PID control of DC motor using Real-time Automation test platform in Linux environment

Chirag Panchal
Department of Instrumentation & Control Engineering
L.D. College of Engineering
Ahmedabad, India.

Prof. V.P.Patel
Head of Department (I.C)
Department of Instrumentation & Control Engineering
L.D. College of Engineering
Ahmedabad, India

Abstract

this paper is concerned with Control of DC motor using real time test platform based on Linux operating system, free open source simulation & control software SCILAB/SCICOS, interfacing driver code COMEDI. The Real-time Application Interface (RTAI) freeware is used to create hard real time system clock. That make Linux kernel full preemptive and deterministic. The main advantage of this type platform is that all software or codes are freely available from internet website. Using this Automation test platform Real time speed control of DC motor is done with Advantech PCI1711 DAQ card.

Keywords: PID, Comedi, RTAI, Scilab/scicos, Linux

I. INTRODUCTION

A real-time system is an integral part of any modern control and monitoring application systems. Normally, real-time control tasks are accomplished by employing dedicated processing units such as digital signal processor or advance micro controller unit [21]. HIL (Hardware in Loop) /RCP (Rapid control prototyping) is a cost and time effective approach to test controllers/components/subsystems in a system context. HIL technique is used to develop and test the Real-Time embedded control system. RCP is a process that used to quickly test and iterate system control strategies. RCP requires two main component one is Component Aided Control System Design (CACSD) and other is a dedicated hardware with a hard Real-Time operating environment [20]. RCP environment is mostly based on simulation software MATLAB/Simulink/Real-Time Workshop CACSD, dSPACE and Lab-VIEW. The main disadvantage of this solution is the cost of the needed software at academic level [2] [16] [17].

There are several kinds of real-time add-ons for free open source Linux in the field of real-time automation by PCs. One of the most widely known is RTAI (Real-Time Application Interface). RTL (Real time Linux) has developed in two ways: *(1) Well supported Commercial version called RT-Linux PRO (2) Non-Commercial version called RT-Linux [10]*. Free is based on RTL PRO. The free version of RTL may be considered as an independent product because the programing environment is quite different [1] [4]. An alternative is to obtain a real-time kernel and add it to a commercial RT-Linux installation by ourselves. It is possible to develop real time control loop in open source mathematical and simulation software SCILAB and Simulink tool SCICOS via COMEDI add-ons for RTL. This environment allows to quickly crating real-time controller for real plants by generating and computing the full control application directly from the Scicos scheme [11] [12].

The most obvious advantage of the designed this type Real-time platform is that all software or codes are free and available in web.

In order to get real-time platform a standard kernel must be configured in a Linux base and before this configuration it will include the patching of Hardware Abstraction Layer (HAL) or Adaptive Domain Environment for Operating System (ADEOS) with kernel. After patching and configuration the kernel, installation of the RTAI package must be carried out including rtai-lab and comedi. After this whole process, a set of kernel module are created in the user specified directory. Loading these modules, the real-time functionality is obtained [15].

The paper is structured as follow: process for developing automation test platform is described in section II. Hardware and software specification for real time PID control of DC motor is described in section III. In Section IV, testing results of automation test platform using real time PID control of DC motor. Section V represents the Conclusion.

II. DEVELOPMENT OF AUTOMATION TEST PLATFORM

Installation and development of open source Automation test platform using Real-time Linux Environment is given below section.

A. Installation process

(1) Install Operating system Ubuntu 10.04 LTS in PC.

(2) A real-time task in RTAI is implemented as a kernel module which is loaded into the kernel after the required RTAI core modules have been loaded. This architecture yields a simple and easily maintainable system that allows dynamic insertion/ removal of the desired real-time capabilities and tasks. The steps followed to install RTAI are briefed below [5]:

- Select the version of RTAI, the Linux kernel version to use, and a Linux distribution that uses that kernel. One can choose them in any order, but it is important to end up using a consistent set. For example, with Ubuntu Linux distribution, to install RTAI 2.6., first need to install Ubuntu 10.04 distribution which has Linux Kernel version 2.6.0. A lower version of kernel 2.6.32 is chosen for which the RTAI patch is available [12].
- Install the Linux distribution.
- Download a clean version of the Linux kernel from the Linux distribution sites.
- Download the tar file for the RTAI release.
- Unpacking and uncompressing the files.
- Using the patch file in the RTAI release that corresponds to the kernel source you have downloaded, patch the kernel source to incorporate the RTAI modifications.
- Generate a configuration file that suits the machine you are going to run on.
- Build and install the Linux kernel.
- Configure, build and install RTAI.
- Check for consistency of the installed files.

