CHAPTER 6

Implementation of Real Time Simulation of Power Systems with Different Controllers

6.1 Introduction

Simulators have been used extensively in the planning and design of electrical systems for the last three decades, the evolution of simulation tools has been driven by the rapid evolution of computing technologies. As computer technologies have decreased in cost and increased in performance, the capability of simulation tools to solve increasingly complex system in lesser time has improved. In addition, the cost of digital simulators has also steadily decreased, making them available to a larger number of users for a wider variety of applications.

The main concentration of this chapter is to validate the developed controllers in real time environment. First, real-time simulation is described with an overview of the evolution of real-time simulators. After this the working of OPAL RT simulator is explained and then validation process of controllers for multi area system starts. In this chapter, for the real time simulation RT 5142 simulator is used which is manufactured by OPAL-RT, Ontario, Canada. The RT LAB software is installed which is essential for real time simulation along with MATLAB 7.4.0 (R2007a) is installed; the RT LAB software helps the MATLAB to create real time simulation environment. This software provides a channel to the computer and RT simulator for communication. The RT LAB simulator version is 10.2.4. Figure 6-1 shows the OPAL RT 5142 simulator.
6.2 Real time simulator

Real-time simulation, the accuracy of computations not only depends upon precise dynamic representation of the system, but also on the length of time used to
produce results. For a real-time simulation to be valid, the real-time simulator used must accurately produce the internal variables and outputs of the simulation within the same length of time that its physical counterpart would. In fact, the time required to compute the solution at a given time-step must be shorter than the wall-clock duration of the time-step. For a given time-step, any idle-time preceding or following simulator operations is lost; as opposed to accelerated simulation, where idle time is used to compute the equations of the next time-step. In such a case, the simulator waits until the clock ticks to the next time-step. However, if all simulator operations are not achieved within the required fixed time-step, the real-time simulation is considered erroneous. This is commonly known as an “overrun”.

A real-time simulator is performing as expected if the equations and states of the simulated system are solved accurately, with an acceptable resemblance to its physical counterpart, without the occurrence of overruns. RT Lab Simulator has main focus on to electrical networks for real time simulation and these are needed to test controllers for stability of networks.

6.3 Why use real-time simulation

The power system is highly nonlinear in nature. The multi area inter connected power system is highly complex system. The complexity of whole systems with sophisticated controller is too high. Here some benefits are shown for real time simulation.

1. **Save time**
   - Allows test engineers to save time in the testing process. Find problems at an earlier stage in the design process. It’s proceeding to a device design while the actual system is not physically available.

2. **Lowering cost**
   - It reduces enormous cost on testing a new device under real conditions. The real-time simulator is capable to could test many possible configurations without physical modification. It also reduces the total cost over the entire project and system life cycle.
3. **Increasing test functionalities**
   - With the help of RT simulator, test all possible scenarios that could happen in real life in a secure and simulated environment.
   - It has high flexibility by being able to modify all parameters and signals of the test system at a glance.
   - It has automatic test script in order to run tests.

4. **Reduces Risks**
   - It reduces the risk of persons’ lives
   - It also reduces the risks which are associated with plant safety

The real time performance is necessary before going online implementation of these controllers.

6.4 **Evolution of Real-Time simulators**

Simulator technology has evolved from physical/analogue simulators (HVDC simulators & TNAs) for EMT (Electro-Magnetic Theory) and protection & control studies, to hybrid TNA/Analogue/Digital simulators capable of studying EMT behavior [SU 08], to fully digital real-time simulators, as illustrated in Figures 6-2 and 6-3. Physical simulators served their purpose well. However, they were very large, expensive and required highly skilled technical teams to handle the tedious jobs of setting up networks and maintaining extensive inventories of complex equipment. With the development of microprocessor and floating-point DSP technologies, physical simulators have been gradually replaced with fully digital real-time simulators.

