Master's Thesis
석사 학위논문

The construction of control environment and control new radiation robot

Deayong Park (박 대 용 朴 大 龍)

Department of Robotics Engineering
로봇공학 전공

DGIST

2015

Master's Thesis
석사 학위논문

The construction of control environment and control new radiation robot

Deayong Park (박 대 용 朴 大 龍)

Department of Robotics Engineering
로봇공학 전공

DGIST

2015

The construction of control environment and control new radiation robot

Advisor : Professor Pyunghun Chang

Co-advisor : Professor Weon kuu chung

by

Dea Yong Park
Department of Robotics Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science, in the Department of Robotics Engineering. The study was conducted in accordance with Code of Research Ethics[1]

07. 08. 2015

Approved by

Professor (Advisor)	Pyung Hun Chang	(Signature)
Professor (Co-Advisor)	Weon kuu Chung	(Signature)

[1] Declaration of Ethical Conduct in Research**:** I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

The construction of control environment and control new radiation robot

Dea Yong Park

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

06. 01. 2015

Head of Committee	Prof. Pyung Hun Chang	(인)
Committee Member	Prof. Weon Kuu Chung	(인)
Committee Member	Prof. Jae Sung Hong	(인)

MS/RT

201323003

박 대 용. Deayong Park. The construction of control environment and control new radiation robot. Department of Robotics Engineering. 2015. 32p. Advisors Prof. Chang, Pyung Hun, Co-Advisors Prof. Chung, Weon Kuu.

ABSTRACT

In this paper, the construct the control environment for the new radiation robot and the control the prototype of the new radiation robot using time delay control (TDC). The design of the new radiation robot has been developed in author's laboratory. In order to control the new radiation robot, it is necessary to build a control environment and be applied to the control law. The control environment used real time application interface (RTAI) to enable strict time control. In strict time control, the control system can perform the intended function in the desired time. The control law used in this thesis is time delay control (TDC). TDC is control scheme to compensate for the amount of the nonlinear dynamics of a plant and unpredictable disturbances by using time delay estimation scheme. This control method has a simple structure and requires small computation. In order to verify the construction of control environment for new radiation robot and to check the tracking error of TDC, experiment is conducted. The sampling time is measured in real time operating system. The number of its data is 21000 and its average is 1ms. The average of latency is 0.003047ms. Tracking error of TDC for all joints is within ±0.2 degree. Thus, the control environment for real-time is well established and it shows the possibility of accurate position control for the new radiation robot by using TDC.

Keywords: New radiation robot, Real time application interface (RTAI), Time delay control (TDC)

Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Background

Worldwide, the number of cancer patients continues to increase. The number of cancer patients in 2014, according to figures released by the WHO was 14.1 million people, and 8.2 million were terminal cases [1]. If this trend continues over the next 20 years, it is expected that approximately 25 million people will have cancer by 2045. Furthermore, in Korea, the number of cancer patients continues to increase. According to statistics from 2012, the number of the cancer patients in Korea was 220,000 while 75,000 of those were terminal cases. [2].

Many methods for the treatment of different types of cancers have been researched. Additionally, methods for treating cancer are generally divided into three types: surgical therapy, chemotherapy, and radiation therapy. We will be focusing on the latter.

Radiation therapy, which came about in the 20th century, was a big step forward in the treatment of cancer. In terms of history, the Roentgen discovered the X-ray in 1895, and Curie found and began radium radiation therapy in 1898. After the first radiation treatment, people began studying radiation biology and radiation physics. These researchers greatly affected the development of radiation therapy equipment and contributed immensely to the rapid development of radiation therapy during the 20th century.

Table 1. History of radiation equipment & technology

Years	Technology	Equipment
1920	Fractionated RT	
1930	Orthovoltage RT	
1950	Megavoltage RT	Co-60 Teletherapy LINAC
1960	Radiosurgery	Gamma-Knife
1970	CT scan	Proton therapy

1980	MRI	LINAC-based Radiosurgery
1990	3D Conformal RT, FSRT	3D RTP CT-simulator
2000	IMRT PET IGRT	Tomotherapy Cyberknife

Table 1 shows the 20th century history of radiation equipment. The first medical linear accelerator (LINAC) was built by Henry Kaplan and Edward Ginzton in 1952, and was first used on a patient in 1953 [1]. In 1968, the first gamma knife using 6oCo was installed at the Karolinska Institute by Swedish neurosurgeon Lars Leksel [2].

The first CyberKnife was invented by Dr. John Adler in 1994. The initial model was created to treat the brain based on the stereotactic radiosurgery system (SRS). Due to technological developments, the Cyberknife can now treat the entire body as well as the brain. This means that tumor tracking and correction are now possible. In other words, the operator of the Cyberknife can more accurately deliver the desired dose at the desired location due to current developments in tracking and position correction scheme [3, 4].

Technology and equipment developed in the 20th century were developed to increase the maximum efficiency of the radiation therapy goal. The radiation therapy goal is 'The maximum dose delivered to tumor and the minimum does delivered to normal cells for minimize side effects'.

In my laboratory, we developed a new radiation robot which can increase the effectiveness of radiation therapy. The link of a new radiation robot is spherical and has always targeting center point as kinematics characterized. Placing the tumor at this point all the time, so the accuracy of the location will be guarantee. However accuracy of the location and the orientation are needed for accurate delivery radiation.

1.2 Objective

The design of the new radiation robot has been completed. In order to precisely control the new radiation robot, it is required to have a new control system. The control system must include the functions necessary for the purposes of the robot. So also it varies according to the robot control system.

The control system is configured to the control environment and the control law. The control environment consists of hardware and software. The hardware of the control system is a combination of devices such as a PC, a DAQ board, and a DA board. These receive an external signal and calculate control law and output signal as well. The software part is made up of the operating system (OS) and the device driver. When establishing a controlled environment, hardware is selected according to the user's requirements. However, there are many considerations in building the software that inevitably determines the hardware one ultimately decides to use.

Thus, one should choose an OS with the construction of software in mind. Operating systems are divided into two types, the General Purpose Operating System (GPOS) and the Real Time Operating System (RTOS). GPOSs such as Linux, Windows, and Mac OS are commonly used for personal computers that don’t require any special needs. The GPOS is useful for general users, but it is not suitable for control systems. GPOSs have no ability for deterministic or time-critical behaviors. However, the RTOS is an operating system that can process a variety of tasks with time scheduling and adjustable priority levels. It can make precise results while running critical tasks [5].

The RTOS is often used in systems that require highly reliable such as robot controls, data acquisition systems, manufacturing plants, and other time-sensitive instruments and machines. The control system of the new radiation robot also requires high reliability. The control system of the new radiation robot should not ever malfunction during surgery hours. Additionally, there are several different types of RTOSs. VxWorks is the most compatible RTOS; Windows CE has been commercially successful; QNX Neutrino is commonly used for multiple node systems; RTAI is also a popularly used RTOS [6].

Of these, the Real-Time Application Interface (RTAI) is viable and effective open-free software. It can be change a GPOS to an RTOS. It is adding hard real time capabilities to the Linux kernel [7]. Additionally, the RTAI has been used in many fields and has proven to be highly reliable. The advantage of the RTAI is that it makes it possible to have strict time control and performance, such as the commercial RTOS[6, 8]. In strict time control, the control system can perform the intended function in the desired time. If this advantage applies to radiosurgery robot position control, the control system may receive position information from the motor encoder and send a control signal in correctly the planned time. In addition, it is also possible to apply image processing

using X-ray. The X-ray image is used to collect information about the position of the tumor. For this reason, the control system is required to have an RTAI.

Time Delay Control (TDC) is control law used in control system. This control law is made by two researchers. In MORGAN's research in 1985, the TDE scheme is used to remove chattering in sliding mode control[9]. The TDC law of general linear or nonlinear systems is derived from K.Youcef-Toumi[10, 11].This TDC law is based on the Proportional Differential (PD) control for robot manipulators and is derived from T. C. Hsia around the same time [12]. TDC is control scheme to compensate for the amount of the nonlinear dynamics of a plant and unpredictable disturbances by using time delay estimation scheme. This control method has a simple structure compared with the development of higher control algorithm so far.

In this paper, the construct the control environment for the new radiation robot and the control prototype of the new radiation robot using TDC will be illuminated.

2. METHODS

2.1 Design of New radiation robot

In this chapter, new radiation robot in authors' laboratory will be introduced in detail. The new radiation robot is developed for the radiation therapy goal. Figure 1 shows the design of new radiation robot. This robot consists of two parts, upper arm and lower arm. The upper arm has joint1, joint2 and Linac1. The lower arm has joint3, joint4 and Linac2. The link of a new radiation robot is spherical. This shape results in the advantages. First is Linacs are always targeting the center point in any posture. If the lesions of patients put on this point, guarantee the kinematic accuracy.

