GETTING STARTED**

# GETTING STARTED

This chapter explains how to retrieve the controller software and install it on CompactRIO modules. The detailed manual and test procedures are also involved, including instructions, how to use this function, how to test this function, and examples.

## 3.1 Download, Installation and Requirement

MG controller software can be downloaded from GitHub (provided in section 1.3). If you have successfully joined our development team, you may pull from the master branch and get the entire package.

This MGCC is programmed utilizing LabVIEW 2016. There are several LabVIEW packages required to be installed before running the MGCC. The requirements are listed as follows:

1. LabVIEW 2016
2. LabVIEW real-time 2016
3. LabVIEW DNP3
4. LabVIEW mathscript
5. LabVIEW FTP
6. C-RIO (CompactRIO) module
7. LabVIEW advance signal processing toolkit
8. LabVIEW watchdog timer

At the same time, the software in the CompactRIO should also be installed according to the above requirements. Note that they can be installed through NI Measurement & Automation Explorer (MAX).

In order to connect the CompactRIO through computers, ensure that they are connected to the same local network.

1. **MICROGRID CENTRAL CONTROLLER**

# MICROGRID CENTRAL CONTROLLER

The entire project architecture can be found in the MGCC\_v2 project explorer, shown in Figure 4‑1. In this chapter, the functionality of each sub-function is described. The related Vis can be found in the named virtual folders. The function blocks in MG controller are listed as follows (DynaMiC\_NET\_MG\_Basic version),

1. Black start
2. Communication
3. Data logging
4. Finite state machine
5. Model management
6. Resynchronization
7. State estimation
8. Planned islanding
9. Load forecasting
10. PV forecasting
11. Energy management.

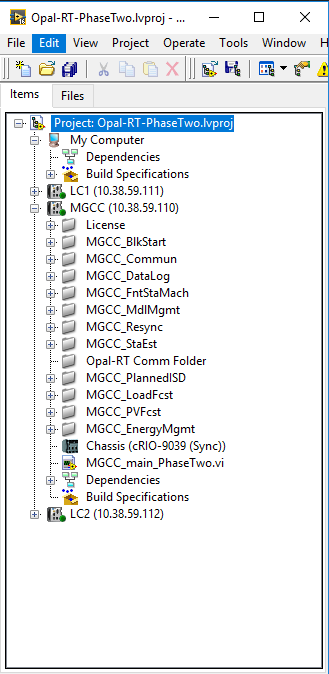


Figure 4‑1 Example Project Explorer

## 4.1 Finite State Machine

### 4.1.1 Functionality description

FSM is the core function block for the MGCC. It classifies and describes which state the MGCC stays in and which could be the next jumping state. It also gathers the feedback and inputs from the LCs and other functions.

### 4.1.2 Function Block diagram

As shown in Figure 4‑2, the block diagram of the finite state machine shows the jumping conditions and the possible destinations of states. The detailed description of each state is shown in Table 4‑1. Basically, there are ten states associated with the MGCC FSM. Other function blocks will generate flags or events which could determine the next/destination of the current state once every control instant. Meanwhile, the destinations of states are listed in Table 4‑2

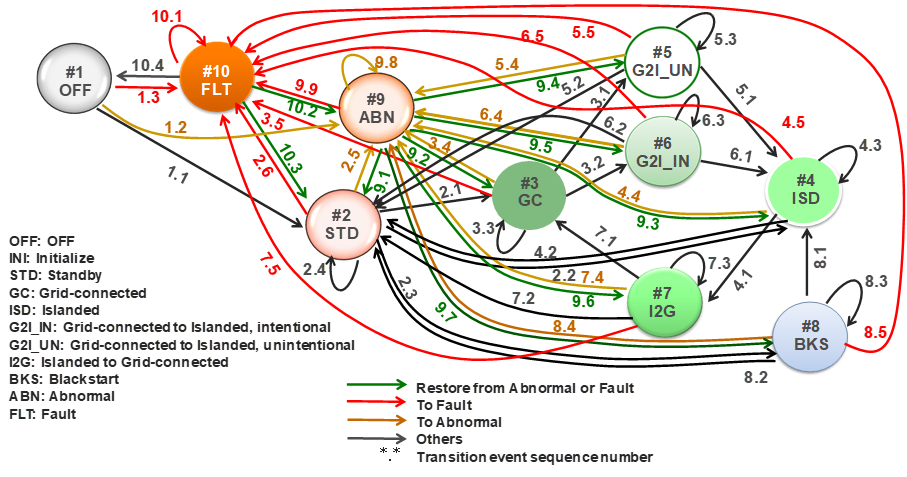


Figure 4‑2 State machine for MGCC

Table 4‑1 MGCC state definition

|  |  |  |  |
| --- | --- | --- | --- |
|  | State Name | Abbr. | Note |
| 1 | OFF | OFF | Power OFF |
| 2 | Standby | STD | Ready to run, update measurements |
| 3 | Grid-connected | GC | Microgrid connected to grid |
| 4 | Islanded | ISD | Microgrid Islanded |
| 5 | Grid-connected to Islanded, intentional | G2I\_IN | Planned Islanding |
| 6 | Grid-connected to Islanded, unintentional | G2I\_UN | Unplanned Islanding |
| 7 | Islanded to Grid-connected | I2G | Resynchronization |
| 8 | Blackstart | BKS | Start MG from Islanding |
| 9 | Abnormal | ABN | Some function blocks cannot run |
| 10 | Fault | FLT | Most of function blocks can run |

Table 4‑2 MGCC state transition events

|  |  |  |  |
| --- | --- | --- | --- |
| Current State | Next State | Transition  Event # | Transition Event Description |
| OFF | Standby | 1.1 | Power up, initialization, etc. |
| Abnormal | 1.2 | Abnormal detected |
| Fault | 1.3 | Any fault detected |
| Standby | Grid-  connected | 2.1 | Grid-connected state detected |
| Islanded | 2.2 | Islanded state detected |
| Blackstart | 2.3 | Blackstart command received |
| Standby | 2.4 | No fault detected, no other state detected |
| Abnormal | 2.5 | Abnormal detected |
| Fault | 2.6 | Any fault detected |
| Grid-connected | G2I\_UN | 3.1 | Unintentional Grid-connected to Islanded state detected |
| G2I\_IN | 3.2 | Intentional Grid-connected to Islanded state detected |
| Grid-  connected | 3.3 | Still in Grid-connected State |
| Abnormal | 3.4 | Abnormal detected |
| Fault | 3.5 | Any fault detected |
| Islanded | I2G | 4.1 | Islanded to Grid-connected state detected |
| Islanded | 4.2 | Still in Islanded State |
| Abnormal | 4.3 | Abnormal detected |
| Fault | 4.4 | Any fault detected |
| Grid-connected  to Islanded,  unintentional (G2I\_UN) | Islanded | 5.1 | Transition to Islanded state succeeded |
| Standby | 5.2 | Transition to Islanded state failed, microgrid outage |
| G2I\_UN | 5.3 | Still in this transition state |
| Abnormal | 5.4 | Abnormal detected |
| Fault | 5.5 | Any fault detected |
| Grid-connected  to Islanded, intentional  (G2I\_IN) | Islanded | 6.1 | Transition to Islanded state succeeded |
| Standby | 6.2 | Transition to Islanded state failed, microgrid outage |
| G2I\_IN | 6.3 | Still in this transition state |
| Abnormal | 6.4 | Abnormal detected |
| Fault | 6.5 | Any fault detected |
| Islanded to  Grid-connected (I2G) | Grid-  connected | 7.1 | Transition to Grid-connected state succeeded |
| Standby | 7.2 | Transition to Grid-connected state failed, microgrid outage |
| I2G | 7.3 | Still in this transition state |
| Abnormal | 7.4 | Abnormal detected |
| Fault | 7.5 | Any fault detected |
| Blackstart | Islanded | 8.1 | Black start succeeded |
| Standby | 8.2 | Black start failed |
| Blackstart | 8.3 | Still in this transition state |
| Abnormal | 8.4 | Abnormal detected |
| Fault | 8.5 | Any fault detected |
| Abnormal | Standby | 9.1 | Abnormal cleared, go back to previous state |
| Grid-  connected | 9.2 | Abnormal cleared, go back to previous state |
| Islanded | 9.3 | Abnormal cleared, go back to previous state |
| G2I\_UN | 9.4 | Abnormal cleared, go back to previous state |
| G2I\_IN | 9.5 | Abnormal cleared, go back to previous state |
| I2G | 9.6 | Abnormal cleared, go back to previous state |
| Blackstart | 9.7 | Abnormal cleared, go back to previous state |
| Abnormal | 9.8 | Abnormal not cleared |
| Fault | 9.9 | Any fault detected |
| Fault | Fault | 10.1 | Any fault not cleared |
| Abnormal | 10.2 | Fatal fault cleared, abnormal detected |
| Standby | 10.3 | All faults cleared, no abnormal condition |
| OFF | 10.4 | Manually control |

### 4.1.3 Algorithm flow chart

The flow chart of finite state machine is shown in Figure 4‑3. It can be observed that the jumping condition can be manually determined or determined by Flags. All these flags come from sub-functions and LC. In each state, there are different enabled functions.

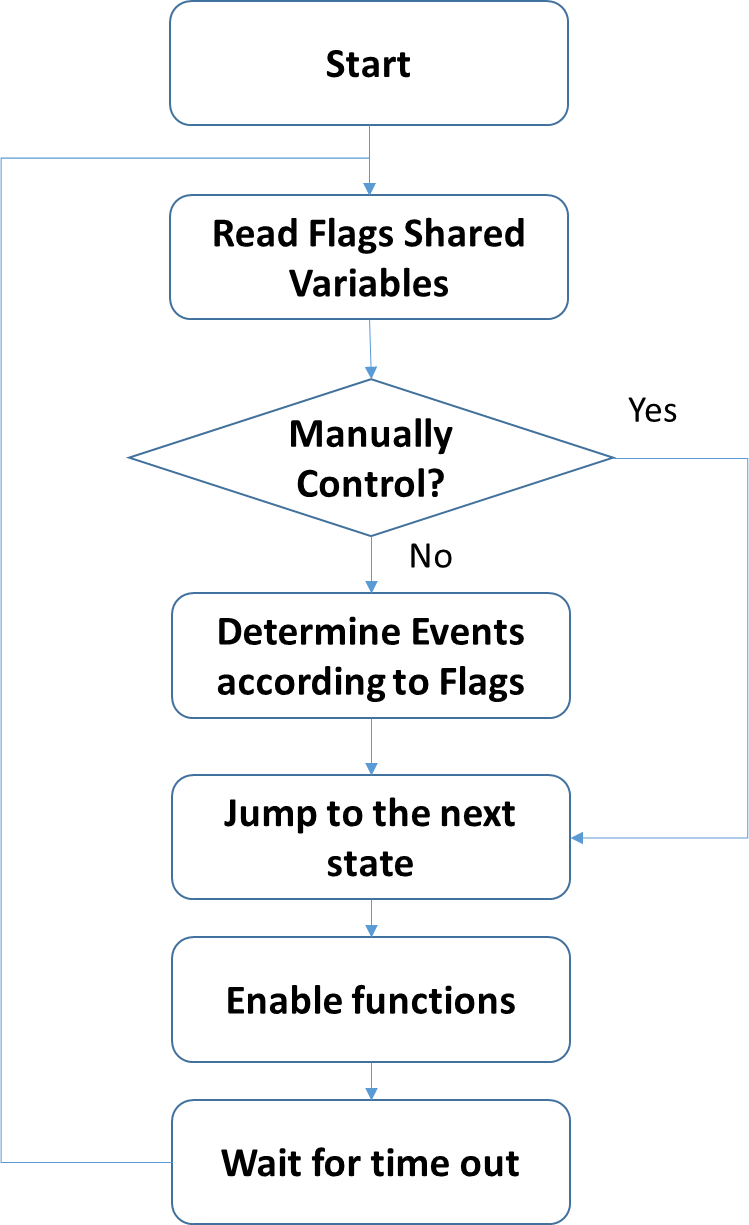


Figure 4‑3 The flow chart for the finite state machine

### 4.1.4 Function inputs and outputs

The finite state machine inputs and outputs are listed in Table 4‑3.

Table 4‑3 FSM inputs and outputs

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Flags from LC and sub-functions | Enable function signals |
| Manually control signals | Next state signals |
| Current state |  |
| Feedback from functions and LC |  |

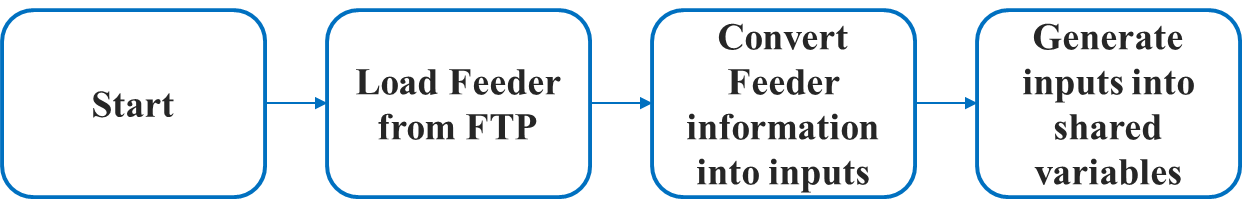
## 4.2 Model Management

### 4.2.1 Functionality description

Model management is the initialization function for the MGCC. The basic function is to load the feeder information and generate the inputs for other sub-functions.