(3) Configuring the kernel for real time applications.
(4) Compilation and Installation of the newly configured kernel.
(5) Updating of the boot loader to access newly installed kernel.
(6) Mesa and EFLTK installation
(7) Installation of COMEDI and COMEDI-LIB [18].
(8) Configuration, compilation and Installation of RTAI.
(9) Installation of Scilab & RTAI add-on to it [7] [12].
(10) Loading RTAI, COMEDI and DAQ modules [11].

B. Automation test platform

Step to develop Real-Time PID control for DC motor in Scicoslab.

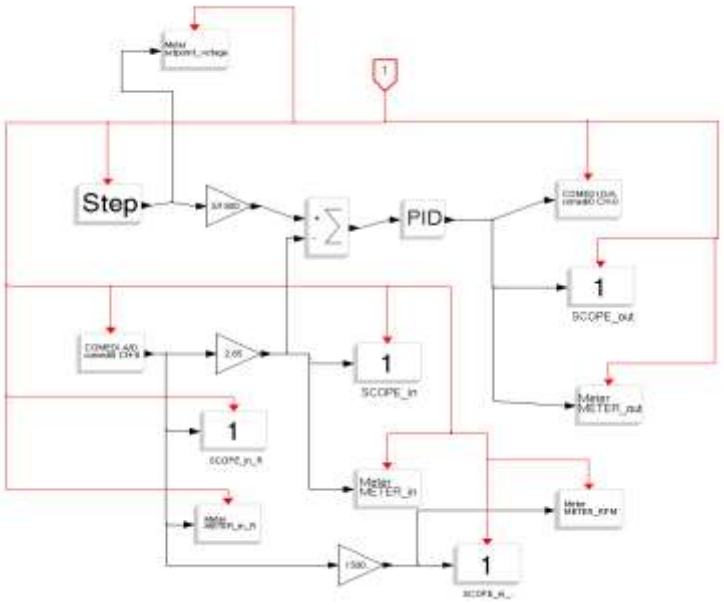

Figure 1. Scicos Diagram for PID control of DC motor

Step-1:

- Create a subfolder in your home folder named *test*
- Open Scicoslab.
- In upper bar, toolboxes -> RTAI. Check for "Scicos-RTAI Ready" output.
- Type scicos in command window to open modeling GUI.
- In the new window, Palette -> Pal Tree to open the list of available blocks.
- Expand Sources and grab the red clock. Place it in the upper part of your model.
- Double-click the clock. From here you can set operating frequency. Type 1/1000 as Period (1KHz)
- In Palettes windows, expand Linear and drag gain block to the model, directly below the clock.
- Now connect blocks as shown in figure 1.

Step-2:

- Make Region to Super Block. It's mandatory that all your model is contained in a single Super Block with only Clock as input as show in figure 2.
- Go back to the main scicos windows, highlight again the Super Block and, in upper bar, RTAI -> RTAI CodeGen.
- Confirm with OK and check Scicoslab window for compiling results. If you see the output "### Created executable: test ###" then your target is ready to run.

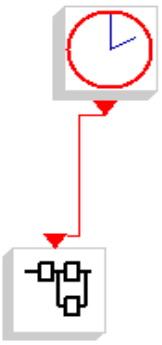

Figure 2. super block of scicos blocks

III. SPECIFICATION AND TEST RESULTS

Hardware and software specification for Real-time control loop of DC motor speed control.

A. Hardware specification

DC motor:

This motor is manufactured by Pranshu Electricals have general specification as below (figure 3);

- Maximum speed: 1500 Rpm
- Maximum input Voltage: 10 vdc
- Maximum input Current: 0.9Amp

Figure 3. DC motor

Optical Encoder Unit:

- This unit is made up of slotted disk which is mounted on back of the motor
- Axis with Infrared light source and detector. This unit provides 12 pulses per revolution.