Other advantage of new radiation robot is 'reduce surgical time' by using two Linacs. While the Linac1 irradiate x-ray, Linac2 is moved to the next location. When modifying an existing treatment plan by using this method, it is possible to reduce the operation time of radiation.

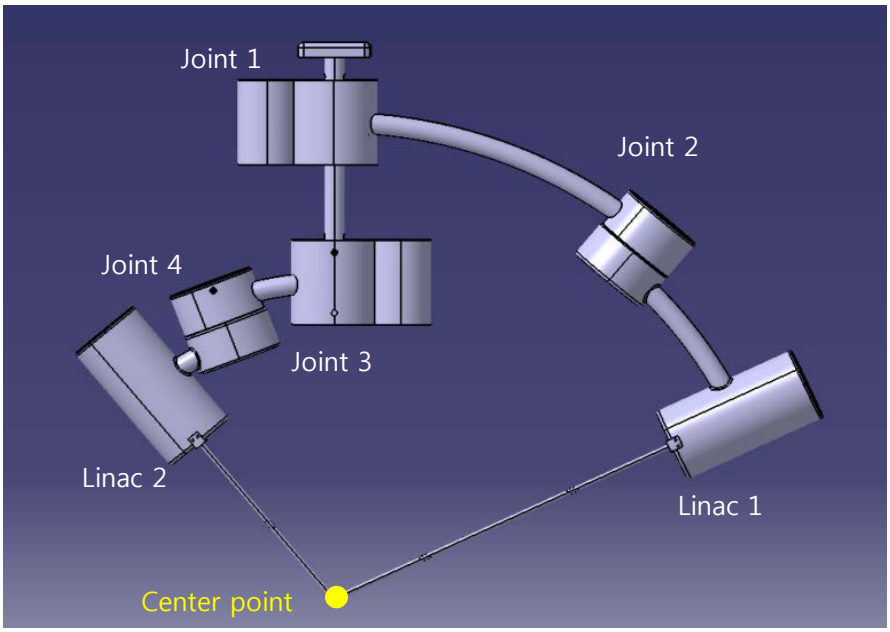

Figure 1. Design of prototype for the new radiation robot

2.2 Control System

In this chapter, we will introduce the control system that consists of three parts; 1) hardware 2) software 3) control law (Time Delay Control, TDC). The hardware part of the device is explained in detail as well as its relationship with the devices. The software part is explained as a real-time operating system and COMEDI drive. The control law part explains about Time Delay Control, TDC.

2.2.1 Hardware

The hardware of the control system is a combination of devices. These devices are an industrial computer, a data acquisition (DAQ) board, an analog output board, and a motor controller.

Below Figure 3 shows the relationship of the devices.

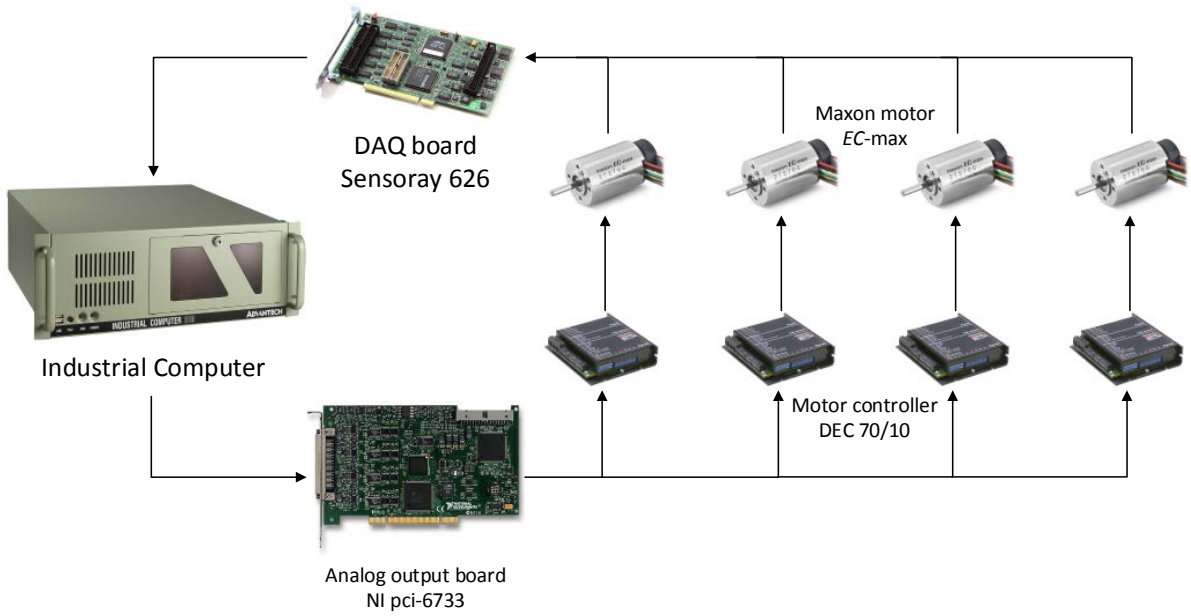

Figure 2. Hardware of control system Diagram

Industrial computer manages other devices using real time operating system (RTOS). Industrial computer sends motor controller (Maxon Motor, 4-Q-EC Amplifier, DEC 70/10) to control commands by using analog an output board (National Instruments PCI-6733). The maxon brushless motor (EC-max 16ϕ) rotates and generates encoder data. The industrial computer receives incremental encoder data that is composed of a binary digital code by using a DAQ board (Sensoray s626) and converts binary digital code to angular position. This angular position is used to feedback control.

2.2.2 Real time operating system

Real time operating system (RTOS) is an integral part of the control system that requires a precise control such as a control system of the new radiation robot. The reason is time scheduling and priories of system [13]. Referring Compare General purpose operating system (GPOS) and RTOS can be easily understood. GPOS such as Windows and Linux, the basic priorities are always selected to keep the operating system. For example, when we run the program in GPOS, GPOS is the case of performing maintenance work related to the high-priority operating system does not process the launcher [13] . Applying the GPOS the robot control system to generate a large error in the robot control. Because priority of operating system kept higher than the precision robot control.

The second is time scheduling [13]. It is related to priorities. In GPOS, when time scheduling, it is difficult to select the relative priority for all applications. For example, the application A is finished and after a period of time, A is restart. However, if the application B is operated before the time to start execution of the application A, application A is delayed than the scheduled time. RTOS is capable of exact time scheduling that is possible to set the priority for each application.

The RTOS various types exist such as RTAI, VxWorks, Windows CE, QNX Neutrino [6]. The Real time application interface (RTAI) is quite effective and open-free-source. However RTAI was used by the robot control environment, the data collection devices, factory automation systems, a number of areas and have a higher reliability. It has better performance of latency than other RTOS [6, 8]. RTAI adds time control capabilities of the RTOS via the patch to Linux based GPOS. [7]. Creating an RTOS by the RTAI, easily add a new function to the robot control law, but also radiation. For example, radiosurgery system, it is also necessary image processing. To add an image processing is added to the device and add the algorithm. These things are possible because of extensibility with the RTAI [7]. RTAI implementation is difficult because of compatibility software issues as well as the complexity of the set-up process. The RTOS consists of a base operating system LINUX, a Real-Time Application Interface (RTAI), and a COMEDI as well.

Table 2. RTOS component

	Version
CPU	Intel i3-3240
Linux	Fedora core 11
Kernel	2.6.32
RTAI	3.8.1
COMEDI	0.7.76
GCC	4.4.1

In this chapter, descriptions of the describe implementation process of the RTAI and COMEI will be provided.

Some essential packages have installed before the RTAI patch at Linux and COMEDI.

- **General**

#yum install cvs subversion build-essential

- **RTAI**

#yum install libtool automake Libncurses.so.5

- **COMEDI-lib**

#yum install bison flex

- **COMEDI-calibrate**

#yum install boost-devel gsl-devel

After these packages were installed, next Linux kernel 2.6.32 and RTAI 3.8.1were downloaded.

cd /usr/src

wget --no-check-certificate https://www.rtai.org/userfiles/downloads/RTAI/rtai-3.8.1.tar.bz2

tar xjvf rtai-3.8.1.tar.bz2

ln -s rtai-3.8.1 rtai

wget --no-check-certificate ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.tar.bz2

tar xjvf linux-2.6.32.tar.bz2

ln –s linux-2.6.32 linux

RTAI patch at linux kernel 2.6.32 in order to use without error and bug.

#cd /usr/src/linux

#patch –p1 < /usr/src/rtai/base/arch/x86/patches/hal-linux-2.6.32.11-x86-2.6-03.patch

Next, the Linux kernel had to configure. In this step, set-up the processor type, the number of the CPUs, select only used hardware, and check the feature of math emulation.