### 4.2.2 Function Block diagram

The block diagram for model management is shown in Figure 4‑4. Basically, all the feeder information come from the FTP and will be converted into shared variables utilized by other sub-functions. Note that the shared variables can be replaced with global variables.



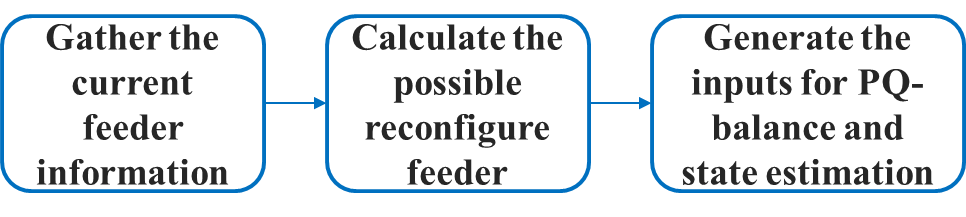
**

Figure 4‑4 Model management flow chart

### 4.2.3 Function inputs and outputs

Table 4‑4 Model management inputs and outputs

|  |  |  |
| --- | --- | --- |
|  | **Inputs** | **Outputs** |
| Static one | Initial feeder information (From CSV) | Feeder line impedance |
| Initial feeder load demand |
| Feeder connection information |
| Dynamic one | Current feeder information (From LCs) | Feeder inputs for PQ balance |
| Feeder inputs for state estimation |

## 4.3 State Estimation

### 4.3.1 Functionality description

State estimation is to estimate the current power flow with limited feeder information. Here, for a reconfigurable feeder, the state estimation function block could estimate the three-phase unbalanced Feeder which can be either island or grid connected.

### 4.3.2 Function Block diagram

The block diagram for state estimation is shown in Figure 4‑5. The state estimation function block requires feeder information from LCs and main grid. In the initialization state, it gathers information from model management function and set up the basic feeder information. Then the feeder information will be renewed once the LC and main grid have any updates. The feeder information will be converted into inputs of the MATLAB based state estimation functions and the results such as error rate and voltage difference will be analyzed after estimation.

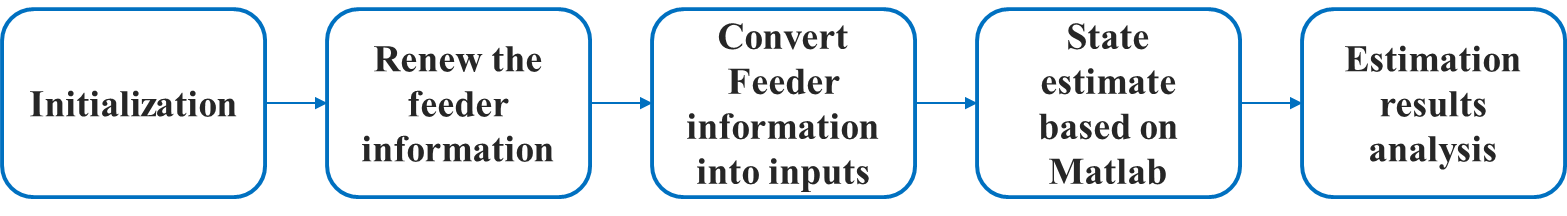


Figure 4‑5 State estimation block diagram

### 4.3.3 Algorithm flow chart

The flow chart of the state estimation is shown in Figure 4‑6.

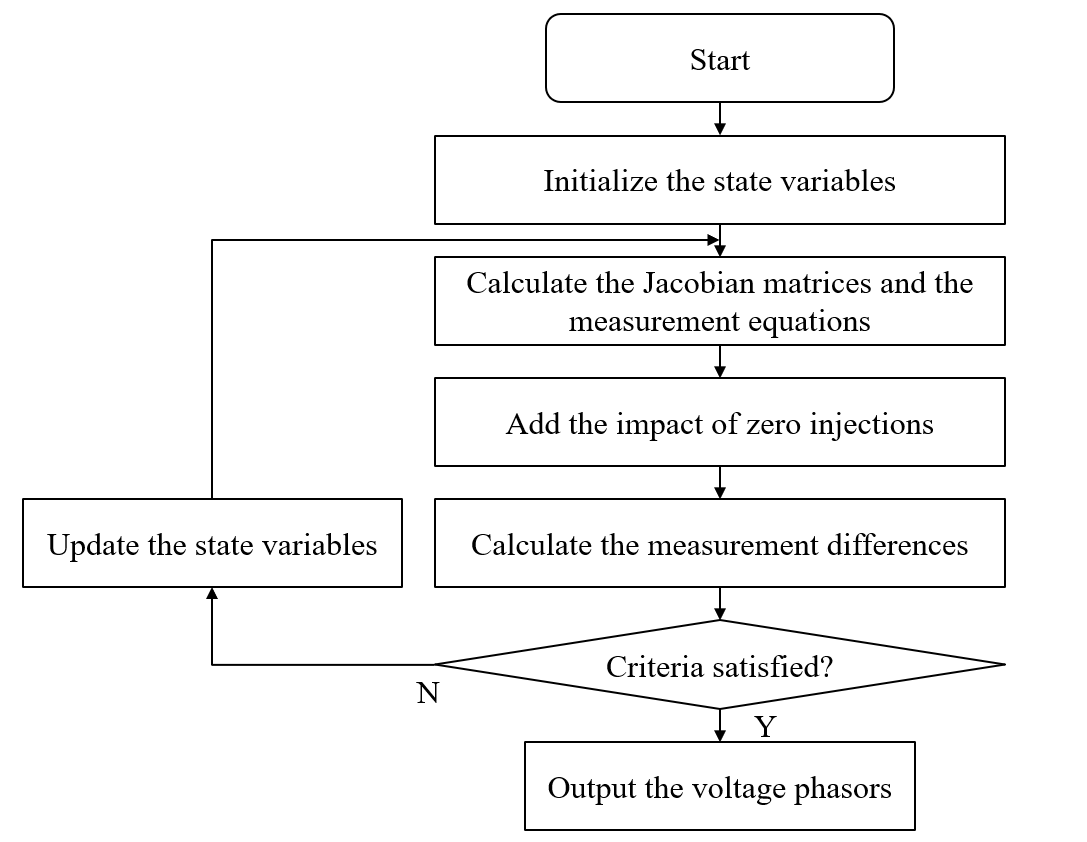


Figure 4‑6 Flow chart of the state estimation algorithm

### 4.3.4 Function inputs and outputs

Table 4‑5 State estimation inputs and outputs

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Feeder information (from model management) | Estimated node voltage |
| Renewed feeder information (from LC and main grid) | Estimated node angle |

## 4.4 Data logging and event recording

### 4.4.1 Functionality description

This function block is to record analog (e.g., frequency or voltage measurements) and binary data (e. g., command or circuit breaker position change) during MG operation. This function block is required for measuring MG system performance and for determining root cause of events. This function block can be also used for diagnosis during debugging and testing.

### 4.4.2 Function Block diagram



Figure 4‑7 block diagram of data logging and event recording function block

### 4.4.3 Function inputs and outputs

Table 4‑6 Data logging and event recording inputs and outputs

|  |  |
| --- | --- |
| **Input** | **output** |
| Analog and binary data indicating microgrid operation status, including frequency, RMS voltage, RMS current, phase angle, real (direction of power flow) and reactive power, energy, demand, PQ (voltage and current harmonic distortions, individual harmonics, voltage sags, voltage swells), reference tracking errors, power quality (distortion); losses and efficiency) | .TDMS files |

## 4.5 PV Forecast

### 4.5.1 Functionality description

This function block aims to make forecasts of the available PV generation. The forecast is based on historical PV generation data/irradiance data. The forecast is used in the Energy Management block.

### 4.5.2 Function block diagram

The block diagram of this function is shown in Figure 10. When enabled, the function reads historical data recorded by the Data Logging function block, puts them through a time series model, and output the forecast.



Figure 4‑8 Block diagram of PV Forecast function block

### 4.5.3 Algorithm flow chart

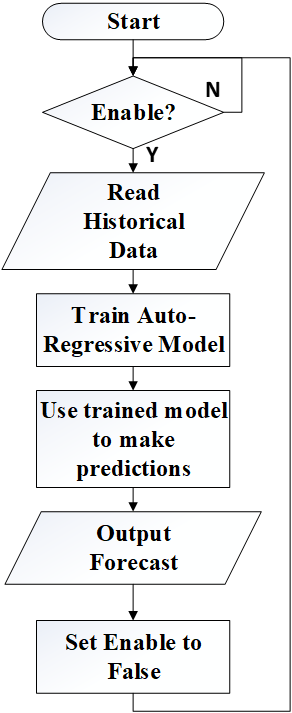


Figure 4‑9 Algorithm flow chart of PV Forecast function block

### 4.5.4 Function inputs and outputs

Table 4‑7 PV forecast inputs and outputs

|  |  |
| --- | --- |
| **Input** | **output** |
| Historical PV generation data | Forecasted PV generation data |
| Enable signal |  |

## 4.6 Load Forecast

### 4.6.1 Functionality description

This function block aims to make forecasts of the electric demand in each load section. The forecast is based on historical load profiles. The forecast is used in the energy management block.

### 4.6.2 Function block diagram

The block diagram of this function is shown in Figure 4‑10. When enabled, the function reads historical data recorded by the Data Logging function block, puts them through a time series model, and outputs the forecast.



Figure 4‑10 Block diagram of Load Forecast function block

### 4.6.3 Algorithm flow chart

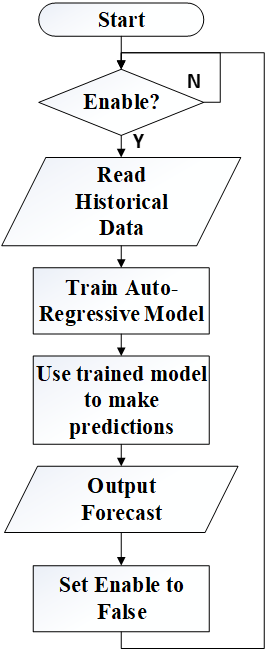


Figure 4‑11 Algorithm flow chart of Load Forecast function block

### 4.6.4 Function inputs and outputs

Table 4‑8 Load forecast inputs and outputs

|  |  |
| --- | --- |
| Input | output |
| Historical load data | Forecasted load data |
| Enable signal |  |

## 4.7 Energy Management

### 4.7.1 Functionality description

This function block aims to make general scheduling/dispatch strategies to reduce the electric bill in grid connected mode, and to provide energy surety to the critical load under islanded mode.

### 4.7.2 Function block diagram

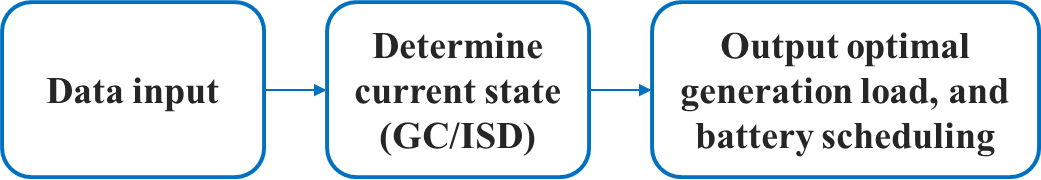


Figure 4‑12 Block diagram of Energy Management function block

### 4.7.3 Algorithm flow chart

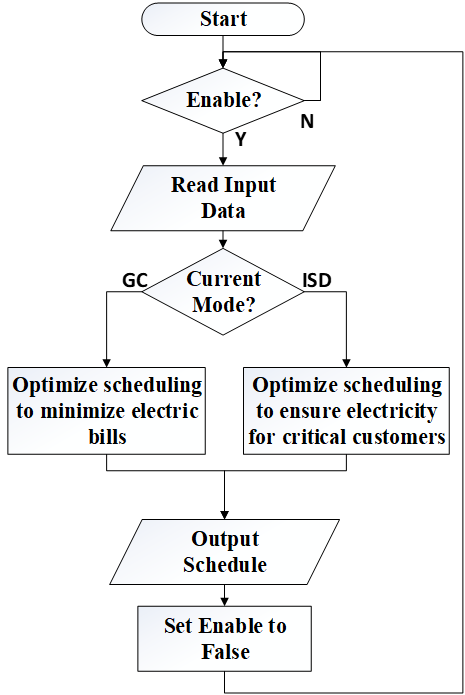


Figure 4‑13 Algorithm flow chart of Energy Management function block

### 4.7.4 Function inputs and outputs

Table 4‑9 Energy management inputs and outputs

|  |  |
| --- | --- |
| **Input** | **output** |
| Load forecast by section | PV generation schedule |
| PV forecast | Battery output schedule |
| Current battery state of charge | Backup generation schedule |
| Energy tariff and demand charge information | Switching schedule |
| Enable signal |  |

## 4.8 Communication

### 4.8.1 Functionality description

This function block(s) enables DNP3 communication from the MGCC to SCADA systems, local controllers, and other remote equipment via the DNP3 network protocol. In its current state, two separate communication function block versions are implemented in the MGCC code. One version acts as a DNP3 master station, while the other acts as a DNP3 outstation. Each communication block is independent, and can be paralleled to enable multiple communication channels, both acting as a master or outstation.