Frequency to Voltage Control unit:

- This unit is provide for speed Measurement. Pulses from optical encoder is fed to the Frequency to voltage (F-V) converter for measurement of speed in terms of DC voltage.
- Range: 0 to 2 volt DC for 0 to 1500 Rpm.

Signal converter Unit:

- It is Non-inverting amplifier with gain 2 using op-amp 741
- It convert 0-5 volt analog signal to 0-10 volt analog signal
- Here DAQ card pci1711 generate 0 to 5 volt analog but DC motor required 0 to 10 voltage so Non inverting amplifier circuit is used for signal converter as shown in figure 4.

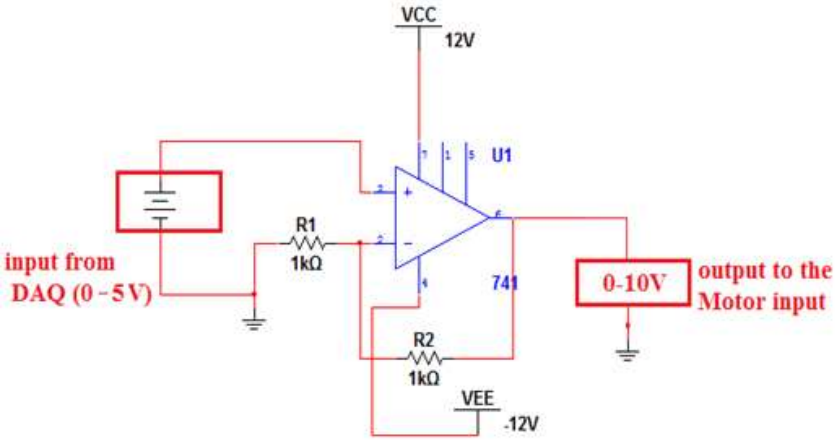

Figure 4. Signal converter circuit

DAQ card PCI-1711 [19]:

- It is hardware interfacing device that used to interface Real-time hardware to PC (figure 5)

Figure 5. DAQ card with its connection

B. Software tools

- RT-Linux
- Scicoslab
- RTAI library
- QrtaiLab

C. Speed control of DC motor (testing loop)

Block diagram of whole setup is given in figure 6.

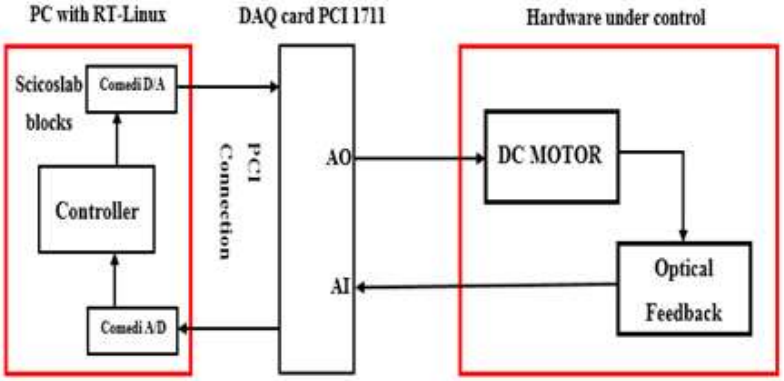

Figure 6. Block Digram

After starting of Automation test platform develop *standalone Executable* file by Scicos code-gen tool of RTAI [9]. To analysis Real-Time simulation follow given procedure.

- In terminal, type
- qrtailab &
- /home/yourusername/testsine
- Check is “TARGET STARTS.” is prompted as output in terminal.
- Switch to QRTAILab window.
- In upper bar, Target -> Connect. You should see in the lower left-hand corner of the window the message “Target: testsine”.