#cd /usr/src/linux

#make menuconfig

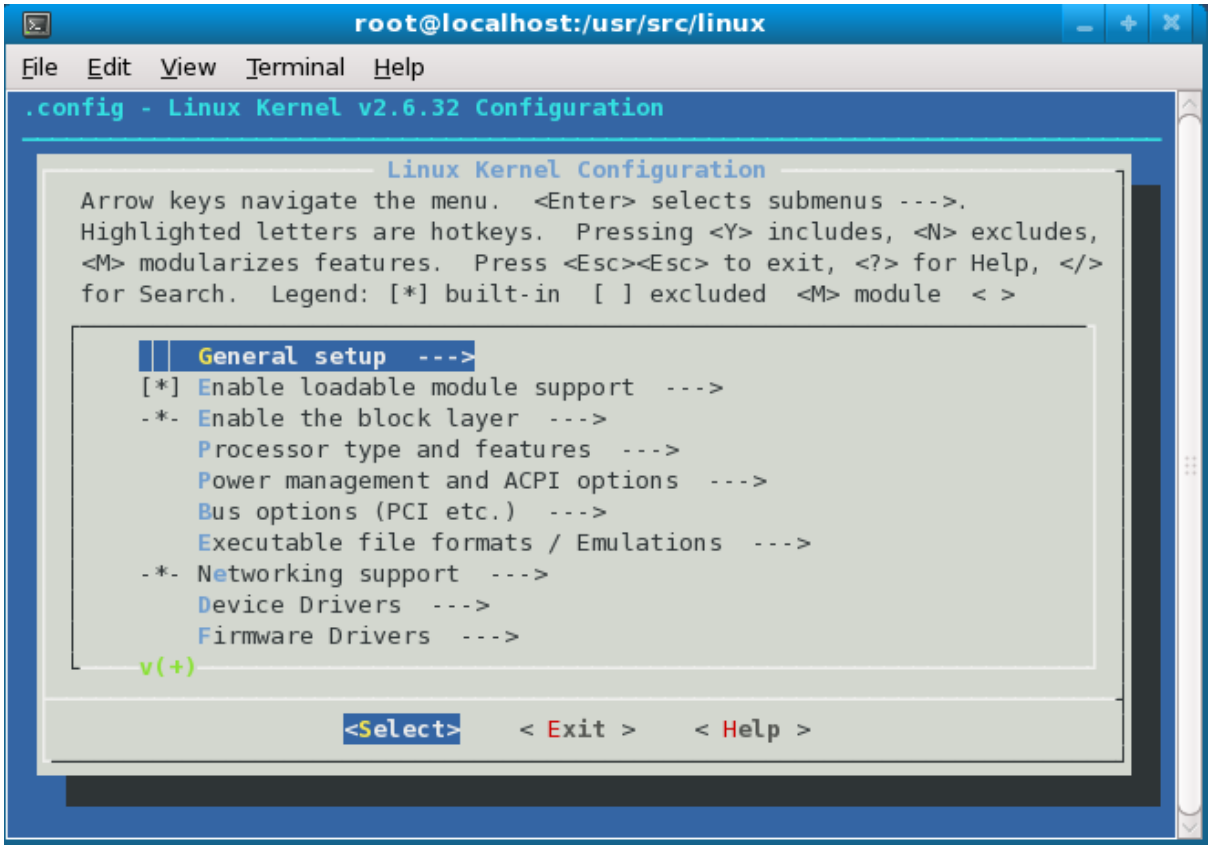

Figure 3. Linux configuration

Compile the kernel and generate the module. Install the new kernel 2.6.32

#make dep

#make clean

#make bzImage

#make modules

#make modules_install

#make install

When the installation is complete, reboot the RTAI patched new kernel. Run the RTAI configure and install. In this step, check the Installation path (:/usr/realtime), Linux source tree (:/usr/src/linux), and the number of the CPU.

#cd /usr/src/rtai

#make menuconfig

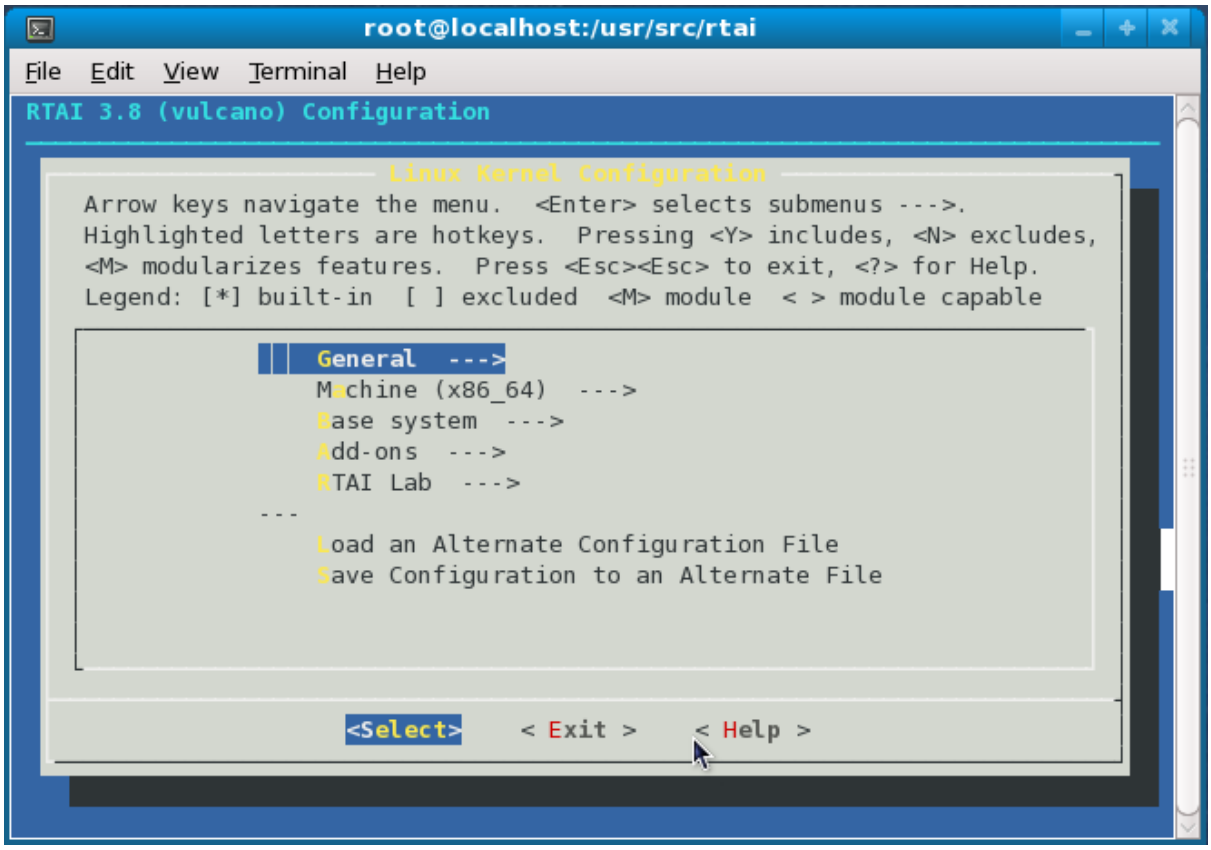

Figure 4 . RTAI configuration

#make

#make install

sed -i 's/\(PATH=\"\)/\1\/usr\/realtime\/bin:/' /etc/environment

```
# export PATH=/usr/realtime/bin:$PATH
```

Reboot RTAI patched kernel and create RTAI devices. In RTAI source tree,

```
# make dev
```

Now the RTAI installation is finished. RTAI provides three example programs. The latency example program is generally used to check the success or failure of installation. If installation is successful, the latency example program will show Figure 6. This program measures the latency is a time interval between the expected switch time (1s) and the actual task time. Figure 6 is acceptable results [14] [15].