### 4.8.2 Function block diagram



Figure 4‑14 Block diagram of Communication function block

### 4.8.3 Algorithm flow chart

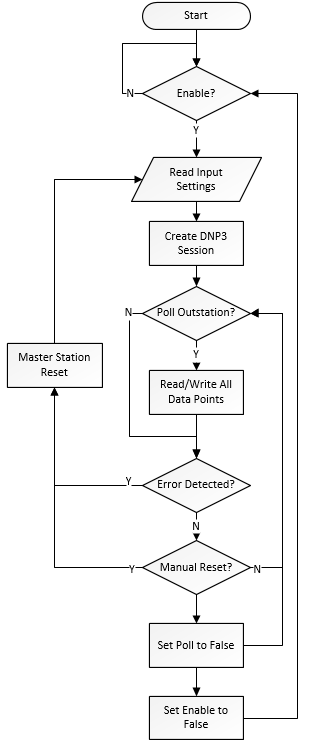


Figure 4‑15 Algorithm flow chart for DNP3 Master Station Comm. Function

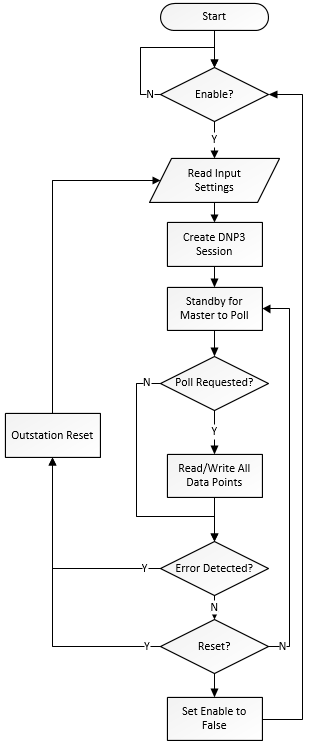


Figure 4‑16 Algorithm flow chart for DNP3 Outstation Comm. Function

### 4.8.4 Function inputs and outputs

Table 4‑10 Communication Master Station Function Version inputs and outputs

|  |  |
| --- | --- |
| Inputs | Outputs |
| Enable Signal | Analog Data Received |
| Poll Signal | Binary Data Received |
| Reset Signal | Error codes (if applicable) |
| Network Settings | Connection Status |
| Communication Settings |  |
| Analog Data to Send |  |
| Binary Data to Send |  |

Table 4‑11 Communication Outstation Function Version inputs and outputs

|  |  |
| --- | --- |
| Inputs | Outputs |
| Enable Signal | Analog Data Received |
| Reset Signal | Binary Data Received |
| Network Settings | Error codes (if applicable) |
| Communication Settings |  |
| Analog Data to Send |  |
| Binary Data to Send |  |

## 4.9 Black Start

### 4.9.1 Functionality description

Black start is to restore the MG from total or partial power loss. A crucial condition for the successful restoration requires a sequence of control actions, which is built in this function block.

### 4.9.2 Function Block diagram

The block diagram for black start is shown in Figure 4‑17. The black start function block requires information from LCs and black start command from central controller. In the initialization state, it gathers information from local controllers and central controller. Then all the load and DERs’ switches will be switched off. After all the disconnecting status information are confirmed, the black start procedure will be initialized. The battery system will be started at the first step. Then the PV will be connected after checking battery’s connecting status. The system will pick up as much load as it can after the PV connected.

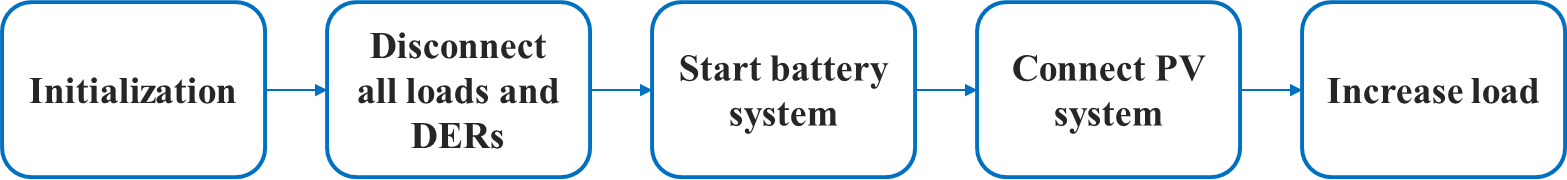


Figure 4‑17 Block diagram for Black Start

### 4.9.3 Algorithm flow chart

The flow chart of black start is shown in Figure 4‑18.

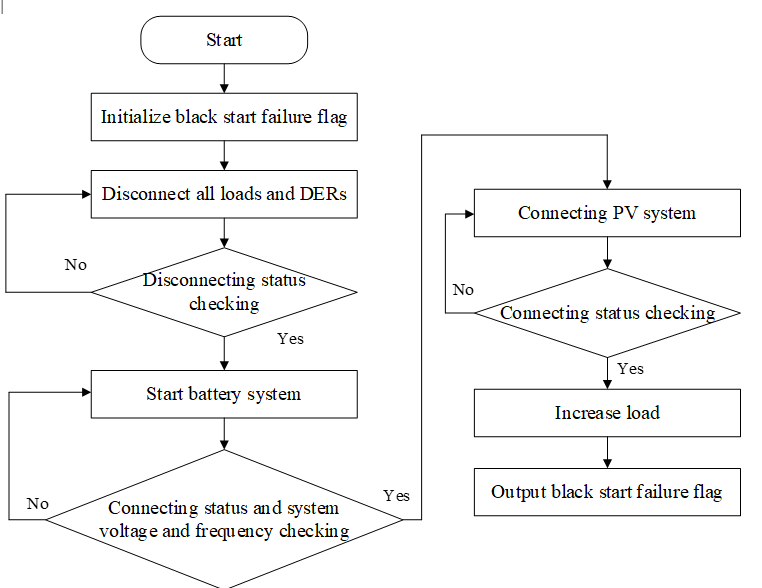


Figure 4‑18 Flow chart of the black start procedure

### 4.9.4 Function inputs and outputs

Table 4‑12 Black Start inputs and outputs

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Local controller information (Switch status, Frequency and voltage) | Switch status reference |
| Central controller information (black start command) | Black start failure flag |

## 4.10 Planned/Unplanned Islanding

### 4.10.1 Functionality description

Planned islanding is designed to make the transition from grid-connected mode to islanded mode with an optimal boundary dependent on the real-time output of DERs. This allows users to intentionally disconnect the microgrid from the grid interface for routine maintenance or per request of main grid operators.

On the other hand, unplanned islanding function is a passive function. Once an unplanned islanding happens, it will identify the islanding state and report to the FSM.

### 4.10.2 Function block diagram

The block diagram of planned islanding is as shown in Figure 4‑19. At the initialization stage, MGCC collects information and measurements from local controllers.

With all the information in place, MGCC determines the optimal boundary that best utilizes the DER output, minimizes the transient during transition to the islanded mode, and maintain stability of the microgrid. Once the optimal boundary is determined, the PCC for the islanding is determined as the point where the grid interfaces the boundary.

The next step is adjusting the DER output such that it matches the total load in the islanded microgrid to minimize the transient during the islanding process. The MGCC will issue a set of active and reactive power reference to the BESS local controller to match the generation and load.

The final step is to check if the DER generation has matched the load well. If yes, the MGCC issues commands to open the IntelliRupter on PCC. If not, the MGCC will repeat the process until the criteria are met.

Similar to the planned islanding, unplanned islanding only has the initialization and detecting state. During the detecting state, with the topology signal from the topology identification, it will identify the current MG state.

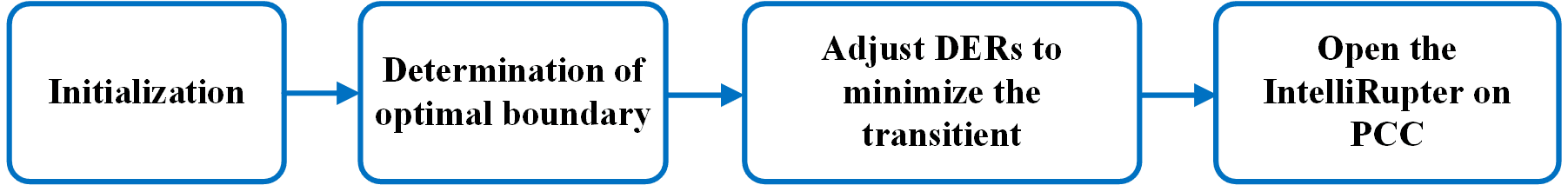


Figure 4‑19 Block diagram of the Planned Islanding function

### 4.10.3 Algorithm flow chart

The flow chart of the planned islanding module is shown in Figure 4‑20.

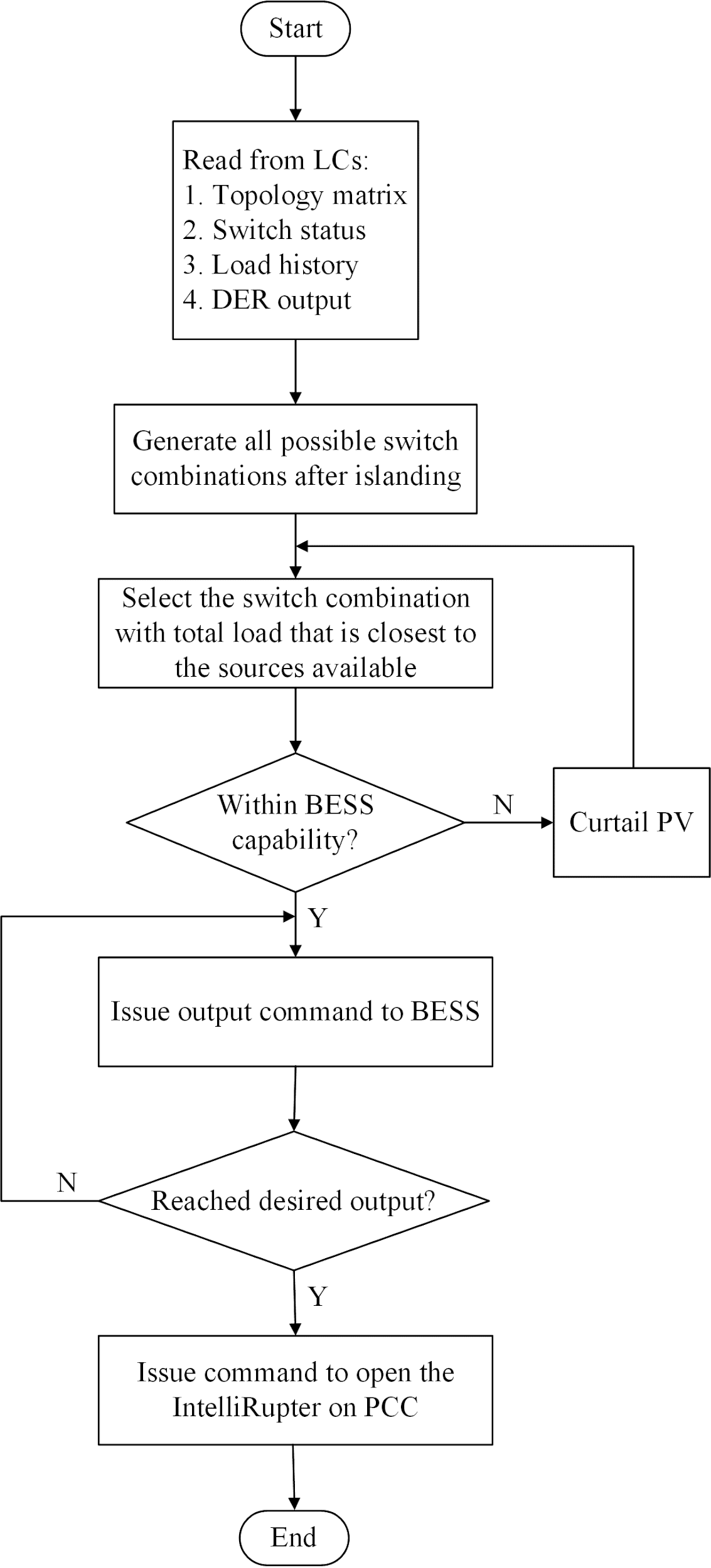


Figure 4‑20 Flowchart of the planned islanding module

### 4.10.4 Function inputs and outputs

Table 4‑13 Inputs and Outputs of the planned islanding function

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Local controller information (Switch status, active and reactive power of DERs) | Switch commands |
| Central controller information (system parameters) | Active and reactive power reference for BESS |

Table 4‑14 Inputs and Outputs of the unplanned islanding function

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Current MG topology | State change signal to FSM |

## 4.11 Fault and abnormal state control

The fault and abnormal state control function is designed to deal with MGCC fault and abnormal cases including communication failure, inputs and outputs errors, and data errors. Under these abnormal cases, the FSM would move from the normal state to the abnormal state with the signal feedback from the fault and abnormal state control function.