- In upper bar, View -> Scope. Check “Show/Hide” option and adjust “sec/div” to get a nicer visualization (i.e. 0.05). Then close manager.
- Now we try to control wave amplitude on-the-fly. In upper bar, View -> Parameter.
- Double click on gain, then select Block Parameters tab. Try different values and see changes in the scope. For Different set point value of DC motor speed analysis output shown by different scope as shown in figure 7 , 8 , 9, 10.
- Now go back to QRTAILab main window. In upper bar, Target -> Stop. Then close QRTAILab

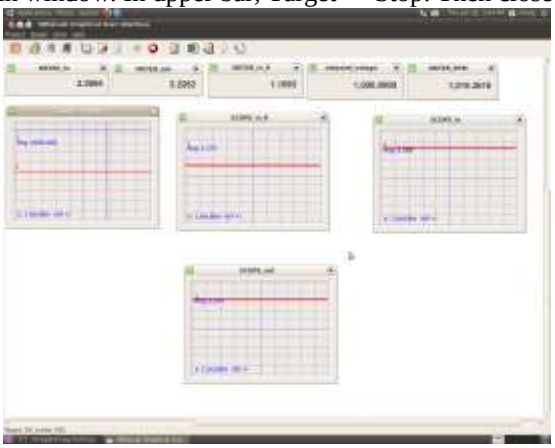

Figure 7. Screenshot of Real-time PID control of motor with set point of 1000 RPM

As show in figure 3 to 5 , when different set point is given to the Controller then speed of the DC motor reach set value which is shown in METER_RPM.

It is hardweare in loop configuration here DC motor speed is controlled by soft controller developed in scicoslab.

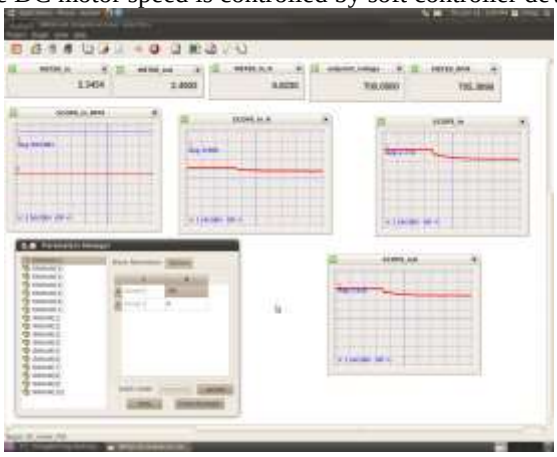

Figure 8. Screenshot of Real-time PID control of motor with set point of 700 RPM

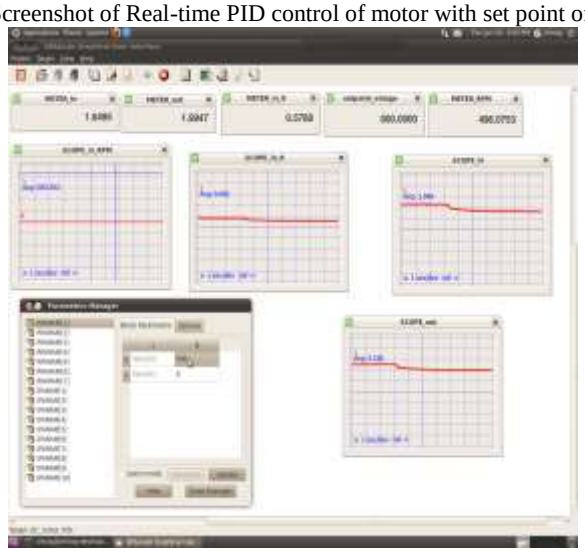

Figure 9. Screenshot of Real-time PID control of motor with set point of 500 RPM

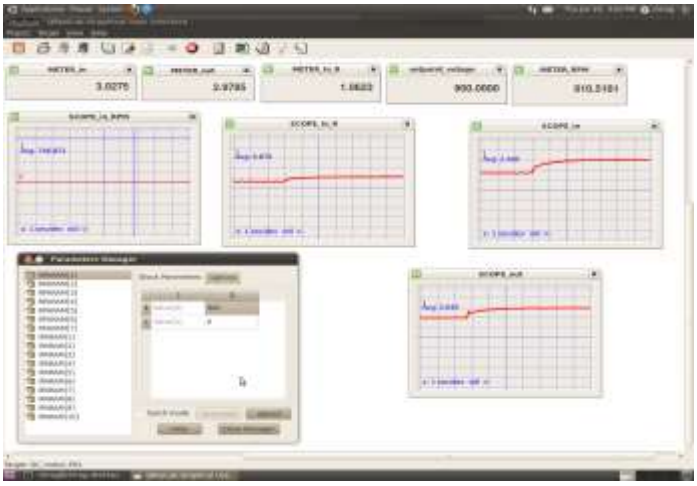

Figure 10. Screenshot of Real-time PID control of motor with set point of 900 RPM

IV. CONCLUSION

It is concluded that development of PID control algorithm for real-time control of DC motor is successfully done with the help of open source Automation test Platform.