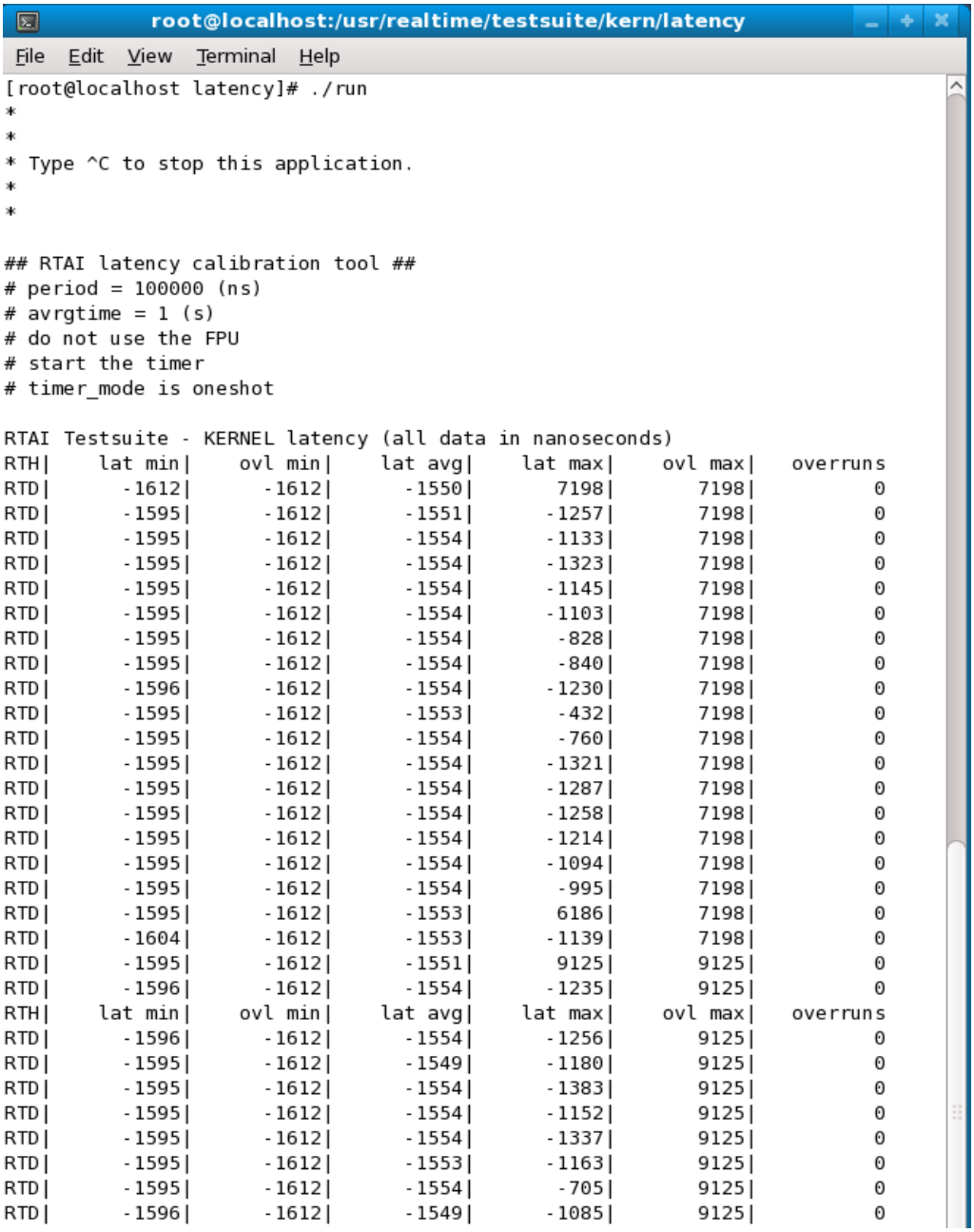

Figure 5. The result of RTAI latency

COMEDI is installed by using a git-package. A git-package is useful for upgrade and downgrades. Download git-core and COMEDI.

#yum install git-core

#cd /usr/local/src/

#git clone git://comedi.org/git/comedi/comedi.git

#git clone git://comedi.org/git/comedi/comedilib.git

#git clone git://comedi.org/git/comedi/comedi_calibrate.git

#git clone git://comedi.org/git/comedi/comedi-nonfree-firmware.git

Configure the linux and RTAI directory. Install the COMEDI.

#cd /usr/local/src/comedi

#sh autogen.sh

#./configure - - with-linuxdir=/usr/src/linux - - with- rtaidir=/usr/realtime

#make

#make install

#make dev

#ldconfig

Install comedilib. Comedilib is a library that provides functions to use COMEDI devices.

#cd /usr/local/src/comedilib

#sh autogen.sh

#./configure

#make

#make install

#mkdir /usr/local/include/linux

Install comedi_calibrate.

#cd /usr/local/src/comedi_calibrate

#autoreconf –I –B m4

#./configure

#make

#make install

RTAI configuration is modified to use COMEDI.

#cp /usr/local/src/comedi/include/linux/comedi.h /usr/local/include/

#cp /usr/local/src/comedi/include/linux/comedilib.h /usr/local/include/

#ln -s /usr/local/include/comedi.h /usr/local/include/linux/comedi.h

#ln -s /usr/local/include/comedilib.h /usr/local/include/linux/comedilib.h

#cd /usr/src/rtai

#make menuconfig

Select ″real time COMEDI support in user space″ in Add-ons menu.

#make

#make install

#cp /usr/local/src/comedilib/include/comedilib.h /usr/local/include/

2.2.3 Time delay control

Typical robot dynamics, the equation (1), has unknown dynamics information and dynamic coupling. Robust control laws are used to control the robot .The new radiation robot also exists such a necessity. Control law used for new radiation robot should ensure a fine position control, and to have a simple control law. The TDC goes to the meeting of these requirements. In explaining the derivation of the control law of TDC, I will explain what TDC is how meets the above requirements.

The dynamics of n – DOF (Degree of Freedom) robot manipulators can be described as follows:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{F}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{1}$$

Where $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{\theta}}$ is the vector of a joint angle of $n \times 1$, the vector of a joint velocity of $n \times 1$, and the vector of a joint acceleration of $n \times 1$, n is the number of joints in the robot manipulators; $\mathbf{M}(\boldsymbol{\theta})$ is $n \times n$ inertia matrix; $\mathbf{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is Coriolis and centrifugal forces of vector $n \times 1$; $\mathbf{G}(\boldsymbol{\theta})$ is gravity vector of $n \times 1$; $\mathbf{F}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is τ is input torque vector of $n \times 1$.

In (1), $\overline{\mathbf{M}}$ represents the known part of $\mathbf{M}(\boldsymbol{\theta})$. $\overline{\mathbf{M}}$ is the constant diagonal matrix and based on $\mathbf{M}(\boldsymbol{\theta})$. Thus, the equation (2) can be described as follows:

$$\boldsymbol{\tau} = \overline{\mathbf{M}}\ddot{\boldsymbol{\theta}} + \mathbf{H}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) \tag{2}$$

Where nonlinear dynamics of the vector $\mathbf{H}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$ denote the nonlinear dynamics of robot manipulators, friction, and disturbance. $\mathbf{H}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$ is described as follows:

$$\mathbf{H}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) = (\mathbf{M}(\boldsymbol{\theta}) - \overline{\mathbf{M}})\ddot{\boldsymbol{\theta}} + \mathbf{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{F}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{3}$$

By using the Computed Torque Control scheme, $\boldsymbol{\tau}$ is designed as follows:

$$\boldsymbol{\tau} = \overline{\mathbf{M}}\boldsymbol{u} + \hat{\mathbf{H}} \tag{4}$$

$$\boldsymbol{u} = \ddot{\boldsymbol{\theta}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{D}}(\dot{\boldsymbol{\theta}}_{\boldsymbol{d}} - \dot{\boldsymbol{\theta}}) + \boldsymbol{K}_{\boldsymbol{p}}(\boldsymbol{\theta}_{\boldsymbol{d}} - \boldsymbol{\theta}) \tag{5}$$

Where $\widehat{\mathbf{H}}$ denotes the estimated value of $\mathbf{H}$; $\mathbf{K_d}$ and $\mathbf{K_p}$ are the diagonal gain matrix of the Proportional Differential (PD) controller. If $\widehat{\mathbf{H}} = \mathbf{H}$ is valid, the relationship between $\ddot{\boldsymbol{\theta}}$ and $\boldsymbol{u}$ is valid from equation (2) and (4) as follows:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{u} \tag{6}$$

Equation (5) applies to (6). The error dynamic of the closed loop of TDC can be described as follow:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K_D}\dot{\boldsymbol{e}} + \boldsymbol{K_p}\boldsymbol{e} = \mathbf{0} \tag{7}$$

Where, $\mathbf{e}$ is position error. $\mathbf{e} \triangleq \boldsymbol{\theta_d} - \boldsymbol{\theta}$.