As shown in Figure 4‑21, with an abnormal error happens, the fault and abnormal state control function would first find out the error through the feedback from each function block and then identify the error type. If it is an abnormal error, the fault and abnormal state control function would send trigger signal to the FSM and then move into the abnormal state. In the abnormal state, the fault and abnormal state control function would keep checking the error feedback from the functions. If the error disappeared, the fault and abnormal state control function would again send trigger signal to the FSM to move back to the previous state. Otherwise, the FSM would stay in the abnormal state and wait for the further information.

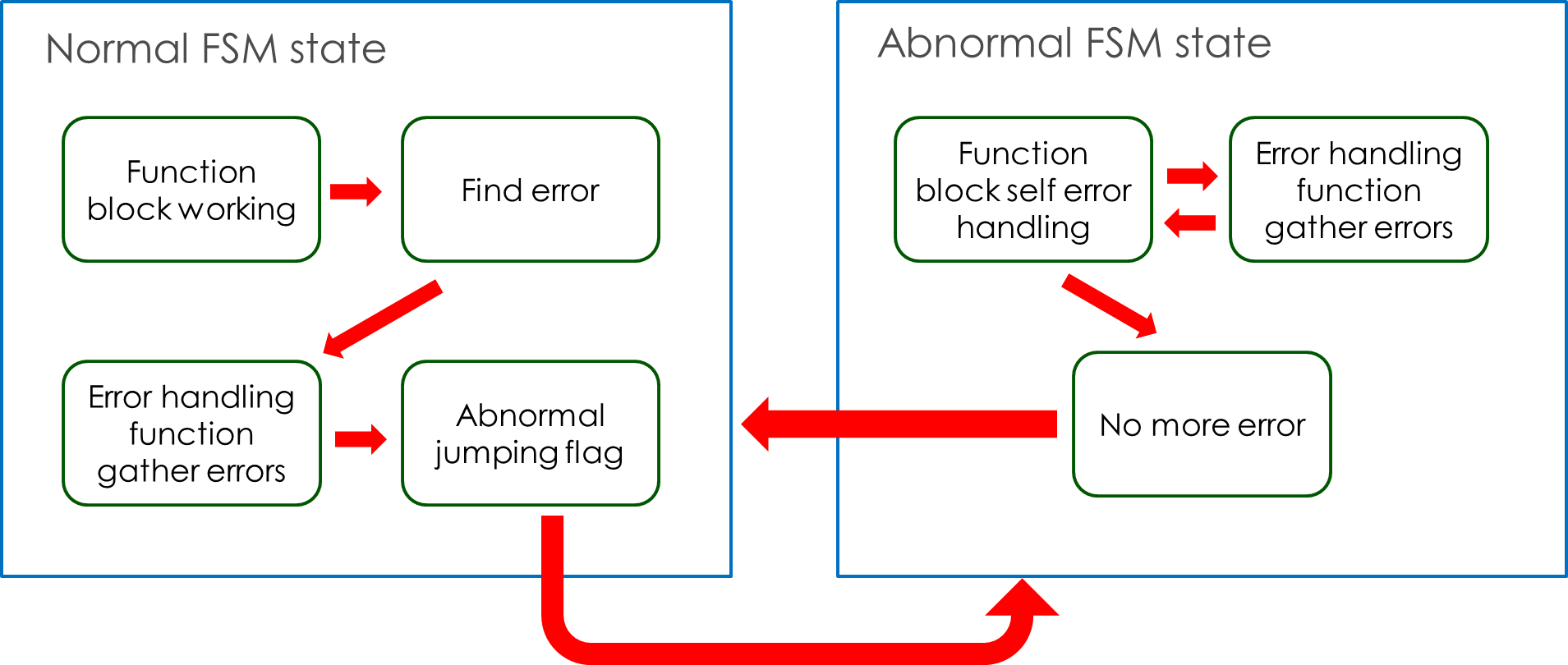


Figure 4‑21 the block diagram of the fault and abnormal state control function with an abnormal error happens

As shown in Figure 4‑22, with a fault error happens, the fault and abnormal state control function would first find out the error through the feedback from each function block and then identify the error type. If it is a fault error, the fault and abnormal state control function would send trigger signal to the FSM and then move into the fault state. In the fault state, the FSM would disable most of the existing functions and showing the fault information to the SCADA. In this case, only if the manually restart can be done on the MGCC.

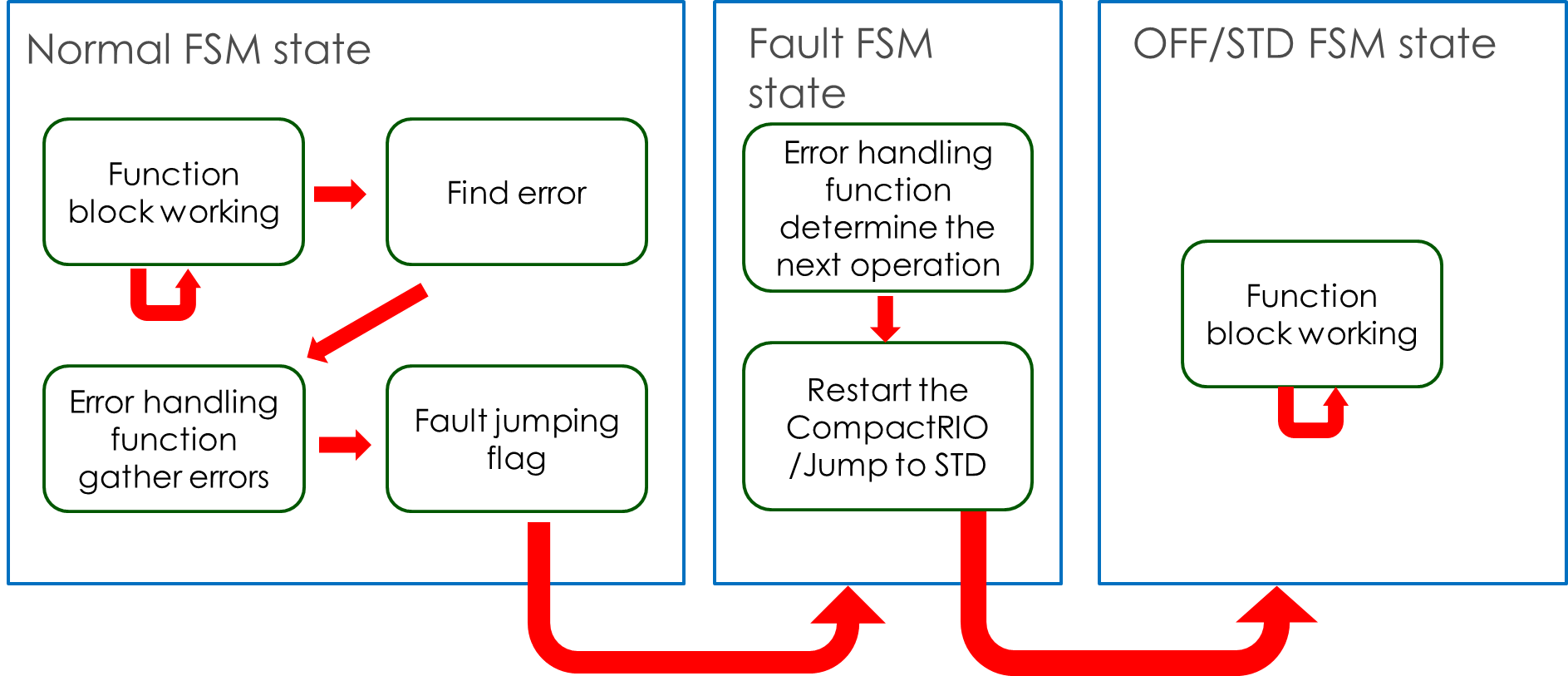


Figure 4‑22 the block diagram of the fault and abnormal state control function with a fault error happens

## 4.12 Resynchronization

This function is described in 5.3

## 4.13 Topology Identification

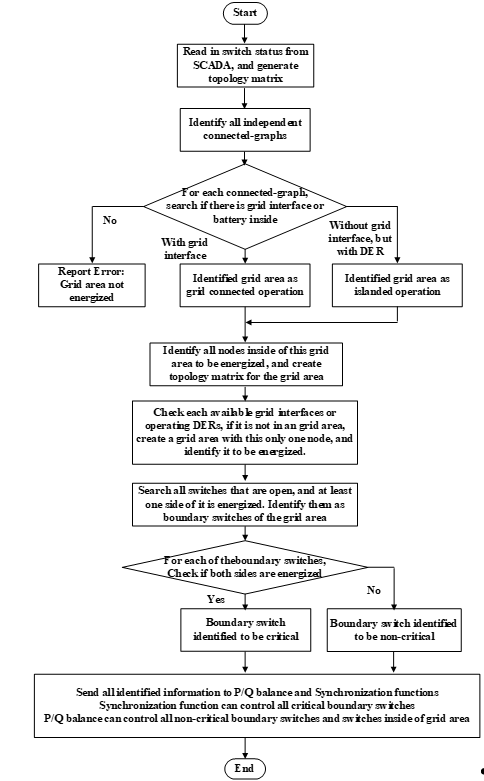


Figure 4‑23 the flowchart of the topology identification

Table 4‑15 PQ balance inputs and outputs

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Switch status | Topology matrix for other functions to use |
| Voltage measurement and Component status to determine the active devices |  |

## 4.14 Islanded state control

The objective of this function is to open the IntelliRupters when unexpected circumstances happen, i.e., after planned islanding or temporal/permanent fault. The flow chart of the function is shown in Figure 4‑25. The basic idea is first to get the enable signal and controllable IntelliRupter list from Finite State Machine and topology identification functions. Then Islanded State Control will determine which IntelliRupter should be opened and thus send out control commands to communication function to open these switches. Note that Islanded State Control function is designed to work with the planned islanding and protection coordination functions.

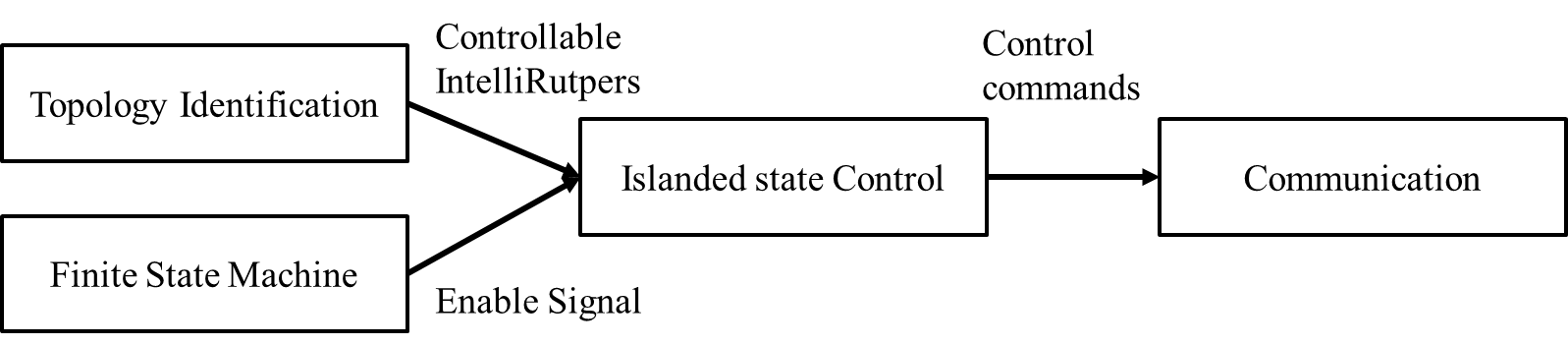


Figure 4‑24 Block diagram of the Islanded State Control function

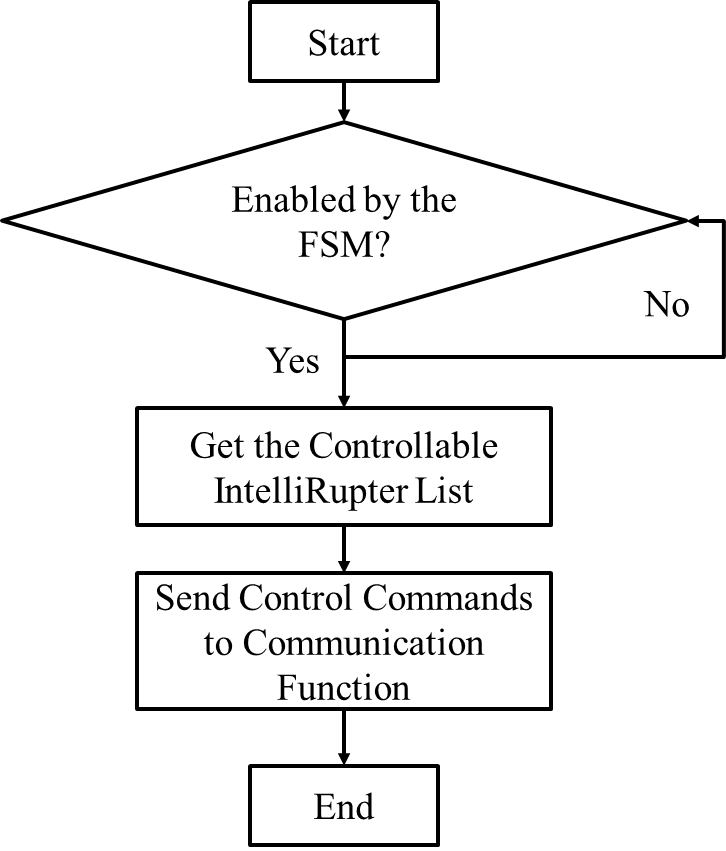


Figure 4‑25 the flow chart of the Islanded State Control function

**5.0 MICROGRID LOCAL CONTROLLER**

# MICROGRID LOCAL CONTROLLER

## 5.1 Active power control

### 5.1.1 Functionality description

This function is in the LC. It controls the output power of the DER during grid-connected operation, and output frequency of the DER during islanded condition. It includes inertia emulation function, P-f droop, secondary control, tertiary control, and curtailment for PV.

