Development of A Digital Excitation System for Synchronous Generator on RTAI-Linux Platform

Deepak. M and K. N. Shubhanga

Department of Electrical Engineering, NITK, Surathkal, 575025, INDIA, E-mail: knsa1234@yahoo.com

***Abstract*—In this paper realization of a digital excitation system for synchronous generator is discussed. Such a real-time task is implemented using Real-Time Application Interface on Linux platform which is an open source effort. In this work a static excitation system is considered which involved the development of additional hardware circuits such as fully-controlled thyristor bridge-rectifier, the associated firing circuit module and data acquisition systems. In addition to normally discussed proportional controller, a proportional-integral controller is also realized to understand its generator voltage regulation performance under different operating conditions. Case studies are presented related to both standalone and parallel operation conditions of the generator without taking up complex controller-tuning procedures. Such a setup is found to augment the generally employed simulation-based power system dynamics analysis.**

***Index Terms*—Synchronous generator, excitation controller, real-time controllers, PI-controller, Linux**

I. INTRODUCTION

The excitation system is an integral part of a synchronous machine as it supplies DC power to the field winding. It performs regulate, control and protective functions essential to the satisfactory performance of power system by controlling the field voltage and thereby field current [1]. An optimally-tuned excitation system not only carries out steady state functions like voltage regulation, reactive power control, but also offers benefits in overall dynamic performance of the system during the conditions of small and large disturbances [2]. In the older generator systems the desired exciter response was obtained using analog system such as potentiometers, lead-lag analog circuits and amplifier blocks [3]. However, with the development of digital technology the excitation systems have seen digital versions as it is simpler to implement and tune controllers [4]-[6]. Many of newly introduced the IEEE specified exciter models [7] incorporate PID schemes which are straightforward to realize on digital platforms as they are not constrained by offset-voltage related problems. From the literature it is clear that dedicated microcontrollers, digital signal processors are the preferred digital platforms for such implementations. These platforms even permit realization of nonlinear models and controllers such as neural network, fuzzy logic and adaptive controllers [8].

In this connection, Real-Time Application Interface (RTAI) on Linux [9]-[12] offers a platform to implement real-time systems on a general purpose PC. Many applications have been presented in [13]-[17] employing RTAI-Linux demonstrating its usability. In this paper the development of a static excitation system for a synchronous generator using RTAI-Linux is presented. The paper is structured as follows: RTAI and hardware setup details are presented in section II. Proportional- and integral-controller implementation is described in section III. Procedure employed for tuning the controllers is discussed in section IV. Controller performance during parallel operation of generators is presented in section V along with some time-domain simulation results.

II. SOFTWARE AND HARDWARE SETUP

A. Software - RTAI

RTAI is a real-time extension for the Linux kernel which allows to write applications with strict timing constraints for Linux. Like Linux, the RTAI is an open source or community effort. RTAI offers the same services of the Linux kernel core, adding the features of an industrial real-time OS. It consists of an interrupt dispatcher to trap the peripheral interrupts and if necessary re-routes them to Linux. It is not an intrusive modification of the kernel; it uses the concept of Hardware Abstraction Layer (HAL) to get information from Linux and to trap some fundamental functions. RTAI considers Linux as a background task running when no real time activity occurs. Thus RTAI just provides real-time functions to Linux enabling a PC to be used for real-time system development. The procedure of installing RTAI is given in [9], [17].

B. Hardware Setup

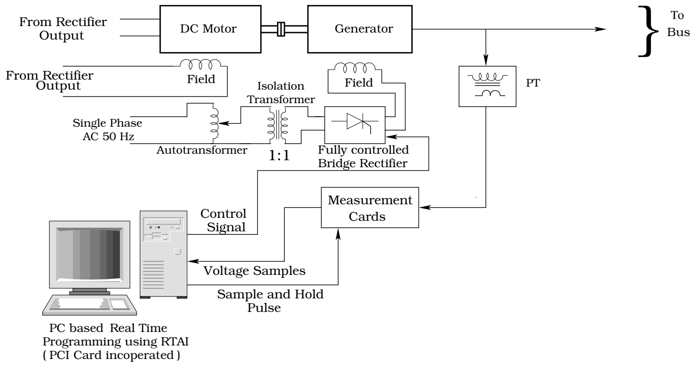
The whole set up of interfacing a digital excitation controller to the field of a synchronous machine is shown in Fig. 1. Here DC motor driven synchronous machine is used and the terminal voltage of the machine is measured using step down transformers followed by signal conditioning cards. Advantech PCI-1710HG is used to acquire data and exchange information with the external devices [18].