Hence, it is possible to replace high cost propriety software with free open source software.

It is possible to develop low cost Real-time platform Environment using free open source software code SCILAB/SCICOS, COMEDI, and RTAI in open source Linux operating system. This Environment can be used to control real-time application and also used for testing of hardware system.

REFERENCES

[1] V. Yodaiken, "RTLinux," FSMLabs, Wind River Systems, [Online]. Available: http://en.wikipedia.org/wiki/RTLinux.

[2] M. works, "Verification, Validation, and Test," [Online]. Available: http://www.mathworks.in/verification-validation/.

[3] P. S. S. K. D. Ujjwal Mondal, "Real-Time Speed Control of a DC Motor using Open Source Code Tool," *nternational Journal of Soft Computing and Engineering (IJSCE),* vol. 2, no. 6, pp. 358-362, january 2013.

[4] R. Team, "RTAI - the RealTime Application Interface for Linux," RTAI distribution, [Online]. Available: http://www.rtai.org.

[5] R. N. ,. S. S. Simone Mannori, "Free and Open Source Software for Industrial Process Control Systems," *National Institute For Research in computer science & control.*

[6] P. P. H. a. K. N. Shubhanga, "Development of a Relay Test Bench and an Arbitrary Waveform Generator in RTAI-Linux Platform," *IEEE ICECCN,* 2013.

[7] SCILAB, "scilab - Open source software for numerical computation," 2003. [Online]. Available: http://www.scilab.org/.

[8] Scilab, "scilab Wikipedia," Scilab Enterprises, [Online]. Available: http://en.wikipedia.org/wiki/Scilab.

[9] Scicos, "Scicos: Block diagram modeler/simulator," INRIA product, 1994-2011. [Online]. Available: http://www.scicos.org.

[10] RTAI-Team, "RTAI - the RealTime Application Interface for Linux," Department of Aerospace Science and Technology of the Polytechnic of Milan, [Online]. Available: https://www.rtai.org/.

[11] S. M. T. N. Roberto Bucher, "RTAI-Lab tutorial: Scilab, Comedi, and real-time control," February-2008.

[12] S. M. T. N. Robert Bucher, "RTAI-Lab Tutorial," 2008.

[13] M. J. L. J. L. Luis Garcia, "Hardware-in-the-loop Environment for Control Systems evaluation under Linux/RTAI," *International Conference on Applied Computer Science, Tenerife, Canary Islands, Spain,* pp. 16-18, December 2006.

[14] Linux, "Linux-Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Linux.

[15] A. Ivchenko, "Real-Time-Linux," 01 May 2001. [Online]. Available: http://www.embedded.com/design/connectivity/4023328/Real-Time-Linux.

[16] N. Instruments, "LabVIEW for Rapid Control Prototyping and Hardware-in-the-Loop Simulation," National Instruments Corporation, [Online]. Available: http://www.ni.com.

[17] d. A. Industry, "DS1104 R&D Controller Board," dSPACE Inc., [Online]. Available: https://www.dspace.com.

[18] C. (. a. M. D. I. Drivers), "Linux Control and Measurement Device Interface," 2012. [Online]. Available: http://www.comedi.org/.

[19] A. P. 1. DAQ, "Industrial Automation - Advantech," [Online]. Available: http://www.advantech.com/.

[20] R. B. a. S. Balemi, "Scilab/Scicos and Linux RTAI - A unified approach," *IEEE Conference on Control Applications Toronto,* pp. 28-31, August 2005.

[21] ,. p. v. p. Chirag panchal, "Testing and Measurement Work bench in Real-Time Linux Environment," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. Volume 3, no. Issue 2, pp. 313-318, February 2014.