### 5.1.2 Function Block diagram

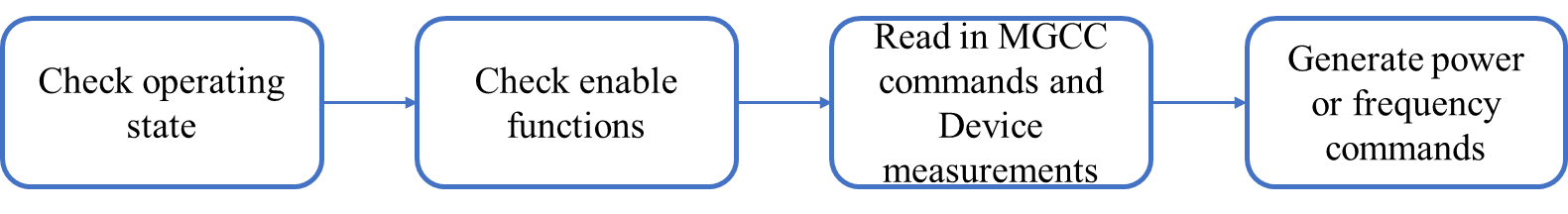


Figure 5‑1 active power control algorithm block diagram

### 5.1.3 Algorithm flow chart

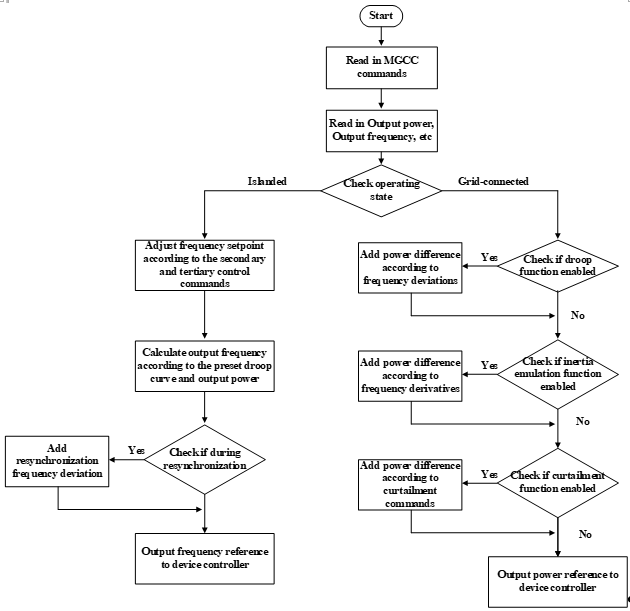


Figure 5‑2 flow chart of the active power control algorithm

### 5.1.4 Function inputs and outputs

Table 5‑1 Active power control inputs and outputs.

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Enable signals from MGCC (inertia emulation, droop, curtailment, etc.) and setpoints (frequency and power, curtailment percentage, etc.) | Output power reference in grid connected state. |
| Measurements from device controllers (output power, grid frequency, etc.) | Output frequency reference in islanded state |

## 5.2 Reactive power control

### 5.2.1 Functionality description

This function is in the LC. It controls the output reactive power of the DER. It includes Q-v droop, voltage closed-loop regulation, and constant power factor control.

### 5.2.2 Function Block diagram

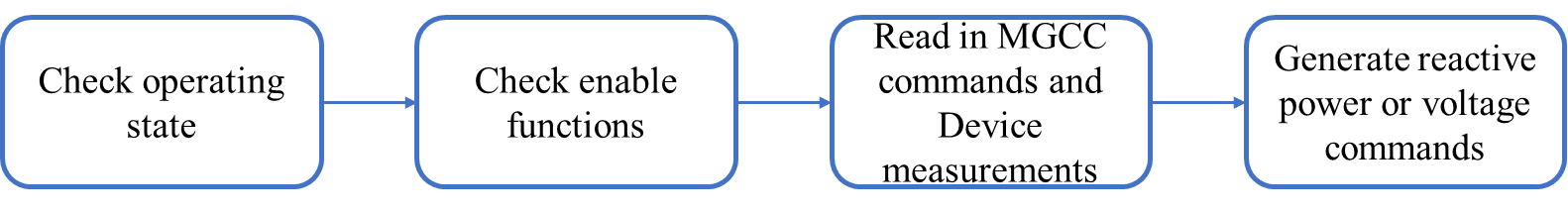


Figure 5‑3 reactive power control algorithm block diagram

### 5.2.3 Algorithm flow chart

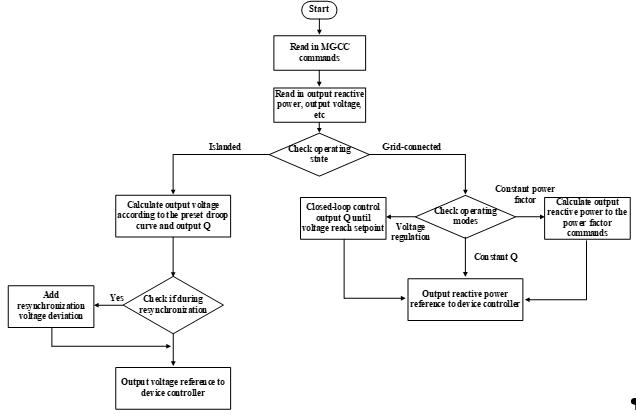


Figure 5‑4 flow chart of the reactive power control algorithm

### 5.2.4 Function inputs and outputs

Table 5‑2 reactive power control inputs and outputs.

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Enable signals from MGCC (Q-v droop, voltage regulation, etc.) and setpoints (reactive power or voltage | Output reactive power reference in grid connected state. |
| Measurements from device controllers (output power, output reactive power, grid voltage magnitude, etc.) | Output voltage reference in islanded state |

## 5.3 Resynchronization

### 5.3.1 Functionality description

This function is in MGCC and LCs, and will only run in the I2G state of the FSM. The MGCC part identifies the boundary switch of the island, and send the voltage magnitude, frequency and angle difference between the two side of the boundary switch to the LC. LC part controls the DER’s output voltage to reduce the voltage differences. Once the differences are smaller than a preset threshold, the MGCC part will issue commands to close the boundary switch, and reconnect the island to the main grid.

### 5.3.2 Function Block diagram

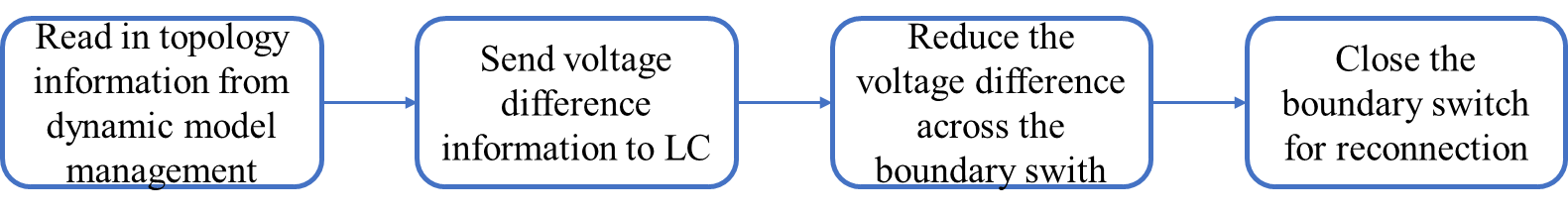


Figure 5‑5 resynchronization algorithm block diagram

### 5.3.3 Algorithm flow chart

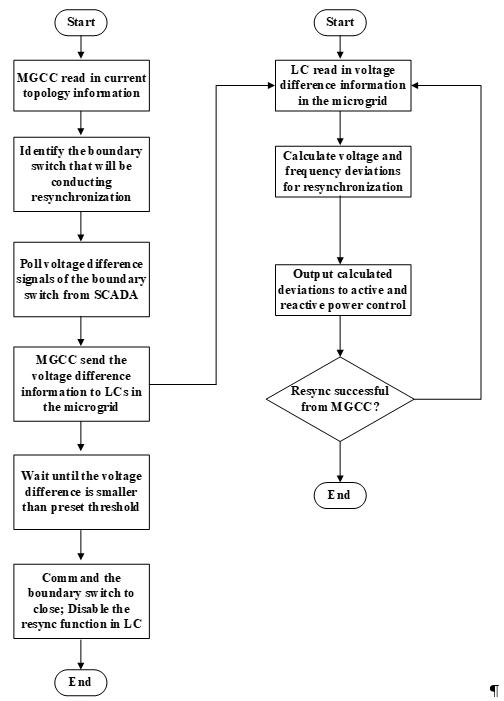


Figure 5‑6 flow chart of the resynchronization algorithm

### 5.3.4 Function inputs and outputs

Table 5‑3 Resynchronization MGCC part inputs and outputs.

|  |  |
| --- | --- |
| Inputs | Outputs |
| Topology information from dynamic model management | Voltage magnitude difference; Frequency difference; and angle difference to local controllers |
| Voltage magnitude difference; Frequency difference; and angle difference from SCADA |  |

Table 5‑4 Resynchronization LC part inputs and outputs.

|  |  |
| --- | --- |
| Inputs | Outputs |
| Enable signals from MGCC | Output voltage and frequency reference |
| Voltage magnitude difference; Frequency difference; and angle difference |  |
| Measurements from device controllers (output power, grid frequency, etc.) |  |

**6.0 INSTRUCTION FOR UTILIZING THE MICROGRID CONTROLLER WITH A NEW FEEDER**

# INSTRUCTION FOR UTILIZING THE MICROGRID CONTROLLER WITH A NEW FEEDER

## 6.1 Microgrid topology acquiring

In order to utilize the proposed microgrid controllers with a new feeder, the first step is to acquire the microgrid topology. There are two popular software tools to set up the microgrid single-line topology, i.e., CYME and SynerGEE Electric. Here, taking CYME as an example, following the steps discussed below, the inputs of the model management can be generated. One thing needs to mention is that this method requires MATLAB 2016 or advanced version to support.

As shown in Figure 6‑1, the first step is to build up a detailed microgrid model in CYME. Figure 6‑2 gives the full model for EPB system as an example. Then, you can push the button shown in Figure 6‑3 to get the reduced model from the full model. The next step is to generate the ASCII file from the reduced model.

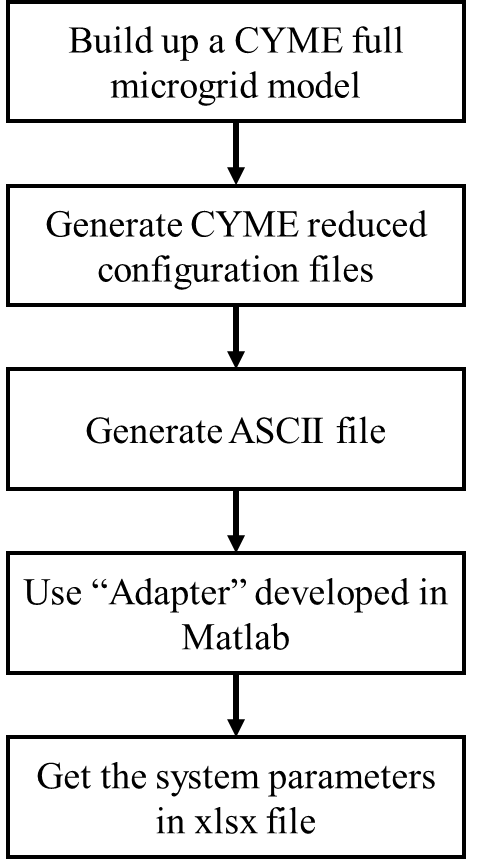


Figure 6‑1 the flow chart of using CYME and “Adapter” to generate the input data

The exported data from CYME contains three .txt files. The load.txt contains the load information of each node. The network.txt contains the topology information and the coordinates. The equipment.txt contains the configuration of each branch. Given the steps from Figure 6‑4, the output ASCII file can be searched, with an example of “load.txt” shown in Figure 6‑5. After that, you may need to use another self-defined MATLAB function, “Adapter” to support. Please put the three output files of the ASCII (txt files named ‘load.txt’, ‘network.txt’ and ‘equipment.txt’) to the same folder of the “Adapter” and then run the “Adapter” in MATLAB environment.