Excitation system consists of a fully-controlled single-phase thyristor-based bridge rectifier with the associated firing circuit. The firing angle is controlled by varying the control voltage through the analog (DAC) output of the PCI card as guided by the real-time task on RTAI. This scheme basically uses a inverse-cosine approach for maintaining linearity in the controller [17].

III. IMPLEMENTATION OF CONTROLLER IN RTAI

The controller block schematic implemented as a real-time task on RTAI program is as shown in Fig. 2. The figure

978-1-4673-2636-0/12/$31.00 ©2012 IEEE



Alternator:$1kVA, 230V, 3-\phi, 50Hz, Yconnected, 1500rpm.$

Fig. 1. Block diagram of data acquisition from synchronous machine with Automatic Voltage Regulator

shows various components such as comparator, regulator and the plant. The comparator generates the error signal using the reference value, V_{ref} (which set as $230V$) and the measured RMS voltage (Line-to-line).

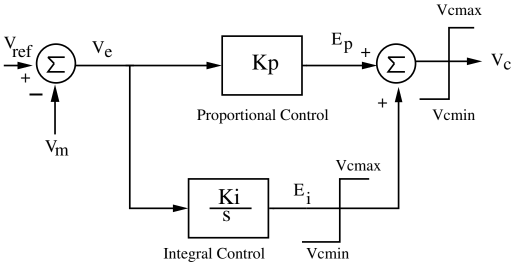


Fig. 2. Proportional-Integral Control System

In the following lines implementation of each of the controllers components/functions are detailed:

A. Voltage Measurement

The RMS value of the generator line-to-line voltage is measured from the instantaneous samples of 3-phase voltages as follows:

$$V_m = \sqrt{(v_a^2 + v_b^2 + v_c^2)} \tag{1}$$

The measured voltage signal smoothened by passing it through a first order filter as given below.

$$\frac{y(s)}{u(s)} = \frac{1}{1+sT} \tag{2}$$

By employing Forward Euler method, the above transfer function is discretized as

$$y_{n+1} = y_n + \frac{\Delta T}{T}(u_n - y_n) \tag{3}$$

where ΔT is the time step of integration.

B. Regulator Implementation

In the regulator function two types controllers are employed: 1) proportional (P) controller and 2) proportional-integral (PI) controller.

1) P - controller: Here, a simple proportional controller is implemented such that the device produces an output signal, $E_p(t)$, which is proportional to the input signal $V_e(t)$ as given by the following rule:

$$E_p(t) = K_p V_e(t) \tag{4}$$

where,$K_p =$ Proportional gain.
Here, the control voltage $V_c = E_p(t)$ and it is known that the choice K_p decides not only the response speed of the controller, but also the steady-state error in the terminal voltage. In this case integral function is not used which is obtained by setting K_i = 0 in Fig. 2.

2) PI - controller: In the PI regulator the P controller is implemented as given in (4) and the integral function is realized employing the Forward Euler method as follows:

$$y_{n+1} = y_n + \Delta T(K_i V_e(t)) \tag{5}$$

with $E_i(t) = y_{n+1}$
where ΔT is the time step of integration and K_i is the Integral gain.
Here, the control voltage $V_c = E_p(t) + E_i(t)$. Also note that the final control voltage is obtained by the applying the appropriate limits on V_c. It is to be noted that the choice K_i mainly decides the speed of the controller (for a given k_p) and drives the steady-state error to zero eventually.

C. RTAI Programming Details

The real-time task performs two functions: sampling of input 3-phase voltages and processing of data to realize control action as per the controller configuration to regulate voltage. This real-time task is assigned the highest priority. Further, the ADC of the PCI card is multiplexed and hence only one channel is scanned at a time. The individual channels data is sampled at every 250μs. For this purpose three signal-conditioning cards are used with associated sample-and-hold circuits. [9] A partial C-code is presented below to demonstrates the implementation of sample-and-hold pulse generation, data reading and conversion to actuals:

```
  while(c<3)
 {
if (counter == CN_MAX)
 {counter=0;
  }
if (counter == CN_0)
{       //**Sample and Hold pulse**//
outb(0x01,base_add+16);
rt_busy_sleep(T_ON);
outb(0x00,base_add+16);
 }
outw(0,base_add +0x0000);
        //** Start Conversion**//
rt_busy_sleep(T_C);
inp_data =  inw(base_add);
        //**To get the input from
```

```
            the register**//
 inp_data1=  inp_data & 0x0fff;
 channel_no= (inp_data & 0xf000)>>12;
         //**Data conversion to actuals**//
 inp_voltage=((inp_data1*20.0/4096.0)-10.0);
  if (channel_no==0)
  {
  va= HALL_RATIOV * inp_voltage;
  }
 else if (channel_no==1)
......
```