DSP-based real-time simulators developed using proprietary technology, and used primarily for HIL (Hardware In Loop) studies, were the first of the new breed of digital simulator to become commercially available [KUF 95]. However, the limitations of using proprietary hardware were recognized quickly, leading to the development of commercial supercomputer-based simulators, such as HYPERSIM from Hydro-Quebec [DO 99], which is no longer commercially available. Attempts have been made by universities and research organizations to develop fully digital real-time simulators using low-cost standard PC
technology, in an effort to eliminate the high costs associated with the use of high-end supercomputers [HOL 08]. Such development was very difficult due to the lack of fast, low-cost inter-computer communication links. However, the advent of low-cost, readily available multi-core processors [RAM 06] (from INTEL and AMD) and related COTS (Commercial Of The Shelf) computer components has directly addressed this issue, clearing the way for the development of much lower cost and easily scalable real-time simulators. In fact, today’s low-cost computer boards equipped with eight processor cores provide greater performance than 24-CPU supercomputers that were available only 10 years ago. The availability of this low-cost, high performance processor technology has also reduced the need to cluster multiple PCs to conduct complex parallel simulation, thereby reducing dependence on sometimes-costly inter-computer communication technology.

COTS-based high-end real-time simulators equipped with multi-core processors have been used in aerospace, robotics, automotive and power electronic system design and testing for a number of years [BEL 07]. Recent advancements in multi-core processor technology means that such simulators are now available for the simulation of EMT expected in large-scale power grids, micro grids, wind farms and power systems installed in all-electric ships and aircraft. These simulators, operating under Windows, LINUX and standard real-time operating systems, have the potential to be compatible with a large number of commercially available power system analysis software tools, such as PSS/E, EMTP-RV and PSCAD, as well as multi-domain software tools such as SIMULINK and DYMOLA. The integration of multi-domain simulation tools with electrical simulators enables the analysis of interactions between electrical, power electronic, mechanical and fluid dynamic systems. The latest trend in real-time simulation consists of exporting simulation models to FPGA [MAT 10]. This approach has many advantages. First, computation time within each time-step is almost independent of system size because of the parallel nature of FPGAs. Second, overruns don’t occur once the model is running and timing constrains are met. Last but most importantly, the simulation time-step may be very short, in the order of 250 nanoseconds. There are still limitations on model size since the number of gates is limited in FPGAs. However, this technique holds promise.
6.5 RT-LAB system architecture

RT-LAB system has basically two parts. One is Host computer and another is RT simulator. Host computer has three basic purposes as

- Edition of Simulink model
- Model compilation with RT-LAB
- User interface

And the RT simulator also has three basic purposes which are

- I/O and real-time model execution
- QNX or Linux OS
- FTP and Telnet communication possible with the Host
These two are connected with TCP/IP as shown in Figure 6-4.

Architecture of RT-LAB simulator has two basic parts. One is Hardware and another is Software. Hardware architecture is given in Appendix and Software architecture is presented here.

6.5.1 Software architecture

The RT-LAB software architecture enhances a user’s comprehension and make him more efficient to work with Opal-RT’s systems. it has three basic parts

A. Meta Controller  
B. RT Lab live view  
C. License
A. Meta controller

The Meta Controller is main application of RT-LAB main application. It acts as a server and it is required to open a new model. It allows a user to open the RT-LAB window to access directly the Monitoring Viewer, the Parameter Control, the Probe Control, the RT-LAB Display and the Dinamo panel. Two more options are available Display the controller application in the taskbar and Enable the Controller Log for debugging purpose.
The Meta Controller is started when the operating system of the host computer starts. It can be accessed in the icon system tray when running. To start it manually, go to Start/Programs/RT-LAB 10.2.4 and select Meta Controller as shown in Figure 6-5.

**B. RT Lab live view**

The RT-LAB window is RT-LAB’s graphical user interface. It allows the user to go through the entire execution process and to access the various functions and operations offered by RT-LAB as shown in Figure 6-6. It also has links to other utilities and advanced features. The Meta Controller application must be running before starting the RT-LAB window.
Figure 6- 6 View of RT Lab window
C. License

Opal-RT’s license enables/disables various RT-LAB features. Some features are automatically included in the license file to match the hardware configuration of the client. A demo license file has an expiration date (not a permanent one) and is, most of the time, linked to only one host computer via the MAC address.

6.6 From Simulink environment to Real-Time environment

In order to use a Simulink model in real-time with Opal-RT’s software and hardware, a user must follow some modeling concepts when building his application.

A. Start from scratch
B. Regroup into subsystems
C. Add the OpComm block(s)
D. Set the real-time parameters
E. Run offline (without RT environment)

A. Start From Scratch

The easiest way to build a Simulink model for any application is to start from scratch right in Simulink. To start with a blank page is a good idea but it is not mandatory.