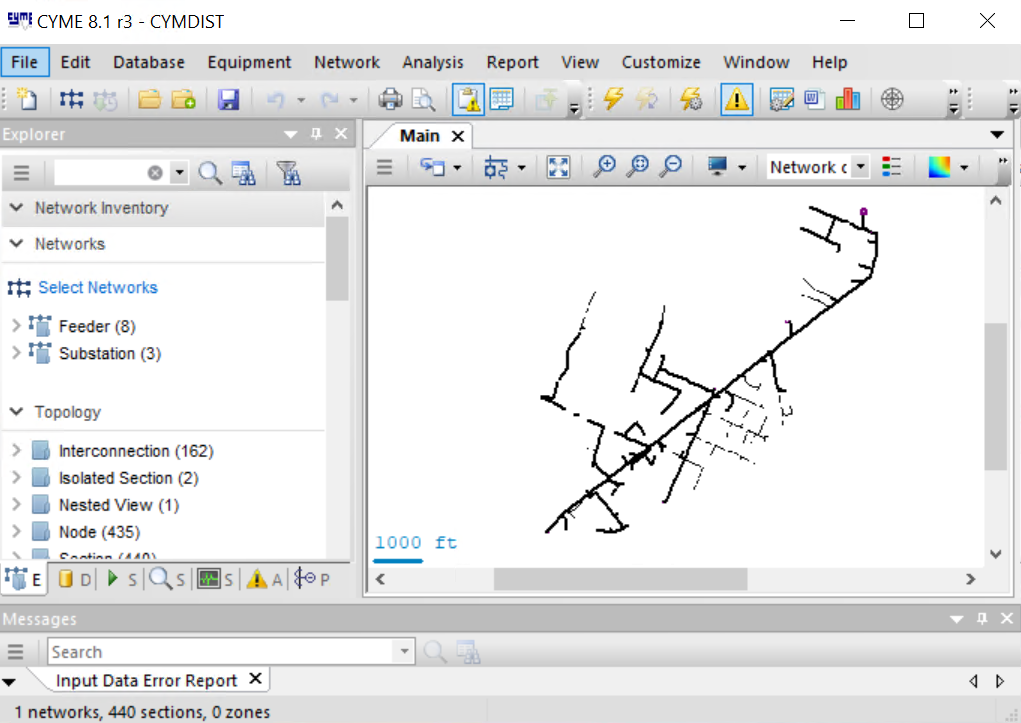


Figure 6‑2 the full feeder model in EPB system

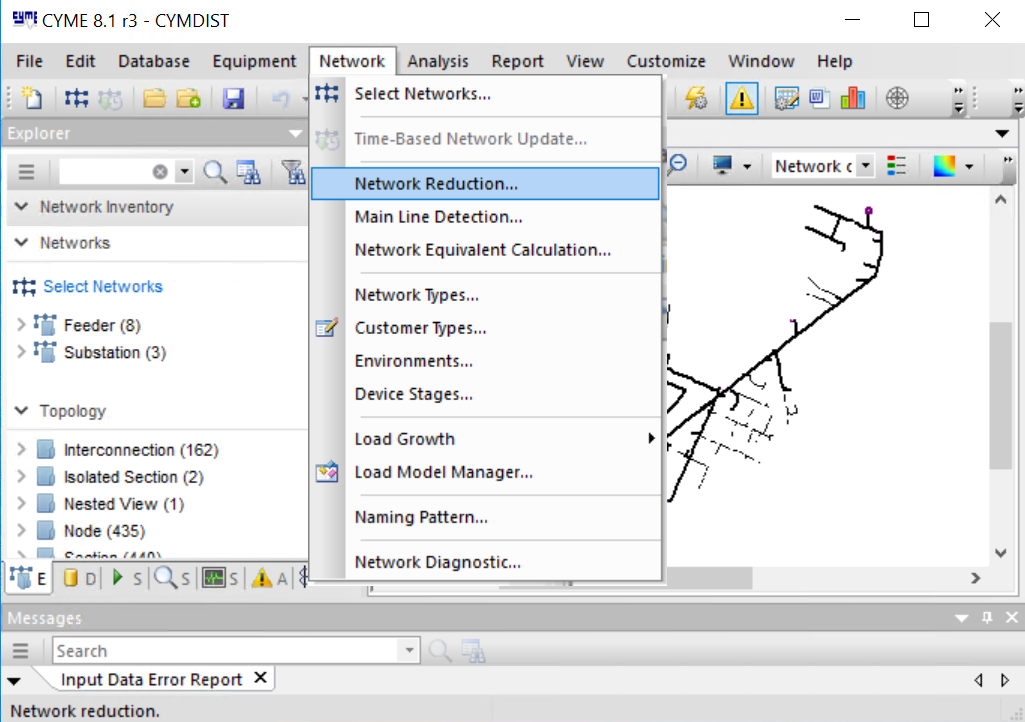


Figure 6‑3 Model reduction using CYME

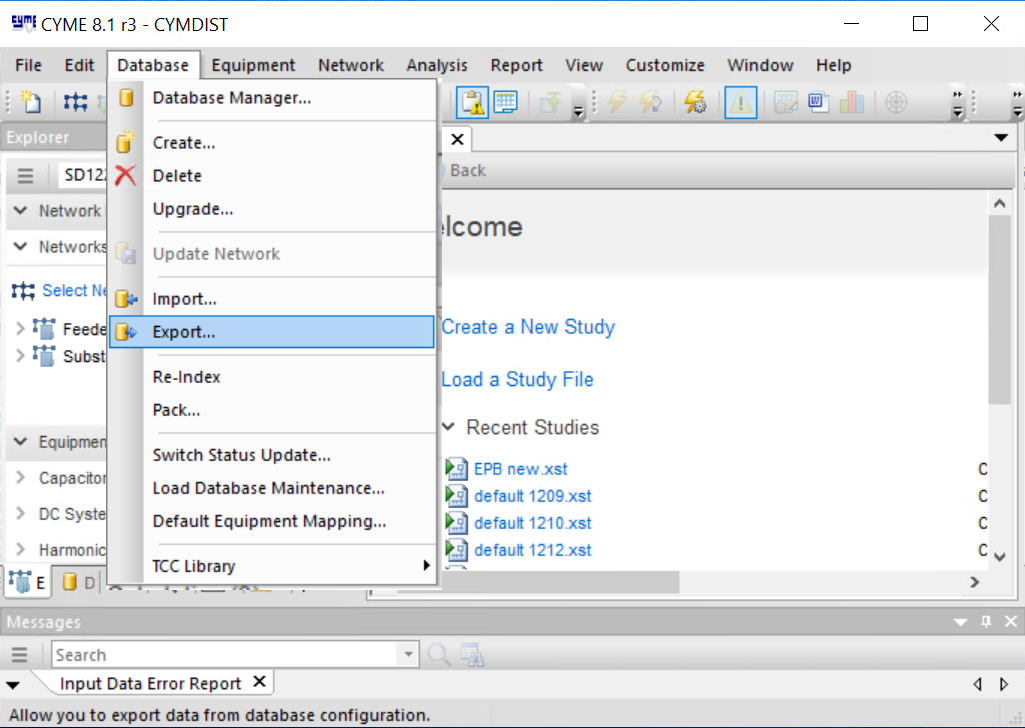


Figure 6‑4 ASCII file Export using CYME



Figure 6‑5 Example of load.txt file

Then you will get the system parameters in “User-defined name.xlsx” file under the same folder. In the “User-defined name.xlsx” file, there are eight spreadsheets. Given an example from IEEE 123-node system, as shown in Figure 6‑6 to Figure 6‑10, they are general data, topology data, configuration data, load data, and graphic data. In addition to these five spreadsheets, three feeder data named “V\_node.csv”, “Y\_bus.csv”, and “I\_node.csv” are also generated by the “adapter”. Besides, the “Adapter” will also generate a separate csv file named “Intellirupter.csv” to indicate which line has an IntelliRupter installed.

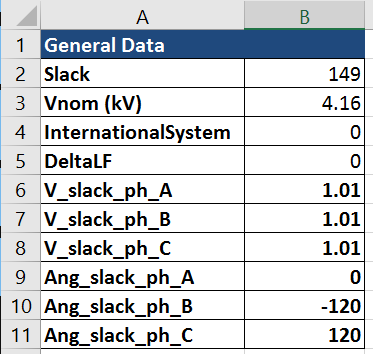


Figure 6‑6 Spreadsheet of general data.

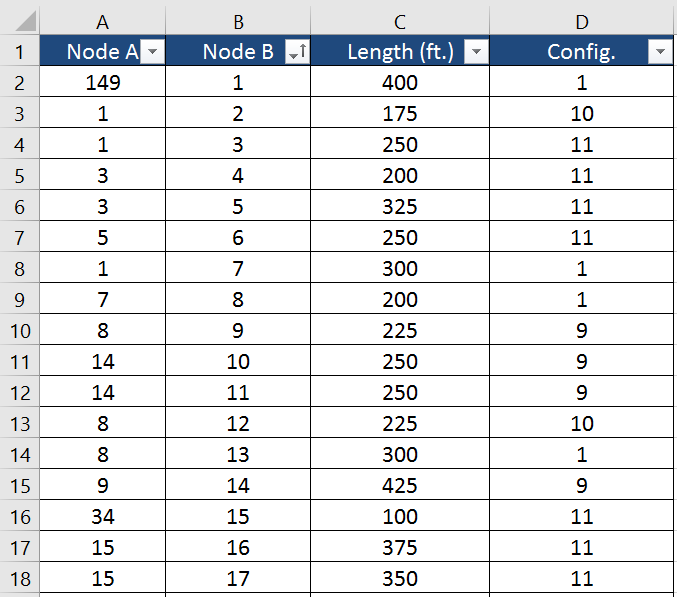


Figure 6‑7 Spreadsheet of topology data.

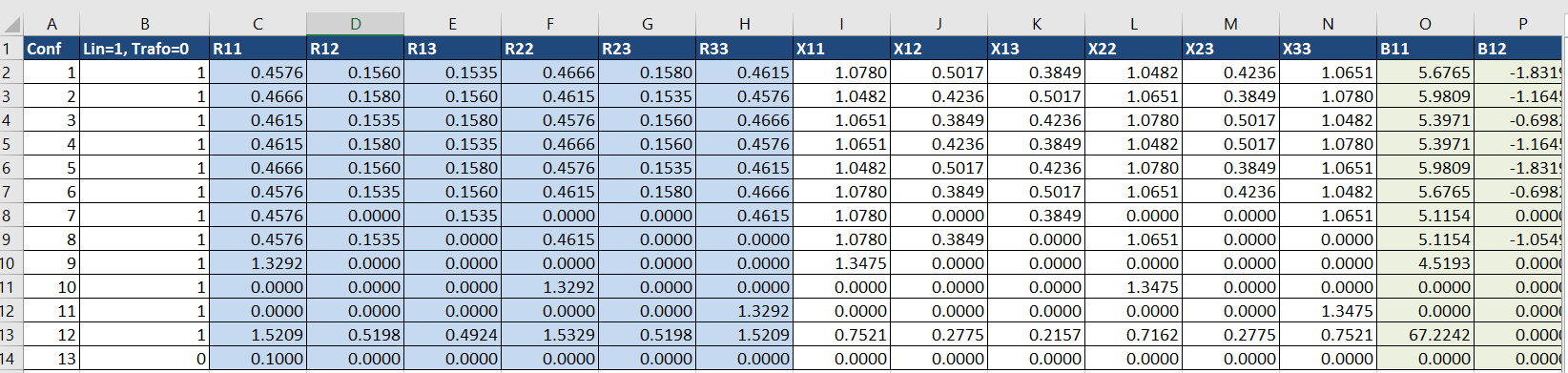


Figure 6‑8 Configuration data spreadsheet.

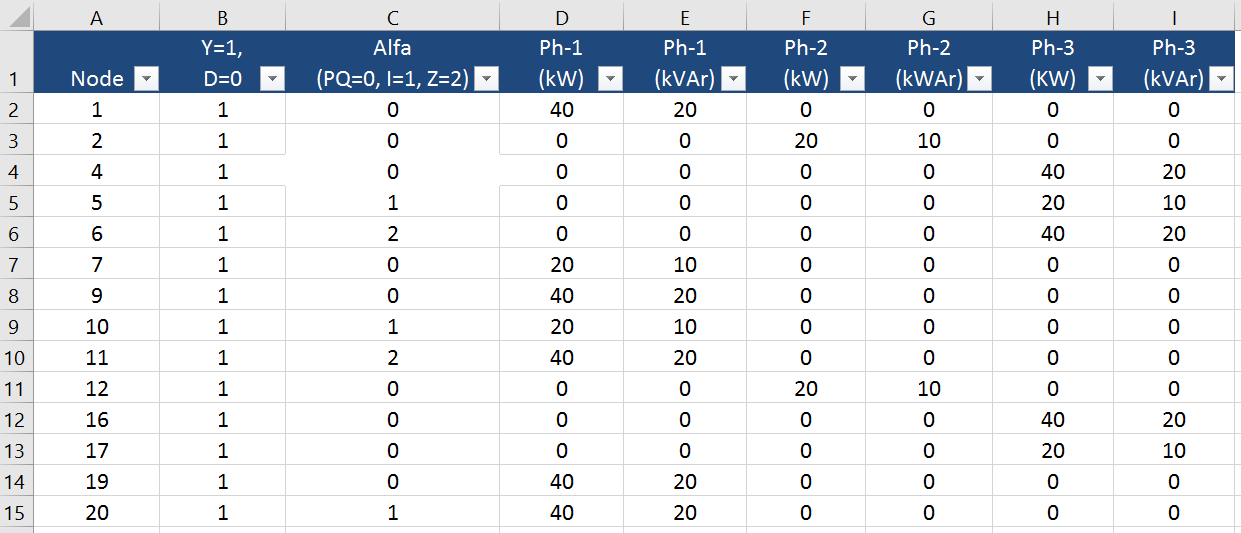


Figure 6‑9 Load data spreadsheet.