IV. TUNING OF CONTROLLERS

In this case studies are presented to tune the P - and PI - controllers by employing trial and error approach. Rigorous analytical designs procedures are not followed as it involves modelling of the complete system including the generator, and requires the data pertaining to the operating condition. Test results are presented for standalone loading conditions.

A. Tuning of Proportional-controller Gain

In this setup the generator is connected to a local inductive load of rating 230 V, 2 A. The controller gain, K_p, is varied for various values ranging from $K_p = 10$ to $K_p = 40$. Fig.3 depicts the terminal voltage of the machine for different values of K_p. The figure also shows the machine voltage without any controller where the voltage dips to nearly 175 V. However, when the controller is enabled the voltage is well above 210 V. The steady-state error with K_p = 20 is relatively more than that with K_p = 30. It can also be seen that when K_p is chosen above 30 the steady-state error does not change appreciably. Hence, K_p is chosen to be equal to 30 for all the remaining tests.

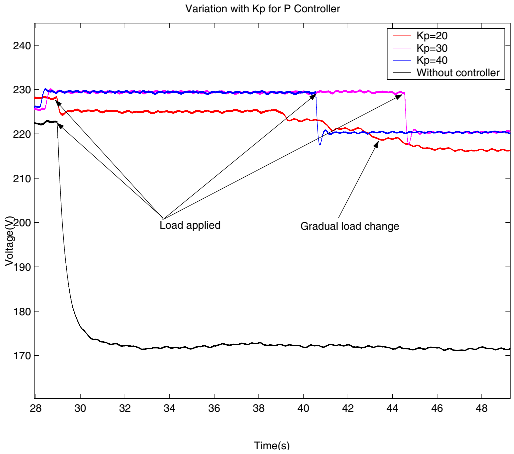


Fig. 3. Response with various values of K_p for inductive load.

B. Tuning of Integral-controller Gain

Here also the generator is connected to a local inductive load of rating 230 V, 2 A. The proportional-controller gain, K_p, is set at 30 and only K_i is varied upto 10 to determine the suitable value by trail and error. No analytical procedure is employed to finalize the gain values. Fig. 4 demonstrates the plots of the machine terminal voltage for different gains. It can be seen for the figure that following the application of the load under steady-state, the terminal voltage is very close to 230 V, i.e., the reference value set for the controller. From the plots it is also clear that for K_i=1, the response is poor, i.e., it takes more time to settle. As K_i is increased towards 10 the response improves, it reaches steady-state quickly. Based on this observation, K_i is simply chosen as 10 without applying any other criterion.

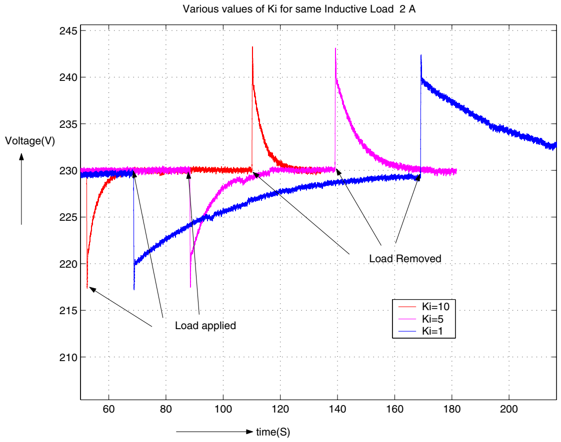


Fig. 4. Response with various values of K_i for inductive load.

C. Comparison of P- and PI-controllers

Here, performance of P- and PI- controllers is compared for standalone resistive and inductive loads each rated at 230 V 2 A. For P-controller, K_p is set to 30 and for PI-controller, K_p= 30 and K_i is set to 10. The results are shown in Figs. 5 and 6. From the figures the following points are evident:

- With inductive load the terminal voltage dip is relative more than that with resistive loads when no controllers are used.
- With PI- controller, both in resistive and inductive cases the steady-state error is zero. Whereas with P- controller there exists an error under steady-state.
- the terminal voltage overshoots beyond the rated value when the load is thrown off when PI-controller is used. This value is relatively more in the case of inductive loads than that with resistive loads. Such overshoots are very small in the case of P-controllers. .
- The speed of response is smaller with P-controller than that with PI-controller.