B. Regroup into subsystems

In a Simulink model that is to be used with RT-LAB, no mathematical content can be found in the top-level of the model. Therefore, subsystems are needed. The user must add a prefix to all the top-level subsystems to allow RT-LAB to manage them. There are 3 types of subsystems. Each subsystem is computed on a different computation node in Opal-RT’s hardware. The three subsystems are

- Master subsystem (SM)
This is the main subsystem and every model must have one SM subsystem. All the computational elements of the model, the mathematical operations, the I/O block, the signal generators, etc.

- **Slave subsystem (SS)**
  This subsystem is needed only if the computational elements must be distributed across multiple nodes. All the computational elements of the model, the mathematical operations, the I/O block, the signal generators, etc.

- **Console subsystem (SC)**
  This is the only subsystem that will be available to the user during execution. It enables the user to interact with the system while it is running.

![Figure 6-7 Communication between the console and the computation node](image)

C. **Add the OpComm block(s)**

RT-LAB uses OpComm blocks to enable and save communication setup information as shown in Figure 6-8. This includes both communication between the console and the computation nodes and communication between the multiple computation nodes in a distributed simulation scenario. All subsystems inputs must first go through an OpComm block before any operations can be done on the signals they are associated with as shown in Figure 6-9.

263
All subsystems inputs must first go through an OpComm block before any operations can be done on the signals they are associated with. OpComm block must be inserted after the subsystems creation.

**In the real-time subsystems (SM or SS)**

- One OpComm receives real-time-synchronized signals from other real-time subsystems
- One OpComm receives asynchronous signals from the console subsystem.

**In the console subsystem (SC)**

- One or more OpComm blocks may be inserted to receive signals from the real-time subsystems. Multiple OpComm blocks define unique “acquisition groups” with their own data acquisition parameters.
Here one OpComm block of master and slave receives signals from consol and one OpComm is inserted to receive signals from the real time subsystems.

**D. Set the real-time parameters**

A real-time model can only run in Fixed-Step mode. Fixed-step solvers solve the model at regular time intervals from the beginning to the end of the simulation. The Fixed-Step size (fundamental sample time) must be chosen carefully regarding the needs of the simulation, the dynamics of the model and must take in account the software/hardware capability.

In a real-time system, we define the Time Step as a predetermined amount of time (ex: $T_s = 10 \mu s$, 1 ms, or 5 ms). Inside this amount of time, the processor has to read input signals, such as sensors, to perform all necessary calculations, such as control algorithms, and to write all outputs, such as control actuators. The term “real-time” is used by several industries to describe time-critical technology. However, the most common usage of “real-time,” and the usage that applies to OPAL-RT, is in reference to embedded systems. Embedded systems are electronic devices designed to interface with the real world and provide control, interaction, and convenience of some form.

In a real-time system, the embedded device is given a predetermined amount of time, such as 1 ms, 5 ms, or 20 ms to read input signals, such as sensors, to perform all necessary calculations, such as control algorithms, and to write all outputs, such as control actuators, and control fuel flow.

**How to define a time step**

For Inputs or Outputs highest frequency sampling should considered. Generally, decreasing the time step increases the accuracy of the results while increasing the time required simulating the system. The rule of thumb is to have around 10 to 20 samples per period for an AC signal. For a 50 Hz signal

\[
\frac{1}{20 \times 50} = 0.001 = 1 \text{ ms}
\]
A typical mechanical application may run around 1ms while a typical electrical system may run around 50us. Here also 1 ms time constant is chosen.

The console subsystem should run up to indefinitely to investigate the responses of systems behavior under varying load condition. In order to do so, the simulation’s stop time must be set to “inf”. With this parameter set, the console will be stopped only when the user decides to reset his model. This is shown in Figure 6-10.

![Configuration parameter window](image)

**Figure 6- 10 Configuration parameter window**

E. Run off-line

Existing model of three area system is checked in off-line mode. This process is important in such a way that Run off-line of model gives the surety of its working in MATLAB.

