

Performance Analysis of Robotic Arm Manipulators Control System under Multitasking Environment

Adnan Al Moshi, Salwa Salam Cynthia, Eftakhairul Islam, Rumana Rahman and Akm Abdul Malek Azad
Department of Electrical and Electronic Engineering, BRAC University, Dhaka 1212, Bangladesh

Received: September 22, 2011 / Accepted: October 11, 2011 / Published: May 25, 2012.

Abstract: This work is to observe the performance of PC based robot manipulator under general purpose (Windows), Soft (Linux) and Hard (RT Linux) Real Time Operating Systems (OS). The same open loop control system is observed in different operating systems with and without multitasking environment. The Data Acquisition (DAQ, PLC-812PG) card is used as a hardware interface. From the experiment, it could be seen that in the non real time operating system (Windows), the delay of the control system is larger than the Soft Real Time OS (Linux). Further, the authors observed the same control system under Hard Real Time OS (RT-Linux). At this point, the experiment showed that the real time error (jitter) is minimum in RT-Linux OS than the both of the previous OS. It is because the RT-Linux OS kernel can set the priority level and the control system was given the highest priority. The same experiment was observed under multitasking environment and the comparison of delay was similar to the preceding evaluation.

Key words: Control system, DAQ (data acquisition) card, jitter, multitasking, RT-Linux.

1. Introduction

In many fields, applications where technical supports are required, man-handling is either dangerous or is not possible [1]. The authors had an opportunity to visit an industry of electronic goods like energy bulb, regulator, ballast, etc. where employees were doing some hazardous job like deep-soldering in hand. In such situations that accuracy and safety are the main concern or more robotic arm manipulators can be used to meet the demand of speeding up the automation process [2-4]. Consequently, the authors thought about a manipulator as well as operating system which could reduce the delay. The main focus of the project is to drive the robotic arm manipulators under general purpose (Windows), Soft (Linux) and Hard (RT-Linux) Real Time Operating Systems (OS) to show that in case of RT-Linux, accuracy is greater than in case of any other OS.

Corresponding author: Akm Abdul Malek Azad, Ph.D., associate professor, research field: control systems. E-mail: a.azad@bracu.ac.bd.

A large number of works has been done on error free system of stepper motor based on microcontroller or using Linux based program or modified driver circuit. But in this work, the specialty is to drive the robotic arm manipulator with maximum accuracy in case of uninterrupted multitasking where the task is processing in RT Kernel [5-7]. The authors implemented an open loop system for stepper motors of robotic arm manipulators and the system was operated with and without multitasking. The driver program can operate multiple robotic arms without any interrupt, i.e., parallel processing can be done. The stepper motors of robotic arm are driven using DAQ card. The performance in RT-Linux environment was compared with the performance found from non real time Linux and Windows OS. The paper is organized as follows: Section 2 discusses the open loop control system; section 3 discusses about the periodic jitter under hard RT OS multitasking environment; section 4 introduces the reference model of the robotic manipulator; section 5 describes the testing phase; section 6 presents the testing results;

section 7 gives conclusions.

2. Open Loop Control System

An open loop control system (OLCS) does not contain any form of feedback in the system. The output is completely dependent on input and does not have any methodology for correction. Depending only on the state of input, output is calculated. For this reason, it cannot be determined whether or not the desired output can be achieved.

Because of its simplicity, low cost implementation, extensibility, portability, scalability and interoperability, open loop control system are used mostly in simple processes where there is a high probability of achieving desired output and undesired output will not result in any malfunction. Open-loop stepper motor modules are used in industrial robotic arm manipulators for each degree of freedom. The authors obtained satisfactory test results under this open loop operation of stepper motor [8].

3. Periodic Jitter under Hard RT OS Multitasking Environment

Periodic jitter is the periodic variation of the significant instances of a signal from their ideal location in time. In case of a stepper motors periodic jitter occurs. Fig. 1 shows periodic jitter. If non hard real time based OS (Linux, Windows, etc.) is used, the periodic jitter causes time delay. On the other hand, RT-Linux does not have any significant time delay, which is efficient for controlling any control system [9].

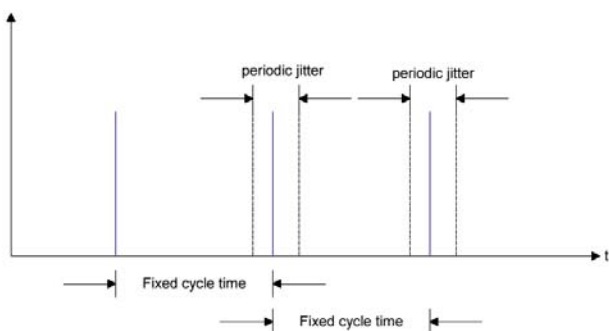


Fig. 1 Periodic jitter.