V. CONTROLLER PERFORMANCE DURING PARALLEL OPERATION

In this section machine with the controller is synchronized with other machine of similar capacity and in another case it is made to work in parallel with mains. The performance of the both P- and PI-controllers are compared.

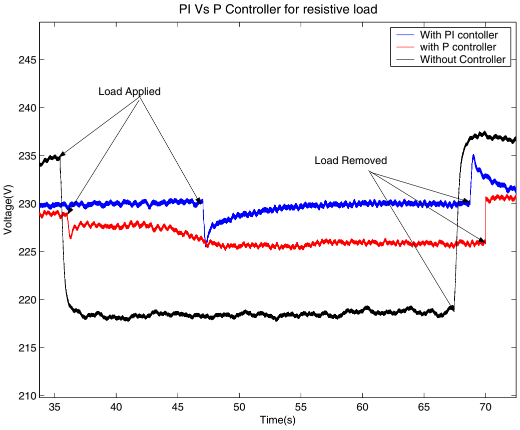


Fig. 5. PI- and P-controllers for resistive load.

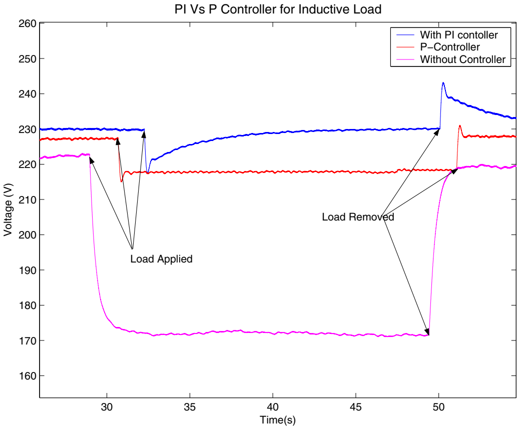


Fig. 6. PI- and P-controllers for inductive load.

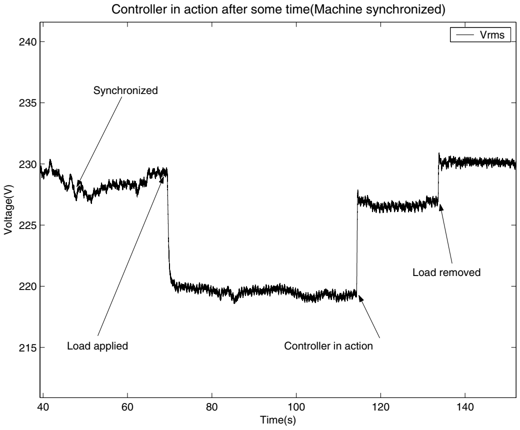


Fig. 7. Parallel operation with another machine: P-controller.

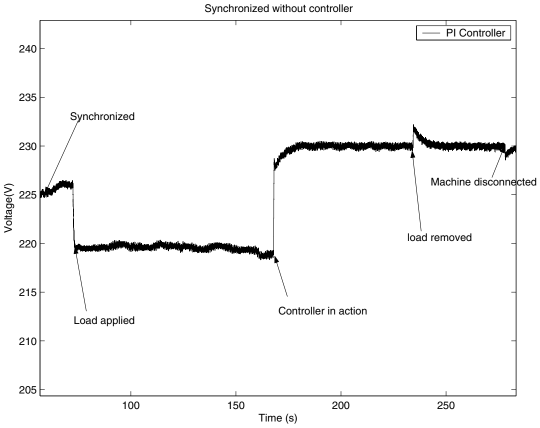


Fig. 8. Parallel operation with another machine: PI-controller.

A. Parallel Operation with Other Machine

In this case machine with the controller is synchronized with other machine of similar capacity by dark lamp method. Here, dark lamp method of synchronization is employed as it is easy to identify the possible minimum angle between the corresponding phases though it is not accurate. The other machine is made to work with a constant excitation. Fig. 7 depicts the terminal voltage when P-controller is enabled for the generator. The figure shows various events that are carried out following the synchronization. At 70 s the generator is made to deliver a real power of 400 W to other machine. At approximately 110 s the P-controller is activated. The control action of the regulator can be clearly seen where it improves the terminal voltage from 220 V to 227 V. At 136 s the load is gradually reduced and the machine is tripped finally.

