Platform Independent Integrated Environment for Simulation and Real-Time Control Experiment

著者	Yano Kentaro, Koga Masanobu
journal or publication title	2006 SICE-ICASE international joint conference
page range	411-416
year	2007-02-26
URL	http://hdl.handle.net/10228/00006397

doi: info:doi/10.1109/SICE.2006.315332

Platform Independent Integrated Environment for Simulation and Real-Time Control Experiment

Kentaro Yano[1] and Masanobu Koga[1]

[1]Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka, Japan
(Tel : +81-948-29-7726; E-mail: yano@mk.ces.kyutech.ac.jp, koga@ces.kyutech.ac.jp)

Abstract: This paper presents the methods which make a design process of control system easy. The Real-Time control framework makes writing Real-Time control programs easy. The framework reduces the dependency on the platform, such as OS and hardware, by using not only C but also Java. Simulation programs are transformed into Java programs, and that can execute platform independent simulation. The idea of the object model is used for the transformation of the codes. Real-Time control programs are generated from simulation program automatically. Factory Method pattern is used for automatic generation of Real-Time control programs. We develop a platform independent integrated environment for the simulation and the Real-Time control using above methods. The integrated environment has the facility of remote control by using distributed objects.

Keywords: Real-Time control, Framework, Object model, Factory Method pattern, Distributed object

1. INTRODUCTION

A design process of control system is generally executed in order of modeling, design of controller, simulation, and control experiment. If a control plant is a robot or an inverted pendulum etc, Real-Time control is required and the control experiment programs should be Real-Time programs(RT programs). The RT programs are the programs which assure the timing of process beginnig and the time limit of process end[2]. And, the control experiment program which is an RT control program is called an RT control program. It is hard to write an RT control program, because special knowledge like scheduling is needed.

An RT control program is often written by using the library provided by the Real-Time OS(RTOS) like RT-Linux[1] [2]. However, if an library is used it is necessary to find the parts which should be changed from whole of the program when control plant changes, because the parts are dispersed at whole of the program. Also, there is a high possibility that the miss which forget the partial change etc. get mixed in with the program. Therefore, the writing of an RT control program is not easy and it is hard to execute efficiently.

A simulation program is often written by a numerical computation language which makes it easy to write mathematical formula[3][4]. After affirmation of the results of simulation, an RT control program is newly written by using the library for RT control programs. Therefore, it is impossible to execute design process of control system efficiently because individually creation of simulation program and RT control program is needed and smoothly change from simulation to control experiment is impossible.

To solve this problem, the methods of creation of RT control program using the information written at the simulation are proposed. For example, RTMATX creates[5] RT programs by edit the function written by MaTX[3]. However, this method has the two problems, one is that edit of program by hand is needed, and other problem is that program depends on platform. Then, the method which automatically generates RT control program from simulation program is also proposed. For example, Real-Time Workshop(RTW)[6] generates RT control program written by C language from Matlba/Simulink. Therefore, change from simulation to control experiment is easy by using automatic generation of RT control program from simulation program.

A simulation is run on general OS like Windows, but an RT control is often run on RTOS. And, the platform(hardware or OS) for simulation and the platform for experiment are often different. Then, it is necessary to deploy the RT control program which is automatically generated from a simulation program on the machine which executes control experiment. Since simulation and control experiment are executed repeatedly, it is necessary to repeatedly change the simulation program, change the RT control program, and deploy it on the experiment machine. Therefore, a design process of control system cannot execute efficiently, since simulation and control experiment are not executed in same platform.

To solve the problem of deployment of an RT control program, the method which uses a Web browser is proposed[7]. In this method, automatic generation and control experiment are able to execute by only selecting the simulation program which is written by Simulink on Web browser. However, operability isn't good and integrated execution of simulation and control experiment cannot execute because simulation and control experiment are executed by another program like Simulink and Web browser.

The purpose of this paper is to propose the methods to efficiently execute a design process of control system which includes control experiment. First, we propose an RT control framework[8] which makes writing RT control programs easy. Second, we propose the separation of platform dependent parts from programs by using a platform independent language. And, we propose the automatic generation of RT control program using by design pattern[9].

2. RT CONTROL FRAMEWORK

An RT control framework provides a frame of the creation of RT control programs. It is possible to create an RT control program by only creating the specific parts of control system by using an RT control program framework. It makes writing RT control programs easy, and it is possible to raise the efficiency of writing RT control programs.

2.1 Framework

A framework provides a frame of the whole software[8]. Based on a framework, it is possible to complete a final product efficiently by the creation where developer corresponds to individual purposes. So, a framework is provided as a semifinished product. And, the parts which are changed or added by developers are called hotspots.

2.1.1 Difference between framework and library

A framework is a type of a library, but the program which is in the library is only used. Programs which are created by using a framework are called and used by the framework(Fig. 1).

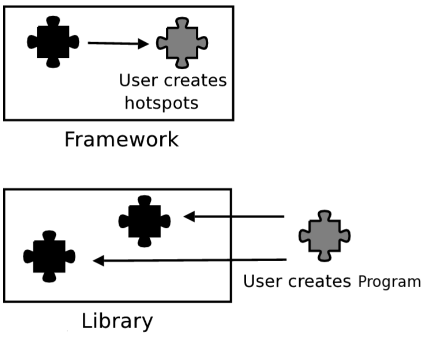

Fig. 1 Framework and Library

If a framework is used for writing an RT control program, it is possible to create an RT control program by only creation of the parts which is inherent parts(hotspots) of control plant.