4. Reference Model of Robotic Manipulator

A reference model of robotic arm manipulator based upon the open loop control system of stepper motor is proposed. There are two robotic arm manipulators in the model; each consists of two stepper motors and one DC motor for gripper. For analysis in different OS, the authors considered the stepper motors only.

4.1 Hardware Part

The hardware part mainly consists of two robotic arms and a conveyer belt. Each robotic arm is consisting of two stepper motors and a DC motor. Fig. 2 shows the project set up in the laboratory.

The hardware part can be further divided into several parts.

4.1.1 DAQ Card

The main purpose of this card is to receive and send signal to the external hardware interface. It is mainly used for multitasking in real time environment [10]. Data Acquisition card (DAQ, PLC-812PG) has following specifications:

- It is a high performance, high speed and multi-function data acquisition card for IBM PC and compatible computers;
- It is ideal for wide range of applications in industrial and laboratory environments. These applications include data acquisition, process control, automatic testing and factory automation;



Fig. 2 Reference model of the robotic manipulator.

- A DAQ card is used to send and receive data from the hardware interface and multiple tasks can be controlled.

4.1.2 Stepper Motor

A unipolar stepper motor has 5 wires with step size of 1.8 degree. It mainly converts the electrical pulse into mechanical movement. It can rotate in clock wise and anti clock wise direction according to the given pulse.

Fig. 3 shows the open loop system for stepper motor.

4.1.3 TIP 122

TIP122 is used for the amplification in the driver circuit in this work. Stepper motors require high current but the DAQ card gives some mille ampere current as output. To achieve current amplification TIP122 is used. The driver circuit takes data from the DAQ card and drives the motor according to the data [11].

4.2 Software Part

4.2.1 Main Specification of RT-Linux

- The RT-Linux kernel separates the hardware from the user-level task.
- Kernel has the ability to use Schedule algorithm to give priority to each task.
- RT-Linux Kernel is situated between the standard Linux and the hardware [12].
- RT-Linux user can introduce and set priority at each and every task.

4.2.2 Flow Chart

The algorithm of the control system is shown in Fig. 4. It describes the manipulation of the two robotic arms.

5. Testing Phase

An Open Loop Control System (OLCS) does not contain any form of feedback in the system. Multitasking is the ability of an operating system of allowing a user to execute more than one computer task at a time. For example, when a web browser and a word

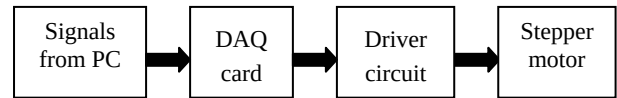


Fig. 3 Block diagram of the open loop system for stepper motor.

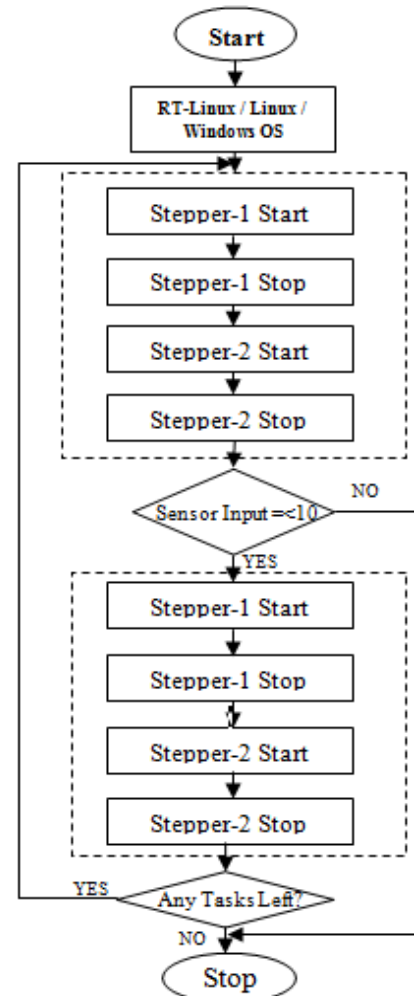


Fig. 4 Flow chart of the system.

file are opened at the same time, it will cause the operating system to do multi tasking. In multitasking, only one CPU is involved, but it switches from one program to another quickly without losing any information. But it cannot allow unlimited number of tasks to be performed at the same time. Each task consumes system storage and other resources. As a result, when more tasks are started, the system may slow down or begin to run out of shared storage. In control system, the authors performed multiple tasks at the same time to analyze the delay in different OS.

6. Test Results

An experiment in BRAC University laboratory has been performed to compare the time delay of the system in different environment (RT-Linux, Linux and Windows). Here the authors have used a DSO (Digital Storage Oscilloscope) to take the reading for the control system with and without multitasking. At that time three tasks were running simultaneously for each of the case. The time resolution for the test is 2 s.