Fig. 8 indicates the performance with PI-controller. Similar events are followed as in the previous case. From the figure it can be seen that with the PI-controller in action, the terminal voltage is brought back to 230 V with load. However, once the load is reduced there is a small overshoot in the terminal voltage as we observed in the standalone loading case.

B. Parallel Operation with Mains

In this section machine with the controller is synchronized to the mains by dark lamp method. The controller is enabled on the generator while performing the synchronization unlike in the previous case. Having synchronized, the generators is made to deliver a real power of 400 W to the mains. When the machine running under steady-state with this load the generator is tripped. The performance of the both P- and PI-controllers are compared under this condition. Figs. 9 and 10 show the performance of the machine for P- and PI- controllers respectively.

From the figures the following observations can be made;

1) After synchronization the terminal voltage of the machine is higher than the reference value, i.e., 230 V. With P-controller the control action is not successful in regulating the voltage. Whereas, PI-controller corrects the terminal voltage of the machine and holds it close to 230 V even when the load is increased.
2) When the machine is disconnected with the load both controllers offer a dip in the terminal voltage. This dip is relatively more in the case of PI-controller.

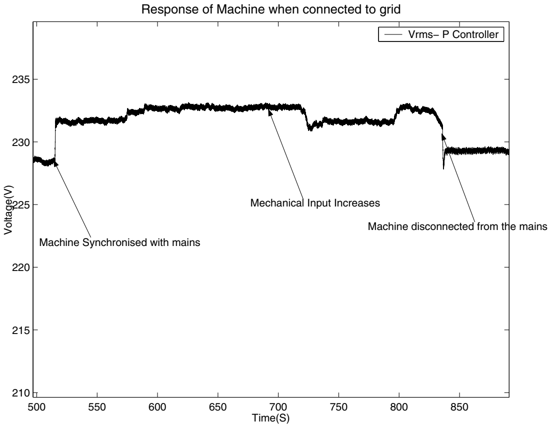


Fig. 9. Machine synchronized to mains with P-controller.

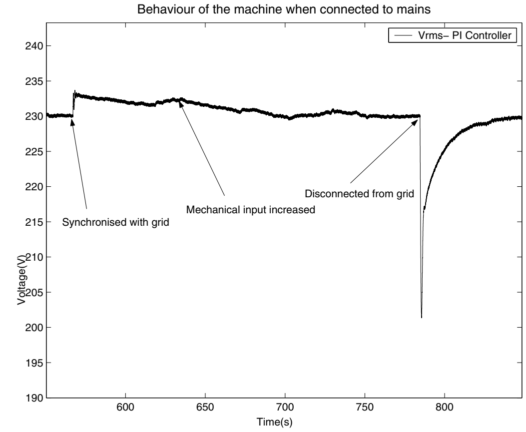


Fig. 10. Machine synchronized to mains with PI-controller.

To understand the performance of the controllers leading to a dip in the voltage following a load thrown off in the synchronization to mains -case, a time-domain, off-line simulation is carried with a simple single-machine connected to infinite bus (SMIB) system (see Fig. 11) employing typical parameters for the generator and the controllers [2].

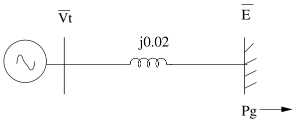


Fig. 11. SMIB system

To mimic the lab condition the line reactance is chosen to be very small (0.02 pu), and the infinite bus voltage is set slightly higher than the generator bus voltage at the time of synchronization. Once the steady-state is reached the generator is tripped with a load of 0.5 pu. Under this condition the plots of the terminal voltage and the E_{fd} are shown in Figs. 12 and 13. From Fig. 12, it can be seen that the dip in the terminal voltage with PI-controller is relatively more than that with P-controller and from Fig. 13, it can be noted that the E_{fd} prior to the occurrence of the disturbance is slightly lower for PI-controller than that for P-controller. This level of E_{fd} and its slow response appear to cause more dip in the terminal voltage when load is thrown off with PI-controllers.

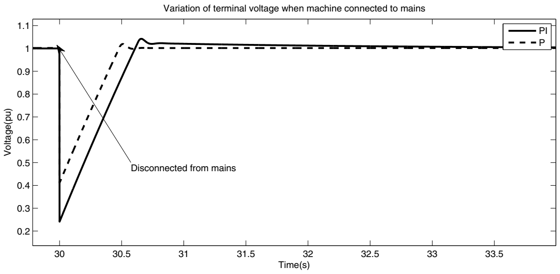


Fig. 12. Terminal voltage when machine disconnected from mains (simulation).