6.6.1 Execution process

After this RT LAB execution process can be started. The execution process allows the user to run his Simulink model with Opal-RT’s software and hardware. It is based on seven (7) main steps as shown in Figure 6-11.
A. Create an RT-LAB Project
B. Open a model
C. Edit a model in Matlab
D. Compilation / Build
E. Assign nodes
F. Loading
G. Execution / Pause
H. Reset

Figure 6-11 Execution Parameters

A. Create an RT-LAB Project

In the RT-LAB window, selecting File> New > Project allows the user to open a new RT-LAB project as shown in Figure 6-12. Enter a Project Name for the Simulink, System Build & EMTP models that can be opened.
B. Open a model

In the “New RT-LAB Project” window, select an existing model to add to the project. Researchers can also add a model to an existing RT-LAB project by right-clicking on the project’s name and selecting Add>Existing model. To open the model, double-click on the model’s name. A new tab should open to display Overview, Development, Execution settings, etc.
C. **Edit a model**

In the RT-LAB window, right-click on the model listed under the project’s name in order to edit the Simulink model using Matlab. Click on the “Edit” option and the model will automatically open in a new instance of Matlab.

D. **Compilation**

In the RT-LAB window, the “Build” button allows the user to compile his Simulink model when it is highlighted. Alternatively, select Simulation>Build. Use drop-down menu to select only specific steps of the compilation. If for any reason the user wants to stop the compilation while it is launched, select Cancel in the “Building model” console progress tab. The model is separated into individual Simulink models (.mdl files). One .mdl file is created for each top-level subsystem (SM, SC and SS). RT-LAB calls Real-Time Workshop (RTW) from Matlab to generate C code from the Simulink models previously separated using a specific template for each platform. Code for each real-time subsystems (SM and SS) is generated separately (individual calls to RTW). The generated C code is transferred from the host computer to the target via an internal RT-LAB process (OpalD). All required files are automatically transferred by RT-LAB, including files generated by RTW for each subsystem. Once on the target, the compiler takes care of building and linking the files to generate a real-time executable (.exe) for each subsystem (SM and SS). The compiler is proper to each platform. For example, QNX’s compiler is named gcc. After everything is built, RT-LAB automatically transfers back the executable(s) to the host computer. This means that a user can compile his model on Target A and launch it on Target B as all real-time executable files are now.

E. **Assign Nodes**

In the RT-LAB window, the Assignation tab allows the user to link the model’s subsystems to the targets. The list of available nodes can be found in the drop-down bar of the Assigned node field. Use the drop-down menu to associate a subsystem to a node. To make sure no unwanted processes and files are still on the target to be assigned, use the
“Clean target” button. Right-clicking on the desired target and selecting Tools> Diagnostic will provide you very useful information about the target selected.

F. Load

In the RT-LAB window, the “Load” button allows the user to load the Simulink model on the target(s). All executables created during compilation are launched on the target(s) in “Pause” mode. This is the last step before executing the model. The console will open and be ready for execution. There is one log window for each real-time subsystem (SM or SS).

G. Execution

Once loaded, the user has two options to execute the model or to reset the model. In the RT-LAB window, the “Execute” button allows the user to execute his Simulink model. During execution, the user can change the controls found in the console and witness changes displayed in scopes, displays, etc. Researchers can pause or reset the model at any time.

H. Reset

When the user wants to stop definitely the model from running, he must use the reset button found in the RT-LAB window. This will stop the acquisition (software & hardware) and the console will close.

6.7 Simulation in Real Time environment

In this section of this chapter, three area systems with and without nonlinearity with different type of controllers in RT Lab Environment are simulated. This section deals with verification of performances of different controllers which are concluded in previous chapters. In the first part of this part, performances of conventional PI controller, fuzzy controller and polar fuzzy controller are checked and compared for the stability of three area linear and nonlinear systems. After this simulation APFC and PFC are verified and compared for the stability of three area linear and nonlinear systems.
6.7.1 Load frequency control of three area nonlinear systems with different type of controllers in RT Lab environment

Development process of models in RT LAB environment was shown in Figures previously. Based on that models are developed. Here one master and two slaves and one console are grouped in Figure 6-13. PI controlled three area hydro-nuclear-thermal system is taken as Master Subsystem, fuzzy controlled three area hydro-nuclear-thermal system is taken as Slave1 Subsystem and Polar Fuzzy controlled three area hydro-nuclear-thermal system is considered as Slave2 Subsystem as shown in Figure 6-14. Figure 6-15 shows the console subsystem before execution. Here one slider load is considered for variation in load. The range of load variation 1%. Figure 6-16 shows the console subsystem after execution. There is one opcom block is attached in each subsystems which helps to communicate. The simulated results are shown in Figures 6-17 to 6-19.
Figure 6- 13 Regrouped Three area system with different controllers in RT Lab Environment
Figure 6-14 Three area sub system with PI controller as Master, Fuzzy controller as Slave 1 and PFC as Slave 2 Subsystem in RT Lab Environment
Figure 6-15 Console of three area sub system in RT Lab Environment before build process of RT Lab
Figure 6-16 Console of three area sub system in RT Lab Environment after execute process of RT Lab
Figure 6-17 Real Time simulated response result of thermal system of three area system when 1% disturbance in thermal system in RT Lab environment
Figure 6-18 Real Time Response of nuclear system of three area system when 1% disturbance in thermal system in RT Lab environment
Figure 6-19 Real Time Response of hydro system of three area system when 1% disturbance in thermal system in RT Lab environment
From Figures 6-17 to 6-19, it is clear that polar fuzzy controller has superior performance over other controllers in real time approach. After the successful testing of PFC in real time environment for linear three area system it can be tested also for three area nonlinear system. From this testing superiority of PFC can be proved.