In case of multitasking environment, tasks running simultaneously were 2games, file system software—emacs and 1 video player which was playing video.

The time delay between the performances of two robotic manipulators in different OS can be seen from Figs. 5-10. The time delay is calculated from differences of the value of cursor A (Cur A) and cursor B (Cur B) where Cur A shows the performance of first manipulator and Cur B shows the performance of second manipulator. Figs. 5-6 show the difference of the time delay for Linux OS with multitasking and without multitasking condition. Similarly, Figs. 7-8 show the difference of time delay for Windows OS. Similarly, Figs. 9-10 show the difference of the time delay for RT-Linux OS with and without multitasking condition.

From Table 1, it can be seen that in case of 2 s time resolution Windows takes 7.52 s, Linux 6.0 s whereas Hard RT-Linux takes 4.0 s in multitasking mode. So, here RT-Linux is better than Linux and much better than Windows OS. If the timing resolution is increased,

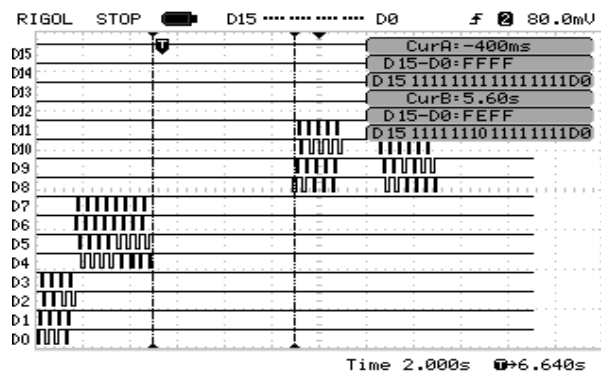


Fig. 5 Wave form for Linux without multitasking. (For 2 s period) Total delay = 6.64 s in one cycle.

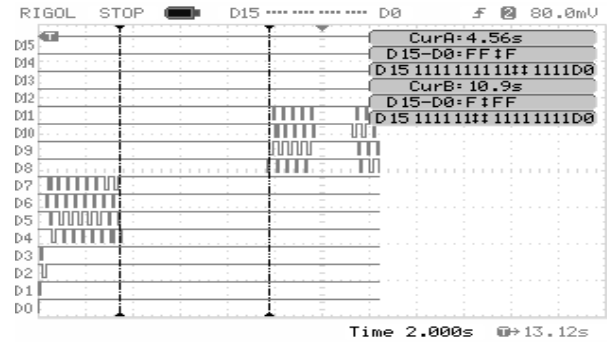


Fig. 6 Wave form for Linux in case of multitasking. (For 1 s period) Total delay = 13.12 s in one cycle.

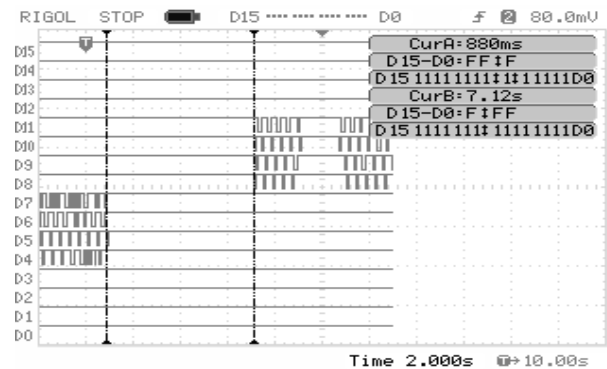


Fig. 7 Wave form for Windows without multitasking. (For 2 s period) Total delay = 10.0 s in one cycle.

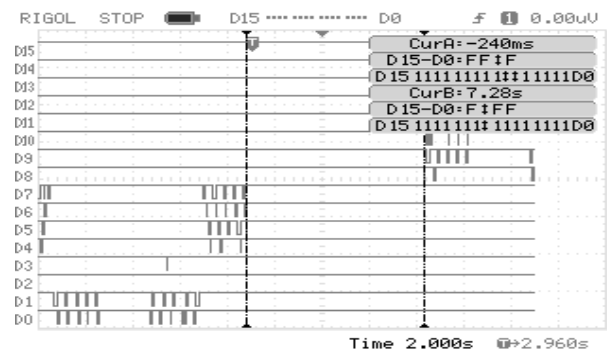


Fig. 8 Wave form for Windows in case of multitasking. (For 1 s period) Total delay = 2.96 s in one cycle.