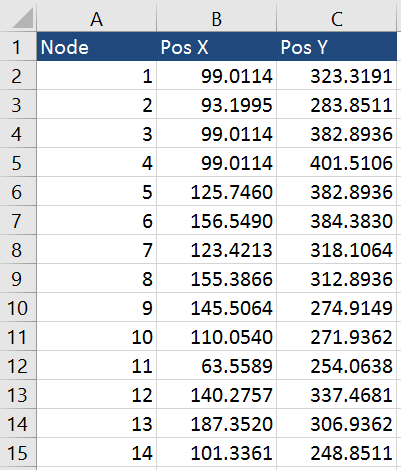


Figure 6‑10 Graphic data spreadsheet.

In addition to the above data, the BESS and PV information should be ready before generating the input of the model management. In order to create this information with new MG feature, please open the “BESS\_parameter.csv” file with excel. As shown in Figure 6‑11, the sequence of the BESS parameters follows the following sequence: BESS location (which node is connecting with a BESS), BESS capacity, BESS power limitation, BESS SOC minimum, BESS SOC maximum, BESS voltage minimum, BESS voltage maximum, BESS frequency minimum, BESS frequency maximum, BESS number, and BESS inverter current limitation. Please **fill in the related BESS location** according to you own MG topology. Similar to the BESS, the PV parameter are listed in “PV\_parameter.csv”. The PV parameters follow the sequence: PV location (which node is connecting with a PV), PV number and PV inverter current limitation. Please **fill in the related PV location** according to you own MG topology.

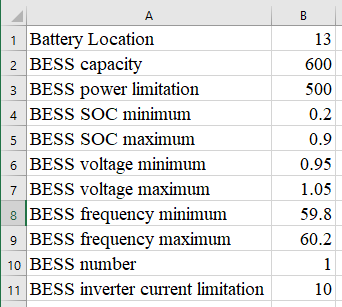


Figure 6‑11 BESS parameter CSV file

In addition, the “Adapter” will provide an individual table for the locations of the IntelliRupters. For example, the location of the IntelliRupters in Opal-RT model is listed in Table 6‑1. Note that, here “Y” for an IntelliRupter, “G” for a grid interface; “N” for a normal line. If the microgrid is designed without IntelliRupters , please still let us know the location of the grid interfaces. This table is stored in the “IntelliRutper.csv”.

Table 6‑1 The location for the IntelliRupters

|  |  |  |
| --- | --- | --- |
| Start Node | End Node | Line Type |
| 1 | 2 | G |
| 2 | 3 | N |
| 3 | 4 | Y |
| 4 | 5 | N |
| 5 | 6 | Y |
| 6 | 7 | N |
| 7 | 8 | Y |
| 7 | 9 | N |
| 9 | 10 | G |
| 4 | 11 | N |
| 11 | 12 | Y |
| 12 | 13 | N |
| 13 | 14 | Y |
| 14 | 15 | N |
| 15 | 16 | Y |
| 16 | 17 | N |
| 17 | 18 | Y |
| 18 | 19 | N |
| 19 | 20 | G |
| 16 | 21 | N |
| 12 | 22 | Y |

Besides, the PQ balance function may need some initial settings such as the PQ balance working cycle, PQ recharge min SOC, and PQ discharge max SOC. If the users do not want to change any default setting, please leave it there. These parameters are listed in the file named “PQ.csv”.

## 6.2 Create the inputs for the model management

The next step is to create the inputs for the model management function. Please open the NI-MAX and select the compact of your MGCC, shown in Figure 6‑12.

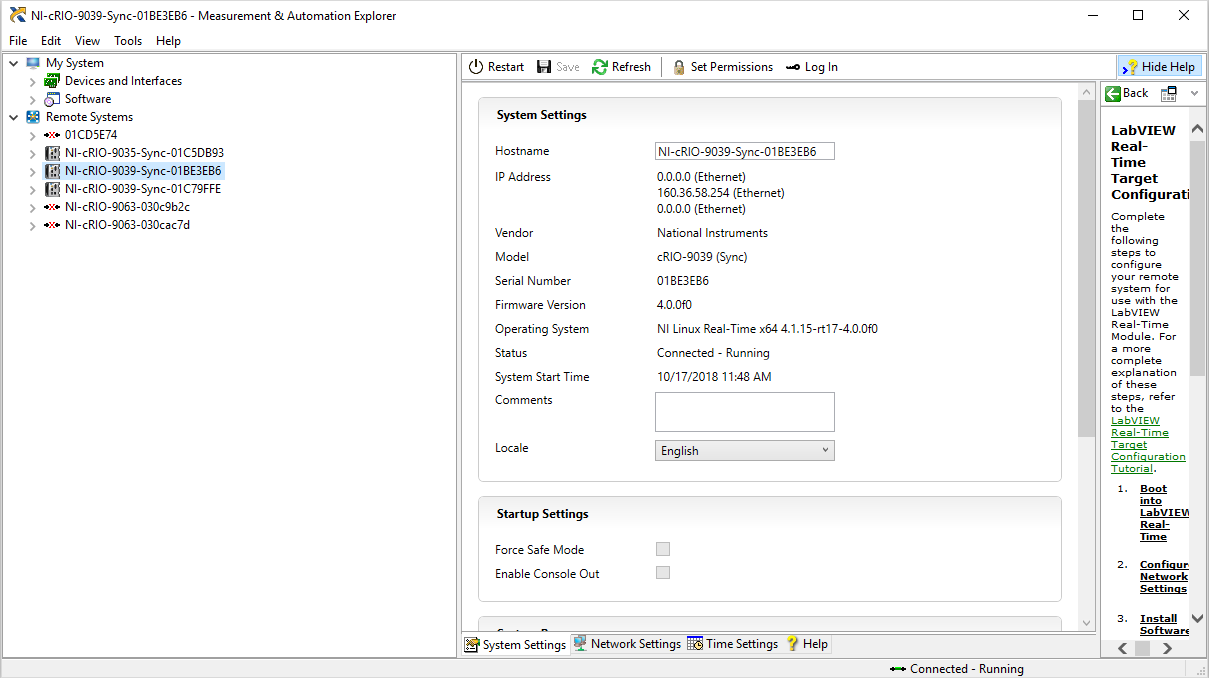


Figure 6‑12 Interface of the NI MAX

Then, please right click the NI-CRIO and choose “file transfer”. If it jumps to EI explorer, please retry the previous step again and check whether you have successfully installed the NI-LabVIEW FTP module. If it jumps to the FTP of the CompactRIO, then please find the “c” folder and go to “model management” inside of it. The next step to move the csv files listed in Table 6‑2 to that folder. If everything succeeds, please close the folder and it is done with this phase.

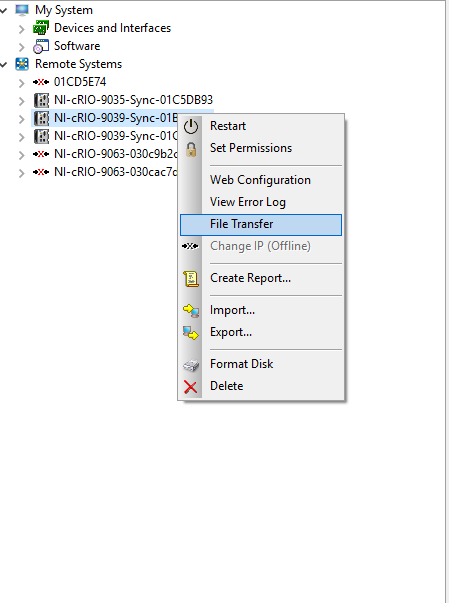


Figure 6‑13 file transfer selection

Table 6‑2 Conclusion of a model management file package for transferring to a new MG

|  |  |
| --- | --- |
| File name | File come from |
| BESS\_parameter.csv | Model management package |
| PV\_parameter.csv | Model management package |
| I\_node.csv | “Adapter” |
| V\_node.csv | “Adapter” |
| Y\_bus.csv | “Adapter” |
| Topology.csv | “Adapter” |
| IntelliRupter.csv | “Adapter” |

## 6.3 Create the communication channels

Beside of the model management, communication channels are required to be initialized. Since the communication channels are based on the number of the IntelliRupters, BESSs, and PVs, you may need some data from the “BESS\_parameter.csv”, “PV\_parameter.csv”, and “IntelliRupter.csv”.

First, for the “IntelliRupter.csv” it mentions all the IntelliRupter locations. The communication data comes from these smart switches. For each IntelliRupter, we have defined 15 variables, listed in Table 6‑3. Here, taking S0102 as an example IntelliRupter, all the description of the communication data point is given. The sequence of the IntelliRupters communication channels follows the same one listed in the “IntelliRupter.csv”.

Table 6‑3 The communication specification of S0102 IntelliRupter

|  |  |
| --- | --- |
| Variable Name | Variable meaning |
| S0102-status | On/Off status |
| S0102-MAG-A-L | Phase A Voltage magnitude on left side of the S0102 |
| S0102-MAG-B-L | Phase B Voltage magnitude on left side of the S0102 |
| S0102-MAG-C-L | Phase C Voltage magnitude on left side of the S0102 |
| S0102-MAG-A-R | Phase A Voltage magnitude on right side of the S0102 |
| S0102-MAG-B-R | Phase B Voltage magnitude on right side of the S0102 |
| S0102-MAG-C-R | Phase C Voltage magnitude on right side of the S0102 |
| S0102-FREQ-L | Frequency on left side of the S0102 |
| S0102-FREQ-R | Frequency on right side of the S0102 |
| S0102-ANG-DIFF | Angle difference between two sides of the S0102 |
| S0102-APF | Active power flow through the S0102 |
| S0102-RPF | Reactive power flow through the S0102 |
| S0102-Reserved13 | Reserved data location 1 |
| S0102-Reserved14 | Reserved data location 2 |
| S0102-Reserved15 | Reserved data location 3 |

In order to establish the communication channels between the SCADA and the MGCC, the LCs and MGCC, the BESS/PV and LCs. It is necessary to setup the DNP3 communication channels with predefined input IP address and ports. In order to setup stable communication channels, please fill in the related communication channel tables. For example, in the “Communication\_BESS\_LC.csv” the first column means the parameters required for the outstation communication between LC and MGCC while the second column means the master communication between LC and BESS. The “Communication\_PV\_LC.csv”, “Communication\_BESS\_BESS.csv”, “Communication\_PV\_PV.csv” are almost the same with different meanings. The explanations of these inputs are listed in Table X. One thing needs to mention is that for the “Communication\_intellirupters\_MGCC.csv”, there are four columns for BESS master, PV master, SCADA master, and SCADA outstation, respectively. Considering the real-time data requirement, for the SCADA, it is designed as bidirectional master and outstation ones.

The next step is to generate correct communication channels. For the SCADA, it has been discussed in Table 6‑3 and related paragraphs. For BESS and PV, please refer to Table 6‑4 for detailed information.

Table 6‑4 Communication specification of BESS

|  |  |  |  |
| --- | --- | --- | --- |
| BESS status name | Meaning | BESS control name | Meaning |
| BESS-STATUS | BESS on/off status | BESS-STATUS-CTRL | BESS on/off control |
| BESS-MAG | BESS voltage magnitude | BESS-APC | BESS active power control |
| BESS-FREQ | BESS frequency | BESS-RPC | BESS reactive power control |
| BESS-APO | BESS active power output | BESS-FREQ-CMD | BESS frequency command |
| BESS-RPO | BESS reactive power output | BESS-VOLT-CMD | BESS voltage command |
| BESS-SOC | BESS SOC |  |  |
| BESS-Reserved7 | Reserved |  |  |
| BESS-Reserved8 | Reserved |  |  |
| BESS-Reserved9 | Reserved |  |  |
| BESS-Reserved10 | Reserved |  |  |
| BESS-Reserved11 | Reserved |  |  |
| BESS-Reserved12 | Reserved |  |  |
| BESS-Reserved13 | Reserved |  |  |
| BESS-Reserved14 | Reserved |  |  |
| BESS-Reserved15 | Reserved |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| PV status name | Meaning | PV control name | Meaning |
| PV-STATUS | PV on/off status | PV-STATUS-CTRL | PV on/off control |
| PV-MAG | BE PV SS voltage magnitude | PV-APC | PV active power control |
| PV-FREQ | PV frequency | PV-RPC | PV reactive power control |
| PV-APO | PV active power output | PV-RESERVED4 | Reserved |
| PV-RPO | PV reactive power output | PV-RESERVED5 | Reserved |
| PV-AAP | PV Irradiance | PV-RESERVED6 | Reserved |
| PV-RESERVED7 | Reserved | PV-RESERVED7 | Reserved |
| PV-RESERVED8 | Reserved | PV-RESERVED8 | Reserved |
| PV-RESERVED9 | Reserved | PV-RESERVED9 | Reserved |
| PV-RESERVED10 | Reserved |  |  |
| PV-RESERVED11 | Reserved |  |  |
| PV-RESERVED12 | Reserved |  |  |
| PV-RESERVED13 | Reserved |  |  |
| PV-RESERVED14 | Reserved |  |  |
| PV-RESERVED15 | Reserved |  |  |

After filling in these CSV files, please make sure put these files into correct locations. For PV LCs, please put it into the model management folder in the PV FTP while please do the same thing to MGCC and BESS LC. For the BESS, PV, and SCADA, please set up related DNP3 setting in your system.