To find the estimated value of the nonlinear dynamics vector $\mathbf{H}$, TDC uses a Time Delay Estimation (TDE) scheme. This scheme needs an assumption. Specifically, Time delay L is sufficiently small, the following approximation holds:

$$\begin{aligned} \boldsymbol{H}_{(t)} &\cong \widehat{\mathbf{H}}_{(t)} \\ &= \boldsymbol{H}_{(t-L)} = \boldsymbol{\tau}_{(t-L)} - \overline{\mathbf{M}}\ddot{\boldsymbol{\theta}}_{(t-L)} \end{aligned} \tag{8}$$

Equation (5) and (8) apply to (4). Then, TDC law can be described in tis final form as follows:

$$\boldsymbol{\tau}_{(t)} = \boldsymbol{\tau}_{(t-L)} - \overline{\mathbf{M}}\ddot{\boldsymbol{\theta}}_{(t-L)} + \overline{\mathbf{M}}\left(\ddot{\boldsymbol{\theta}}_{\boldsymbol{d}} + \boldsymbol{K_D}\left(\dot{\boldsymbol{\theta}}_{\boldsymbol{d}} - \dot{\boldsymbol{\theta}}\right) + \boldsymbol{K_p}(\boldsymbol{\theta_d} - \boldsymbol{\theta})\right) \tag{9}$$

If $\overline{\mathbf{M}}$ is selected as a constant diagonal matrix, TDC is independent joint of the controller with PD and $\overline{\mathbf{M}}$ gain. Then, the TDC block diagram can be described simply in Figure 7. In other words, by using the TDE scheme, the TDC does not compute complicated nonlinear dynamics. Advantages of TDC are its efficiency and it also has a light burden in terms of computation.

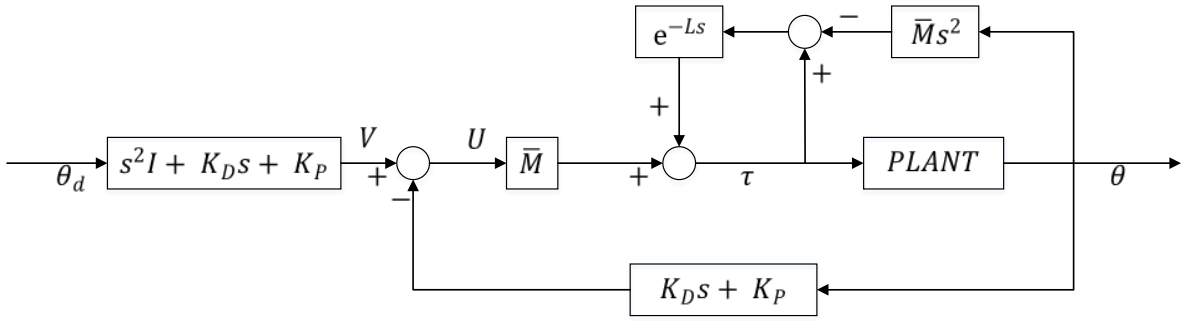

Figure 6. Block diagram of TDC

The robustness of TDC in closed loop systems is determined by precise TDE value in terms of nonlinearity, un-modeled dynamics, and other factors. In the efficiency of the TDE scheme, time delay *L* is very important. In a manner of speaking as qualitative, time delay *L* should be selected with an assumption that it is continuous, to remain valid. Namely, time delay *L* should be as tiny a value as much as possible to estimate required the information well.

3. RESULTS

3.1 Experiment

In this chapter, constitutes an experiment that is possible to check the control environment is well established and control law TDC is applied well. Check the control environment has the error or not by using the run time and latency. Tracking error of the TDC is checked by depicting a part of a conventional Cyberknife treatment plan.

3.1.1 Experiment Setup & Plan

The prototype of new radiation robot is used in this experiment and is shown in Figure 8. This robot is made by 3D printer. This joint of robot used to Maxon BLDC motor. It has gear box and encoder. The ratio of gear box is 157:1. The resolution of encoder is 512 pulse / rev. Thus, the resolution of each robot joint is$0.448 \times 10^{-2} degree$. This robot is controlled by TDC and control environment.

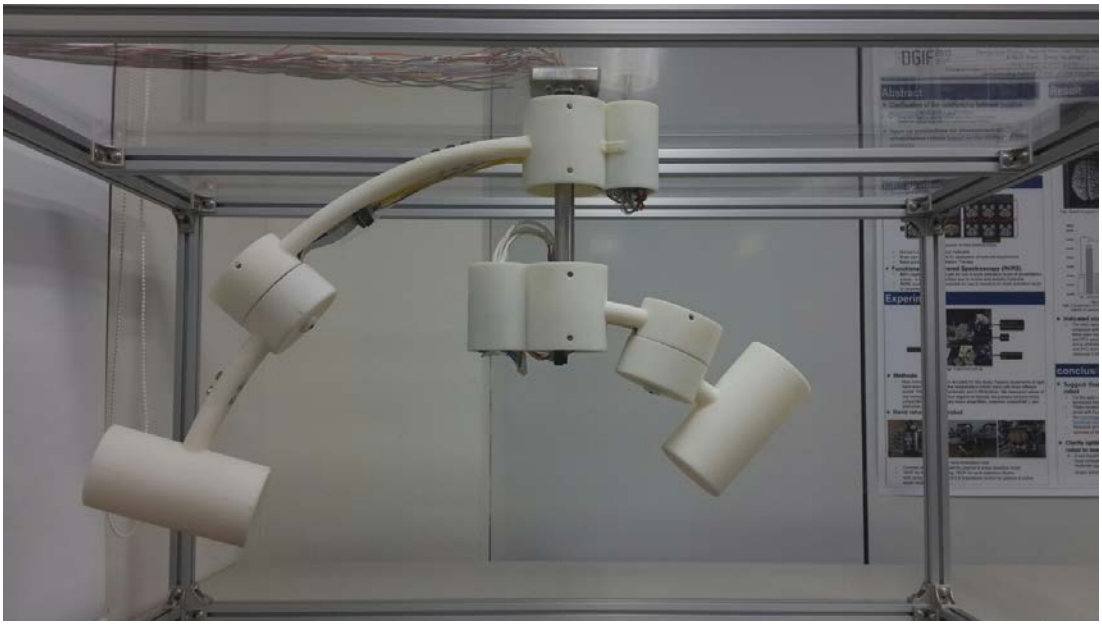

Figure 7. The prototype of new radiation robot made by 3D printer

Through check the run time and latency of sampling time in the RTOS can be sure that the RTAI is properly applied. Run time is the time during which a program is running. If the run time is greater than a predetermined period, it gives an adverse effect on the next operation. Latency is a time interval between the expected time and actual time. Latency of sampling time is a time interval between the programed sampling time and measured sampling time. If the RTAI strict time control feature works well, latency of sampling time will come about 10 microseconds[6]. The significantly value of the latency of sampling time affects the robot control. It means RTAI strict time control doesn't work. While the robot is controlled, obtain the runtime and latency of sampling time. Check the control environment is working well by using these data.

CyberKnife treatment planning includes information such as the intensity of the radiation beam, irradiation time, robot movement, the position of linac. As one example, briefly describe the motion of the robot arm during brain treatment planning. In the brain treatment, virtual sphere centered at the isocenter point with an 800 mm radius. The position of linac is determined in the virtual sphere surface. This predefined position of linac is called "node". The number of nodes is about 130 in brain treatment plan. CyberKnife robot arm will follow 130 nodes in sequence. A sequential way is called "path set". Movement of the robot arm is moving from the initial position to the first node and irradiate beam during the robot arm is stopped. Since beam irradiation, the robot arm moves to the next node. The robot arm of the Cyberknife repeats movement and stop during 130 nodes[3]. Create two nodes and create a path set that connects the nodes. By applying path set to the new radiation robot, check the tracking error of the TDC.

New radiation robot has two robot arms that are upper arm and lower arm. The two nodes are arbitrary selected in each work space of robot arm. The work space and node of upper arm is shown in Figure 9. Green dot point is isocenter that is tumor position. Red circles are work space of upper arm. Blue dot point is node. The work space and node of lower arm is shown in Figure 10. Green dot point is isocenter. Black circles are work space of lower arm. Blue dot point is node. Path set is made by 5^{th} polynomial trajectory. The 5^{th} trajectory has small rate of change in acceleration at the beginning and end of the movement. By this characteristic, new radiation robot requires small torque at both starting and end of trajectory, resulting in small residual vibration.

The trajectory of each joint is shown in Figure 11. In 0~3 second, all joint move to 60 degree so that linac is located in node. In 3 ~ 6 second, new radiation robot stop and irradiate beam. In 6~9 second, robot moves next node. New radiation robot repeats these motion 3 times.

The run time and latency of sampling time are acquired during new radiation robot moving. Below program code is calculation method of run time and latency of sampling time.

```
a_sam_time[cnt] = rt_get_time_ns();
radiation_robot_control();
p_run_time[cnt] = rt_get_time_ns();
```

The function of radiation robot control is main function that has reading encoder, make trajectory, calculate control law, and saving date. Thus, the difference between p_run_time[i] and a_sam_time[i] is runtime. ‘i’ represents the time. , The difference between a_sam_time[i+1] and a_sam_time[i] is sampling time. This measured sampling time is used to calculate latency of sampling time.