6.7.2 Three area nonlinear systems with different type of controllers in RT Lab environment

As mentioned above, three area systems with nonlineariety and with different type of controllers in RT Lab Environment are simulated for testing the different controllers performance. Development of systems is shown in Figures 6-20. Here one master and two slaves and one console are grouped. PI controlled three area hydro-nuclear-thermal system is taken as Master Subsystem, fuzzy controlled three area hydro-nuclear-thermal system is taken as Slave1 Subsystem and Polar Fuzzy controlled three area hydro-nuclear-thermal system is considered as Slave2 Subsystem as shown in Figure 6-21. Here one slider load is consider for variation in load. The range of load variation 0 to 1 %. The output could found through Console Subsystem after execution process same as in previous section. The simulated results are shown in Figures 6-22 to 6-24.
Figure 6-20 Three area nonlinear systems with different type of controllers in RT Lab Environment
Figure 6-21 Three area sub system with PI controller as Master, Fuzzy controller as Slave 1 and PFC as Slave 2 Subsystem in RT Lab Environment
Figure 6-22 Real Time Response of thermal system of three area nonlinear system when disturbance in thermal system in RT Lab environment
Figure 6-23 Real Time Response of nuclear system of three area nonlinear system when disturbance in thermal system in RT Lab environment
Figure 6-24 Real Time Response of hydro system of three area nonlinear system when disturbance in thermal system in RT Lab environment
From Figures 6-22 to 6-24, it is clear that polar fuzzy controller has superior performance over other controllers also in real time approach. From the testing superiority of PFC is proved in linear and nonlinear systems both.

In the next section performance Adaptive PFC will be compared with PFC in RT LAB environment. It is concluded in chapter 5 that performance of adaptive PFC is superior to the PFC in varying load conditions. This process helps in automation of Power Generation Industry.

**6.7.3 Real Time testing of Adaptive PFC and PFC in three area linear system**

This section deals with verification of performances of Adaptive polar fuzzy controller and polar fuzzy controller. In this part, performances of Adaptive polar fuzzy controller and polar fuzzy controller are checked and compared for the stability of three area linear system. After this simulation APFC and PFC are verified and compared for the stability of three area nonlinear system. These are verified also for one and multi area disturbances.

**6.7.3.1 Single area disturbance**

Here one master and one slave and one console are grouped in Figure 6-25. APFC controlled three area hydro-nuclear-thermal linear system is considered as master subsystem and PFC controlled three area hydro-nuclear-thermal system is considered as slave subsystem as shown in Figure 6-26. Figure 6-27 shows the console subsystem before execution. Here one slider load is considered for variation in load. The range of load variation 0 to 1%. Figure 6-28 shows the console subsystem after execution. There is one opcom block is attached in each subsystem which helps to communicate. The simulated results are shown in Figures 6-29 to 6-31.
Figure 6-25 Three area linear systems with APFC and PFC in RT Lab Environment
Figure 6-26 Subsystem of three area with APFC controller as Master Subsystem and PFC as Slave Subsystem in RT Lab Environment
Figure 6- 27 Console sub-system before Execution in RT Lab Environment

Figure 6- 28 Console sub-system after Execution in RT Lab Environment
Figure 6-29 Real Time Response of hydro area of a three area linear system with APFC and PFC when disturbance in thermal system in RT Lab environment
Figure 6-30 Real Time Response of nuclear area of a three area linear system with APFC and PFC when 1% disturbance in thermal system in RT Lab environment
Figure 6-31 Real Time Response of thermal area of a three area linear system with APFC and PFC when 1% disturbance in thermal system in RT Lab environment
From Figures 6-29 to 6-31, it is shown that Adaptive Polar fuzzy controller has superior performances over PFC controller in real time approach. After the successful testing of APFC in real time environment for linear three area system when disturbance in thermal area, it is also tested for linear three area system when disturbance in thermal and hydro area. From this testing superiority of APFC over PFC can be proved in linear system.