Table 6‑5 Example communication configuration input table

|  |  |  |
| --- | --- | --- |
|  | outstation settings | Master settings |
| IP1 |  | 160 |
| IP2 |  | 36 |
| IP3 |  | 56 |
| IP4 |  | 202 |
| DNP3 port |  | 18000 |
| DNP3 Master Address |  | 7 |
| DNP3 outstation Address |  | 8 |
| Analogue statues channel number |  |  |
| Analogue control channel number |  |  |
| Digital status channel number |  |  |
| Digital control channel number |  |  |

Table 6‑6 All input csv files for the communication configuration

|  |  |
| --- | --- |
| Configuration file name | Configuration file meaning |
| Communication\_BESS\_LC.csv | The communication input for BESS local controller |
| Communication\_PV\_LC.csv | The communication input for PV local controller |
| Communication\_BESS\_BESS.csv | The communication input for BESS |
| Communication\_PV\_PV.csv | The communication input for PV |
| Communication\_intellirupters\_MGCC.csv | The communication input for MGCC |
| Communication\_intellirupters\_SCADA.csv | The communication input for SCADA |

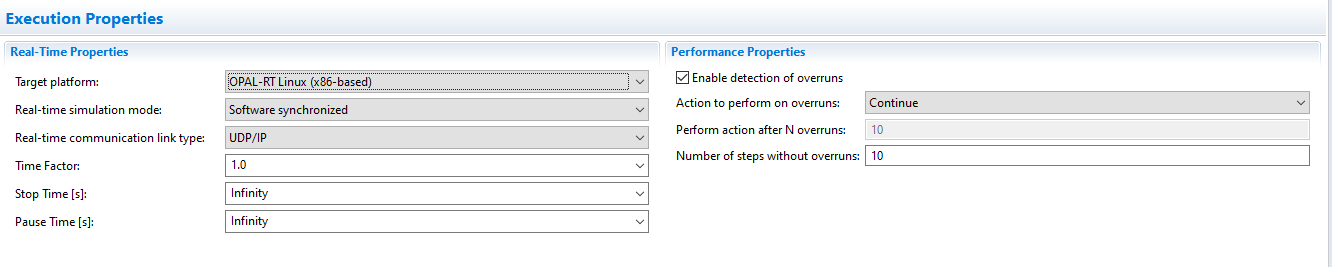
## 6.4 Set up the Opal-RT model for controller test

The model used for hardware-in-the-loop (HIL) testing has been created in MATLAB/Simulink and uses the Opal-RT platform to interact with the microgrid controller which is implemented on NI’s hardware - CompactRIO. The model is available at Opal-RT-PhaseTwo OpenSource Release\OpalRT model. The circuit, which is based on the GridLAB-D taxonomy feeder graph R1-12.47-4 developed by PNNL (<http://item.bettergrids.org/handle/1001/155>) [4], has been modified to include one point of main grid connection, two battery energy storage systems, one PV array system, and one backup generator. The circuit is separated into eight load sections by smart switches and can be split into two sub-microgrids. Once sub-microgrid is supported by one battery, the PV system, and the generator, while the other one is supported by the second battery. The MATLAB/Simulink model file (.mdl) is divided into three subsystems which interact with each other:

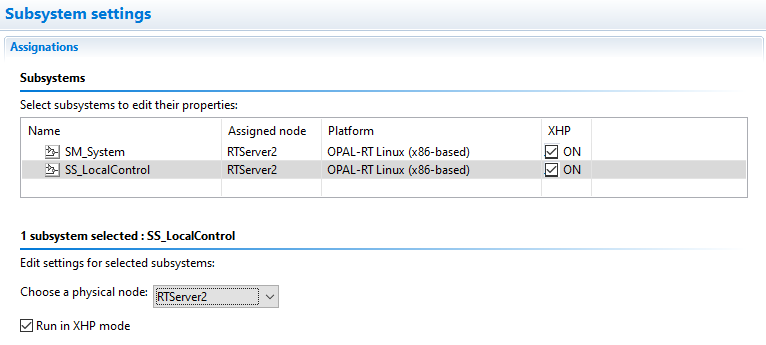
1. SM\_Subsystem is essentially the main circuit model comprising of sources, loads, line sections, transformers and switches which interacts with and sends measurements to the Opal-RT console and the controller in the loop. This layer in inaccessible when the model is running.
2. SC\_Console provides access to the model when it is running in real-time to those elements that are not controlled by the hardware.
3. SS\_LocalControl includes the underlying models for the batteries and PV system used in the main circuit. This layer is also inaccessible when the model is running.

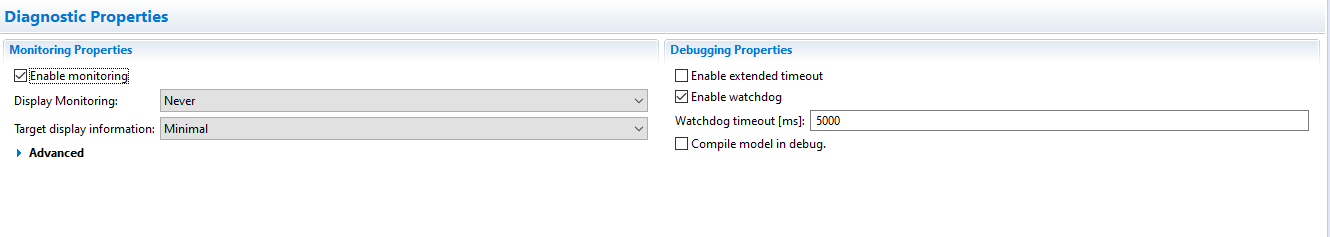
This model requires MATLAB version R2016b and any RT-Lab version compatible with MATLAB R2016b as well as the Opal-RT target. Information regarding RT-Lab versions and installation instructions can be found on the Opal-RT website (<http://www.opal-rt.com>). After installing RT-Lab and connecting the Opal-RT target, create a new project and import the .mdl file into it. Before building, confirm the following settings:

1. The simulation mode is S*oftware synchronized* in **Execution Properties**:



1. Ensure the subsystems are appropriately assigned with XHP enabled in **Subsystem Settings**:



1. Disable extended timeout and enable watchdog in **Diagnostic Properties**Once these settings are confirmed, the model can be built to run on the target. The model must be rebuilt every time the file is changed. After building the model, load the model on to the target and the console will popup. The model is now ready for testing. Run the model using the *Execute* button, pause it using the *Pause* button, and stop and clear the model using the *Reset* button. To restart the model, reload and execute it.

## 6.5 Initialize the microgrid controllers

In this section, the initialization procedure is given for the microgrid controllers. Since all the controllers are programmed with startup setting, the model management and the communication settings will be read once they are powered on. So that when the controllers are powered on, please use the NI MAX to upload the related csv files and then restart the controllers to renew the settings. After this step, the controllers are all under off state and waiting for further commands from SCADA.

## 6.6 Microgrid controller operations

Before operate the microgrid controller, please make sure the Opal-RT model has been loaded, the DNP3 communication IP address is correct on both controller and model side, and the controller initialization is successful.

After all these steps, please open the Opal-RT-PhaseTwo.lvproj. Then right click the LCs and MGCC to connect them. If the connection is successful, you may first run the LC1\_main and LC2\_main. An alternative way is to run the My Real-Time Application under build specifications. (this may trigger a controller restart.) Then you may run the MGCC\_main\_PhaseTwo under MGCC. In this basic version, a visualization interface is also provided to show the microgrid status. In order to view the visualization interface, you may open the MGCC\_visualization\_phasetwo\_v2. The visualization interface is given as following:

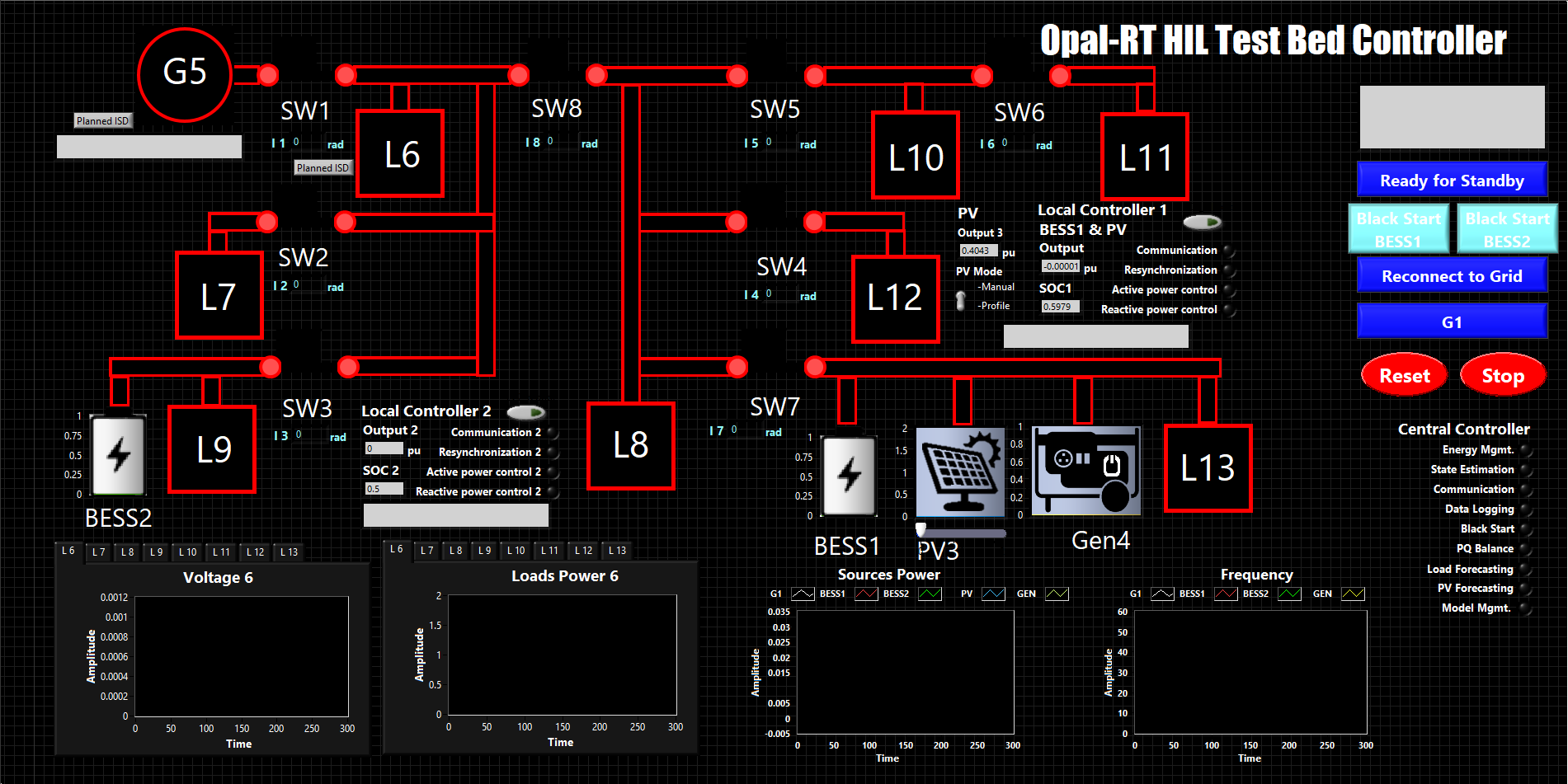


Figure 6‑14 Visualization interface

In the interface, the Black start for two BESSs and resynchronization buttons can be found in the right part. The Planned islanding is on the left side beside of G5 and L6. The basic controller status and load status can also be found on the interface.

This section will be updated in the later versions.

1. **REPORT AND FEEDBACK**

# Report and feedback

This chapter provides a detailed reporting and feedback way to further improve the software of the controller.

## 7.1 Report an error

Currently, the only way to report an error could be utilizing e-mail:

[hyin8@utk.edu](mailto:hyin8@utk.edu)

[yma13@vols.utk.edu](mailto:yma13@vols.utk.edu)

[lzhu12@utk.edu](mailto:lzhu12@utk.edu)

## 7.2 Feedback

We are looking forward to any feedback from users and scholars. Please contact us in the forum or with the following e-mail address:

[hyin8@utk.edu](mailto:hyin8@utk.edu)

[yma13@vols.utk.edu](mailto:yma13@vols.utk.edu)

[lzhu12@utk.edu](mailto:lzhu12@utk.edu)