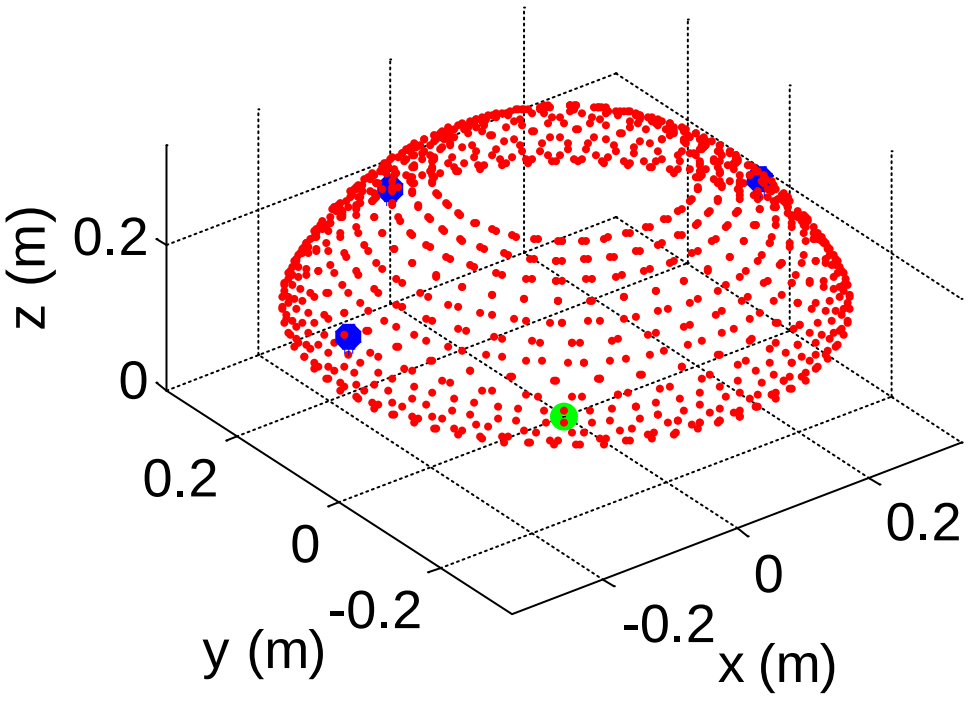

Figure 8. The upper arm work space and node

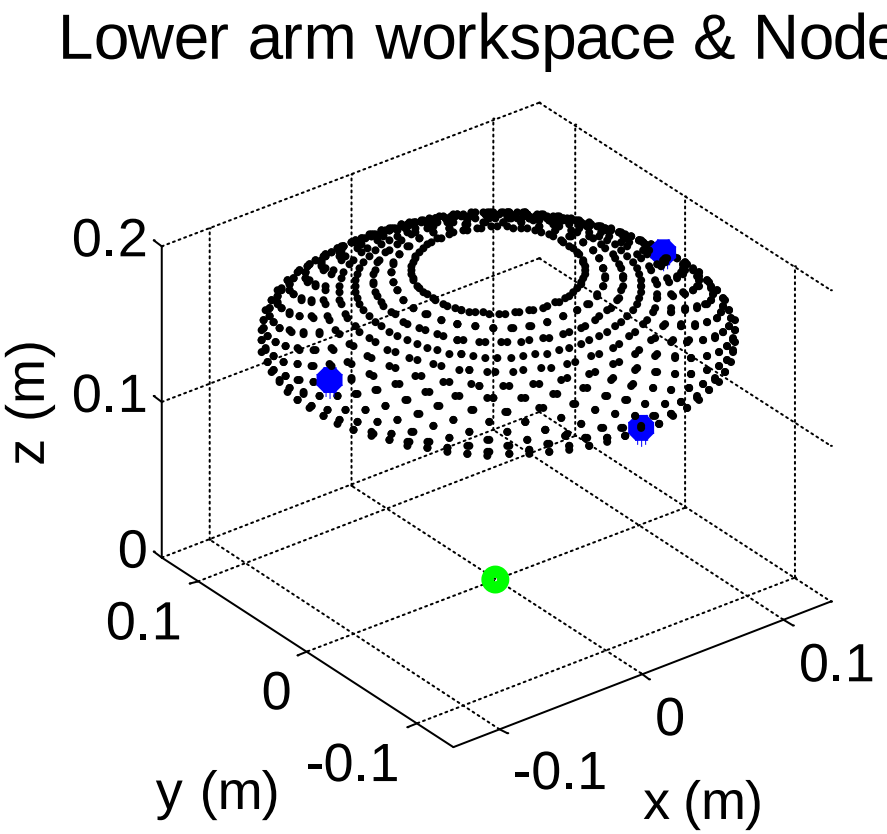

Figure 9. The lower arm work space and node

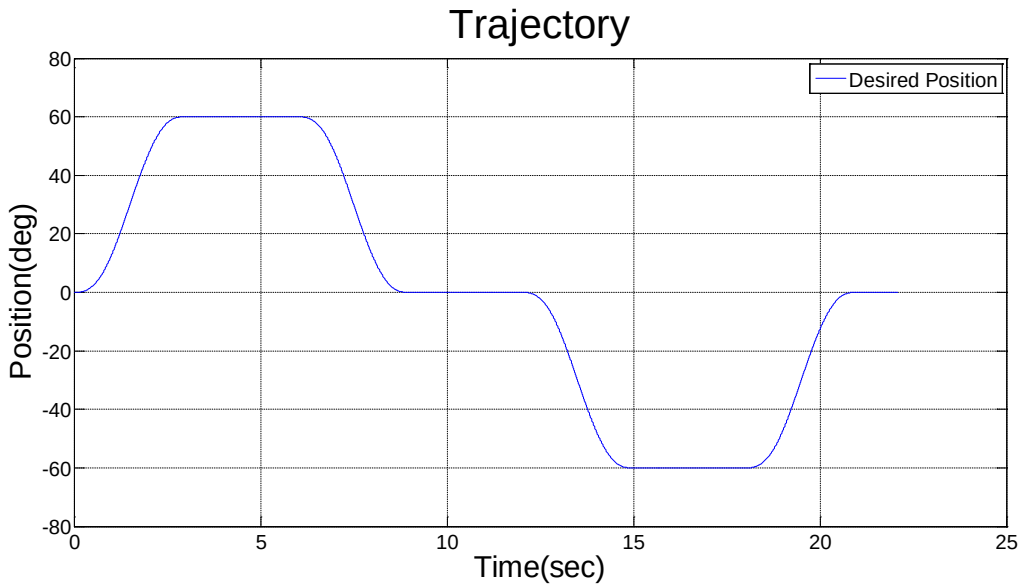

Figure 10. The trajectory of joint 1, 2, 3, and 4

3.1.2 Experiment Result

The TDC Tracking error of each joint is shown in the following Figure 12 ~ Figure 15. Figure 12 shows the result of joint 1. Tracking error of joint1 is ± 0.1 degree. Steady state error of joint 1, Linac is emitting radiation beam, is ± 0.01 degree. Figure 13 shows the result of joint 2. Tracking error of joint2 is ± 0.175 degree. Steady state error of joint 2 is 0.002 degree. Figure 14 shows the result of joint 3. Tracking error of joint 3 is ± 0.08 degree. Steady state error of joint 3 is ± 0.02 degree. Figure 15 shows the result of joint 4. Tracking error of joint4 is ± 0.08 degree. Steady state error of joint 4 is 0.002 degree. Run-time and sampling time were measured during 21 seconds, the new radiation robot moves. Figure 16 shows the 21000 of measured in run time. Maximum run time is 136.477μs (136,477ns). Minimum run time is 45.756μs (47,756ns). The average of run time is 89.022μs (89,022ns). Figure 17 shows the 21000 of measured sampling time. Maximum sampling time is 1.022814ms (1,022,814ns). Minimum sampling time is 0.980569ms (980,569ns). The average of sampling time is 1.000000ms (1,000,000ns). The latency of Sampling time is 0.003047ms (3047ns).