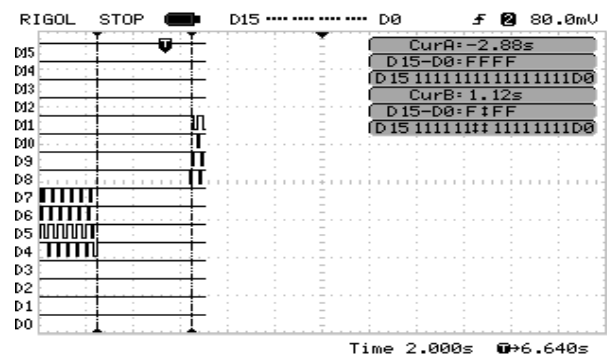


Fig. 9 Wave form for RT-Linux without multitasking. (For 2 s period) Total delay = 6.64 s in one cycle.

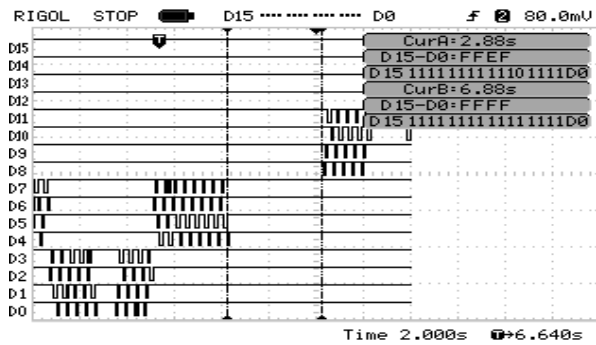


Fig. 10 Wave form for RT-Linux in case of multitasking. (For 2 s period) Total delay = 6.64 s in one cycle.

Table 1 Comparison among soft real time and hard real-time OS with and without multitasking environment.

Operating system	Application	Time comparison		
		$T_1^*(s)$	$T_2^{**}(s)$	$T_1 - T_2(s)$
Windows	Non real time	7.52	6.24	1.28
Standard Linux	Soft real time	6.34	6.0	0.034
RTOS Kernel: Knoppix	Hard real time	4.0	4.0	0

* T_1 = Delay under multitasking (s)

** T_2 = Delay without multitasking (s)

then Linux and Windows will give irregular movement of the stepper motors, but RT-Linux will work properly. At the time of testing, it have been seen that there are also some miss steps of stepper motors in increased time resolution. On the other hand RT-Linux is working properly.

Similarly, the authors have checked the system response for hardware interrupt. Because that for emergency case like short circuit when fire alarm would rang, that alarm can be used as hardware interrupt to shut down the whole system. In this case, immediate response is most important which can be done using RT Linux Operating System, as it can execute operations as far priority and hardware interrupt (fire alarm) will have the first priority.

7. Conclusions

From the results, it can be observed that for the control system of the robotic arm manipulator in multitasking environment performances on the basis of time delay and stability criteria RT-Linux (RTOS) based program performs better than Soft Real Time

and general purpose OS.

Acknowledgment

This project has been financed by EEE department of BRAC University to accomplish the final year thesis project.

References

- [1] T. Kanade, M.L. Reed, L.E. Weiss, New technologies and applications in robotics, Communications of the ACM 37 (3) (1994) 58-67.
- [2] W. Darrell, A genetic algorithm tutorial, Computer Science Department, Colorado State University Fort Collins, CO 80523.
- [3] J.J. Craig, Introduction to Robotics: Mechanics and Control, Pearson Education Asia, 2004.
- [4] S.R. Deb, Robotics Technology and Flexible Automation, Tata Mc-GrawHill, 2002.
- [5] G. Harsha, G.B. Dhamnekar, A Linux based simulator for a programmable multi axes controller, Raman Research Institute, Bangalore, 2004.
- [6] T. Hanamoto, H. Ikeda, T. Tsuji, Y. Tanaka, Sensorless speed control of synchronous reluctance motor using RTLinux, in: Proceedings of the Power Conversion Conference 2002, Osaka, Vol. 2, pp. 699-703.
- [7] T.C. Chin, D.P.Mita1, M.A. Jabbar, A stepper motor controller, Nanyang Technological Institute, Singapore.
- [8] A.A. Amin, A.A. Faruk, M.A. Alam, A.M. Azad, Stepper motor performance under real-time multitasking environment, in: TENCON 2009, Singapore, November 23-26, 2009.
- [9] Introduction to Linux for Real-Time Control, National Institute of Standards and Technology Prepared for Intelligent Systems Division National Institute of Standards & Technology 100 Bureau Drive, Mail Stop 8230 Gaithersburg, MD 20899-8230.
- [10] User manual of PCL-812PG (enhanced multi-lab card) by advantech automation with PCs.
- [11] R.V. Aroca, D.M. Tavares, G. Caurin, Scara robot controller using real time Linux, in: Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2007, pp. 1-6.
- [12] C.M. Hussain, M. Alam, A.M. Azad, Performance comparison of PID controller in soft and hard real-time operating systems, in: China-Ireland International Conference on Information and Communications Technology (CICT2008), Beijing, September 26-28, 2008.