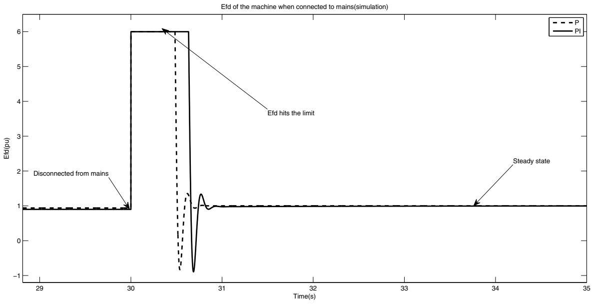


Fig. 13. The plot of E_{fd} (simulation).

A photo of the partial experimental setup in the Power System Lab, EE Dept., NITK Surathkal, India, is shown in Fig. 14. The panel arrangement for interfacing the MG-set and the metering system is depicted in Fig. 15.

Fig. 14. Experimental setup in the lab.

Fig. 15. Panel arrangement for MG-set.

VI. CONCLUSION

In this paper an attempt is made to develop a digital excitation system supported by real-time systems built on RTAI-Linux platform. Except the hardware circuitry, the implementation of the real-time task is straightforward as it involves simple programming in the Linux environment with extensive open source support. Though analytical tuning procedures have not been followed, the realized P- and PI- excitation controllers clearly validate our understanding about their performances in a lab environment. With the availability of system data including the generator and the operating condition details a rigorous design can be taken up at any point of time. Such setups can be easily extended to handle auxiliary controllers such as power system stabilizer with minimal programming and hardware realization efforts.

REFERENCES

[1] P. Kundur, *Power System Stability and Control*, McGraw-Hill Inc., New York, 1994.

[2] K.R. Padiyar, *Power System Dynamics - Stability and Control*, BS Publications, Hyderabad, India, 2002.

[3] IEEE Std 421.5-1992, *IEEE Recommended Practice for Excitation Systems Model for Power System Stability Studies*.

[4] C. A. Morse, R. F. Martinez, C. R. Mummert, I. A. Gibbs, E. C. Prather, ”New Multiprocessor Digital Excitation System”, Power Engineering Society Summer Meeting IEEE, Vol 1, pp. 643-649, 2000.

[5] Richard C. Schaefer, Kiyong Kim, “Tuning a PID controller for a digital excitation control system”, *IEEE Transactions on Industry Applications*, Vol. 41, No.2, pp. 485-492, March-April, 2005.

[6] J. B.Micheal, “Excitation Systems: The Current State of Art”, IEEE Power Engineering Society Summer Meeting IEEE, pp. 18-22, 2006.

[7] IEEE Std. 421-5-2005, *IEEE Recommended Practice for Excitation System Models for Power System Stability studies*.

[8] Jerkovic V, Miklosovic K and Spoljric Z, “Excitation system models of Synchronous generator”, Proceeding SIP-2010, Chengdu, China, 2010.

[9] Real Time Application Interface. [online]. Available: http://www.rtai.org

[10] Doug Abbot, *Linux for Embedded and Real-Time application*, Newnwess Publications.

[11] Pasi Sarolahti, *Real-Time Application interface*. University of Helsinki, Department of Computer Science, 2001.

[12] Aeolean Inc.,“Introduction to Linux for Real-Time Control”, National Institute of Standards and Technology, Version 2.0.0,2002.

[13] Lorenzo Dozio, Paolo Mantegazza, *Linux Real Time Application Interface (RTAI) in low cost high performance motion control*, DIAPM, Milano, 2003.

[14] R. S. Guerra, H. J. Goncalves Junior, W. F. Lages, *A Digital PID Controller Using RTAI*, Federal University of Rio Grande do Sul, Electrical Engineering Department, 2005.

[15] Chetana Savithri, “Real Time Control of Power Electronics Controllers in a Scaled Power System”, Master’s thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, July 2008.

[16] Danda Surendra, “Development of Real-time Systems Using RTAI-Linux For Power System Applications”, Master’s thesis, Department of Electrical Engineering, National Institute of Technology Karnataka,Surathkal, July 2011.

[17] D. Surendra , K. N. Shubhanga, *Development of A Power System Laboratory Supported By Real-time Systems*, International Conference on Power and Energy Systems (ICPS)-2011

[18] *PCI-1710 Series User’s Manual,* Advantech. [Online]. Available: http://www.advantech.com/support.