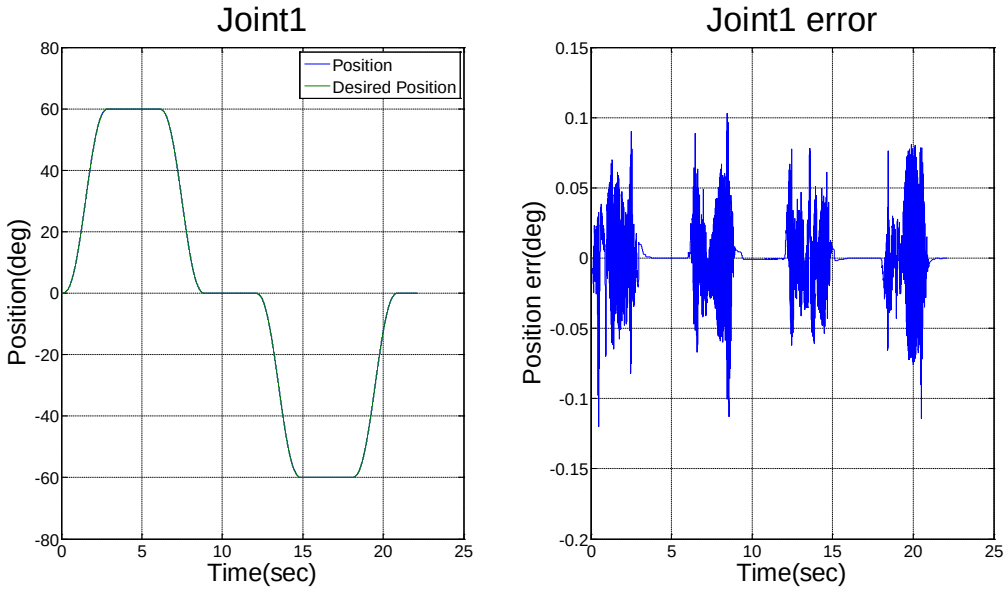

Figure 11. Joint1 trajectory and error

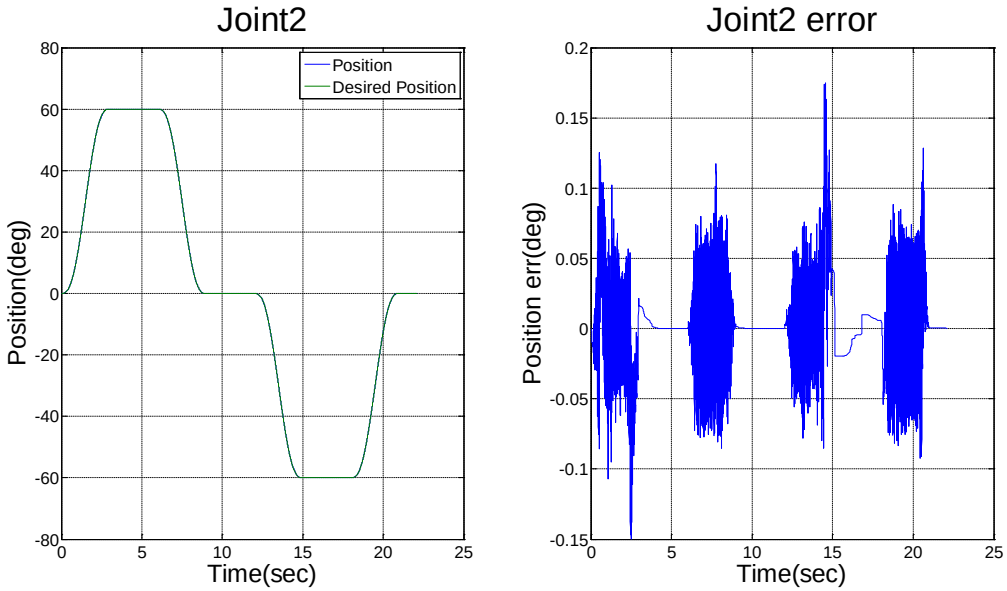

Figure 12. Joint2 trajectory and error

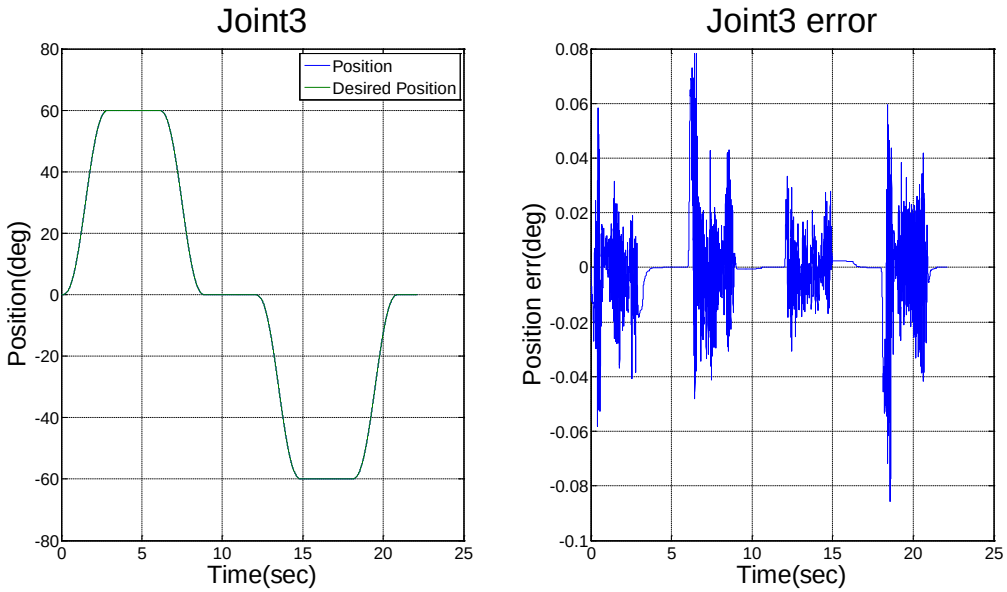

Figure 13. Joint3 trajectory and error

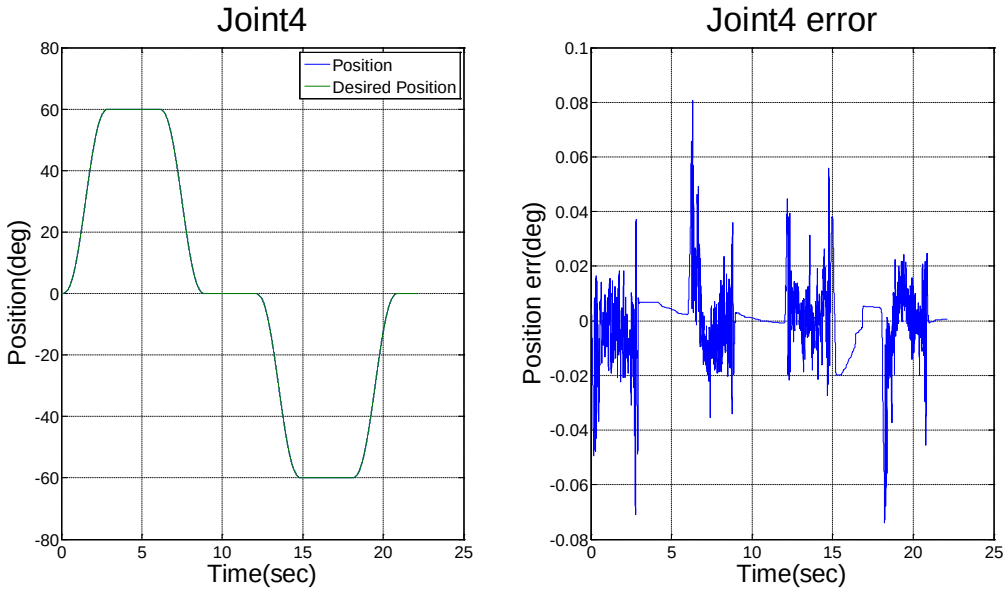

Figure 14. Joint4 trajectory and error

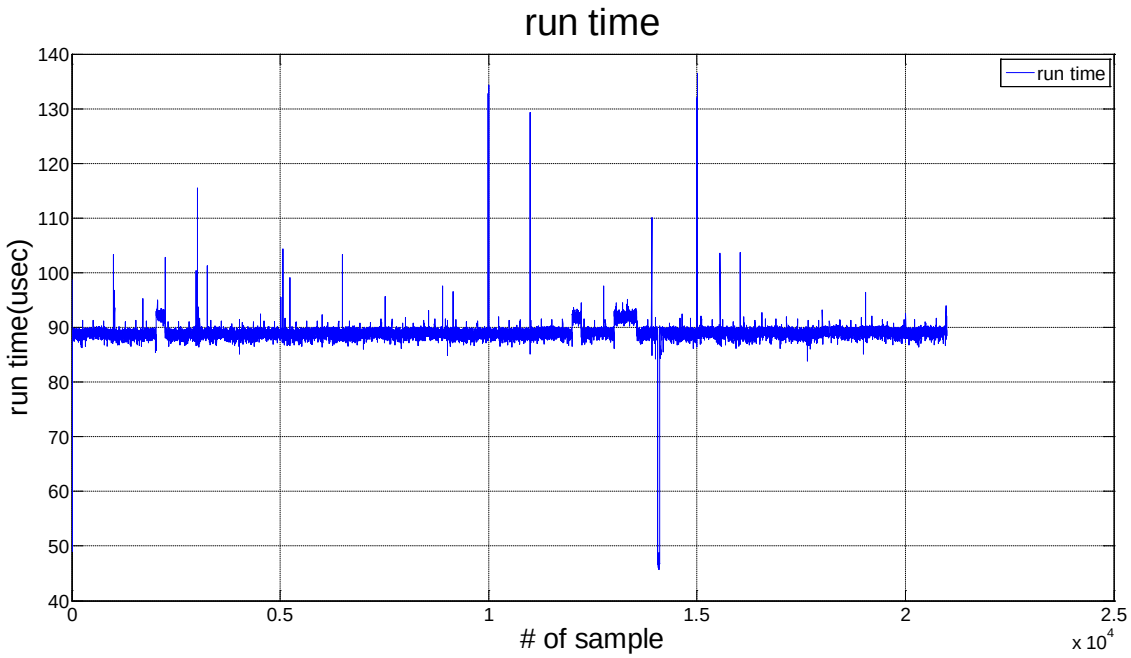

Figure 15. The result of run time

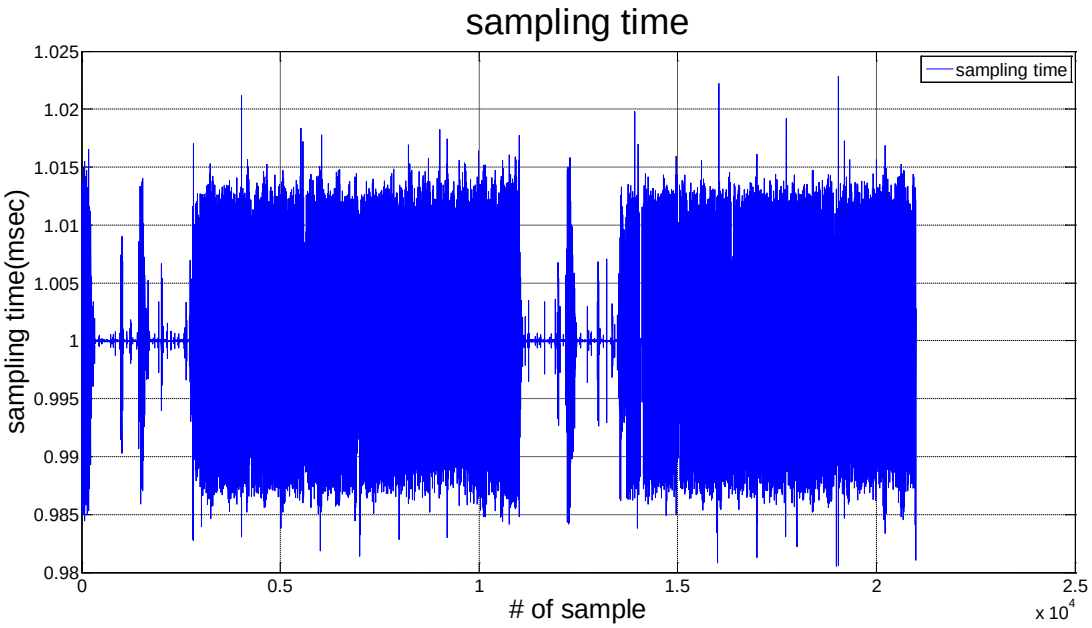

Figure 16. The result of sampling time

4. CONCULSION

4.1 Conclusion

In this paper, for the control of the new radiation robot, it has established a control environment using RTAI and applying a time delay controller for exact position control. The construction of control environment using RTAI should be considered that is variety of combinations of devices and software. So this paper was explained combination of RTOS that is a version of Linux, RTAI, and COMEDI and how to build RTOS. The new radiation robot was applied to TDC for accurate positioning. Accurate positioning is required for precision dose delivery. TDC estimates the unknown dynamics and disturbance. TDC has small computation. Through experimentation, check the control environment has error or not and tracking error of TDC. The tracking error of TDC is $\pm$0.2 degree and a steady-state error of TDC is $\pm$0.02. The average of 21000 measured sampling times in 21seconds is 1milli second. The control environment has to read encoder and calculate TDC and send control signal to motor in 1ms. The time of read encoder is 0.0005ms and maximum run time is 0.136ms and analog output time is 0.004ms. The sum of three times is 0.1405ms and this time is smaller than sampling time, 1ms.

The control environment works well and tracking error of TDC is 0.2 degree.

REFERENCES

[1] D. I. Thwaites and J. B. Tuohy, "Back to the future: the history and development of the clinical linear accelerator," *Physics in medicine and biology*, vol. 51, p. R343, 2006.

[2] A. Wu, G. Lindner, A. Maitz, A. Kalend, L. Lunsford, J. Flickinger, *et al.*, "Physics of gamma knife approach on convergent beams in stereotactic radiosurgery," *International Journal of Radiation Oncology* Biology* Physics*, vol. 18, pp. 941-949, 1990.

[3] J. Yang, J. P. Lamond, J. Feng, X. Wu, R. Lanciano, and L. W. Brady, "CyberKnife System," in *Stereotactic Body Radiation Therapy*, ed: Springer, 2012, pp. 37-52.

[4] J. R. Adler Jr, S. Chang, M. Murphy, J. Doty, P. Geis, and S. Hancock, "The Cyberknife: a frameless robotic system for radiosurgery," *Stereotactic and functional neurosurgery*, vol. 69, pp. 124-128, 1997.