6.7.3.2 Two area disturbances

The APFC is employed for multi-area system when disturbances applied on more than one area. For this it is assumed that 0 to 1% varying load disturbances is applied over nuclear and thermal area under different load conditions. Performances of APFC and PFC are compared in 6-32 to 6-34.
Figure 6-32 Real Time Response of hydro area of a three area linear system with APFC and PFC under different disturbances in thermal and nuclear systems
Figure 6- 33 Real Time Response of nuclear area of a three area linear system with APFC and PFC under different disturbances in thermal and nuclear systems
Figure 6-34 Real Time Response of thermal area of a three area linear system with APFC and PFC under different disturbances in thermal and nuclear systems
6.7.4  **Real Time testing of Adaptive PFC and PFC in three area non-linear system**

In this section, comparative performances of Adaptive polar fuzzy controller and polar fuzzy controller are shown in nonlinear environment. In this part, performances of Adaptive polar fuzzy controller and polar fuzzy controller are checked and compared for the stability of three area nonlinear system. These are verified also for one and multi area disturbances.

**6.7.4.1 One area disturbance**

Its development process in real time environment is shown in Figure 6-35. It is same as linear system. Here one master and one slave and one console are grouped. APFC controlled three area hydro-nuclear-thermal nonlinear system is considered as master subsystem and PFC controlled three area hydro-nuclear-thermal nonlinear system is considered as slave subsystem as shown in Figure 6-36. Output is taken from Console Subsystem as shown in previous section. Here one slider load is consider for variation in load. The range of load variation 0 to 1 %. There is one opcom block is attached in each subsystems which helps to communicate. The simulated results are shown in Figures 6-37 to 6-39.

From Figures 6-37 to 6-39, it is shown that Adaptive Polar fuzzy controller has superior performance over PFC controller in real time approach. After the successful testing of APFC in real time environment for nonlinear three area system when disturbance in thermal area, it can be tested also for linear three area system when disturbance in thermal and hydro area.

**6.7.4.2 Two area disturbances**

The performances of APFC and PFC are verified under multi-area disturbances for three area nonlinear system. For this it is assumed 1% varying load disturbances are applied in nuclear area and thermal area under different load conditions. Performances of APFC and PFC are compared in 6-40 to 6-42.
Figure 6-35 Three area nonlinear systems with APFC and PFC in RT Lab Environment
Figure 6-36 Subsystem of three area with APFC controller as Master Subsystem and PFC as Slave Subsystem in RT Lab Environment
Figure 6-37 Real Time Response of hydro area of a three area nonlinear system with APFC and PFC when 1% disturbance in thermal system in RT Lab environment
Figure 6-38 Real Time Response of nuclear area of a three area nonlinear system with APFC and PFC when disturbance in thermal system in RT Lab environment
Figure 6-39 Real Time Response of thermal area of a three area nonlinear system with APFC and PFC when disturbance in thermal system in RT Lab environment.
Figure 6- 40 Real Time Response of hydro area of a three area nonlinear system with APFC and PFC under different disturbances in thermal and nuclear systems
Figure 6-41 Real Time Response of nuclear area of a three area nonlinear system with APFC and PFC under different disturbances in thermal and nuclear systems
Figure 6- 42 Real Time Response of thermal area of a three area nonlinear system with APFC and PFC under different disturbances in thermal and nuclear systems
From 6-40 to 6-42 it is proved that APFC performed better in varying multi area disturbances.

6.8 Conclusions

In this chapter it is concluded that Adaptive Polar fuzzy controller shows its superiority over PFC, fuzzy and PI controllers. Its superiority is checked in real time environment. These controllers are tested for three area hydro-nuclear-thermal linear and nonlinear systems with one and multi are disturbances, with the help of OPAL-RT real time simulator OP 5142 v 10.2.4. It is found that Adaptive Polar fuzzy controller shows best performance among aforesaid controllers in terms of settling time, less frequency dip and minimum oscillations in real time environment. Now this Adaptive Polar fuzzy controller can be implemented for online approach.

******