[5] W. Atmadja, B. Christian, and L. Kristofel, "Real Time Operating System on embedded linux with ultrasonic sensor for mobile robot," in *Industrial Automation, Information and Communications Technology (IAICT), 2014 International Conference on*, 2014, pp. 22-25.

[6] P. Hambarde, R. Varma, and S. Jha, "The Survey of Real Time Operating System: RTOS," in *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*, 2014, pp. 34-39.

[7] L. Dozio and P. Mantegazza, "Linux Real Time Application Interface (RTAI) in low cost high performance motion control," *Motion Control*, vol. 2003, 2003.

[8] R. V. Aroca, G. Caurin, and S. Carlos–SP–Brasil, "A real time operating systems (rtos) comparison," in *Workshop de Sistemas Operacionais (Operating Systems)(WSO'2009)*, 2009.

[9] R. Morgan and U. Ozguner, "A decentralized variable structure control algorithm for robotic manipulators," *Robotics and Automation, IEEE Journal of*, vol. 1, pp. 57-65, 1985.

[10] K. Youcef-Toumi and O. Ito, "A time delay controller for systems with unknown dynamics," *Journal of dynamic systems, measurement, and control*, vol. 112, pp. 133-142, 1990.

[11] K. Youcef-Toumi and S.-T. Wu, "Input/output linearization using time delay control," *Journal of dynamic systems, measurement, and control*, vol. 114, pp. 10-19, 1992.

[12] T. Hsia and L. Gao, "Robot manipulator control using decentralized linear time-invariant time-delayed joint controllers," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 1990, pp. 2070-2075.

[13] S. Furr, "What is real time and why do i need it?," *QNX Software Systems Ltd*, 2002.

[14] https://www.rtai.org/

[15] RTAI 3.4 User Manual rev 0.3

요 약 문

새로운 방사선 로봇을 위한 제어환경 구축과 제어

본 논문은 프로토 타입의 새로운 방사선 로봇을 제어하기 위한 제어 환경을 구축하고 시간 지연 제어기 (TDC)를 이용하여 로봇을 제어해보았다. 새로운 방사선 로봇 디자인은 필자의 실험실에서 개발되었다. 이 로봇을 제어하기 위해서는 제어 환경을 구축하고 제어 법칙을 적용 시켜야 한다. 제어 환경은 Real time application interface(RTAI)를 이용하여 리눅스 기반의 운영체제를 실시간 제어가 가능하게 하였다. 실시간 제어가 가능한 운영체제는 엄격한 시간 제어가 가능하다. 엄격한 시간 제어는 정해진 시간에 계획된 기능들을 수행할 수 있다는 것을 의미한다. 엄격한 시간 제어가 가능한 제어 환경에서 시간 지연 제어기를 이용하였다. 시간 지연 제어기는 플랜트의 알지 못하는 동역학 정보나 외란 등을 시간 지연을 이용하여 값을 추정하는 제어기이다. 이 제어기는 구조가 간단하고 필요한 계산량이 적다. 새로운 방사선 로봇을 위한 제어 환경이 잘 구축 되었는지 와 시간 지연 제어기의 tracking error 를 확인 하기 위하여 실험을 실시하였다. 그 결과, RTAI 를 이용한 실시간 제어가 가능한 운영체제에서 취득한 21,000 개의 데이터의 평균 샘플링 타임은 1ms 이었다. 그리고 시간 지연 제어기의 tracking error 는 ± 0.2 도 이하였다. 따라서, 새로운 방사선 로봇을 위한 실시간 제어환경을 구축하였고 시간 지연 제어기를 적용하여 정확한 위치 제어가 가능함을 보였다.

핵심어: 새로운 방사선 로봇, Real time application interface(RTAI), 시간 지연 제어기(TDC)