2.2 Architecture of the framework

The architecture of the framework which we propose is shown in Fig. 2, and user can create an RT control program by only creating the parts which are interpret of commands, display of data, and Real-Time task.

3. SEPARATION OF PLATFORM DEPENDENT PARTS

To separate the platform dependent parts of RT control programs, this paper proposes that using both the language adapted for the platform and the platform independent language. Platform independent parts are written by platform independent language in this method. This paper also proposes the method which transforms a simulation program written by numerical computation language to the program written by platform independent language.

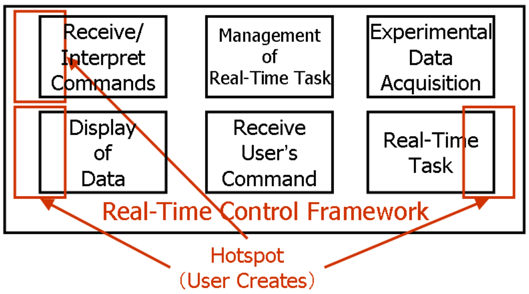

Fig. 2 Architecture offramework

The object model are used when transform a simulation program. It is possible to execute simulation and control experiment on the same platform by these separations of platform dependent parts.

3.1 Separation of platform dependent parts of hotspots

An RT control program are divided into two parts. The parts which need Real-Time processing depend on platform. The other parts are independent from platform because those don't need Real-Time processing. If all parts are written by platform dependent language, whole of an RT control program depend a platform.

Then, this paper proposes that developers write hotspots which depend on platform in the language adapted for the platform, and the other parts are written in the platform independent language. As mentioned above, platform dependent parts are separated by using both platform adapted language and platform independent laguage. The detail of parts written by platform adapted language and platform independent language are shown by Table 1.

Table 1 Separation of platform dependent parts

Platform dependent pars	Platform -independent parts
Real-Time Task	Receive and interpret data from user
Initial setting of Real-Time Task	Display of data

3.2 Transformation of simulation program

The Simulation program written by numerical computation language are transformed into platform independent program. It is possible to execute platform independent simulation by transformation of simulation program. Then, this paper proposes the transformation of simulation program into platform independent program using object model.

3.2.1 Object Model

The object model of a programming language is a set of classes to create the objects which have same information of a source code that was written by the language.

By creating the object model of a source code as shown in Fig. 3, it is possible to create the object tree which has the same meaning of the source code. It is possible to transform to other languages by pursuing the created object tree and calling the methods of creating other language which are written at nodes.

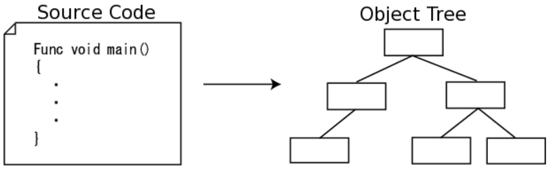

Fig. 3 Object Model

3.2.2 Transformation of simulation program using object model

Figure 4 shows the outline of transformation of simulation program by using object model.

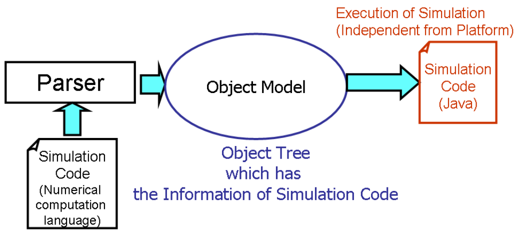

Fig. 4 Transform of simulation program using Object Model

When simulation program are transformed, first the Parser receives the source code written by numerical computation language. The Parser construes the source code, and creates the object tree based on the syntax. The object tree which has the information of the source code written by numerical computation language can be obtained by the method above. Then, it is possible to generate the platform independent source code from the created object tree. It is possible to execute platform independent simulation by transformation of the simulation program into platform independent program by using object model.

4. AUTOMATIC GENERATION OF RT PROGRAM

This paper proposes the automatic generation of an RT program from a simulation program by using object model and the Factory Method pattern[9]. By automatic generation of an RT program, if a simulation program is created, it is possible to execute simulation and control experiment seamless.

4.1 Factory Method Pattern

The Factory Method pattern is a design pattern which decides how to create an instance of a class is defined in a super class, and the concrete statement for the instantiation is defined in each subclass. Figure 5 shows the class diagram of Factory Method Pattern, it defines that Creator creates Product. And ConcreteCreator which is a subclass of Creator creates ConcreteProduct which is created instance. Then, the class of instantiation frame and the class of a real instance generation are separated.

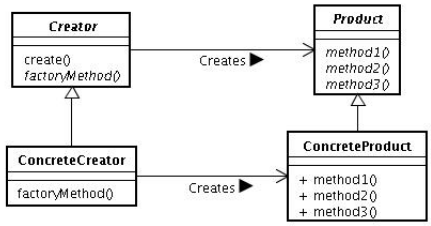

Fig. 5 Class Diagram of Factory Method Pattern

By using the Factory Method Pattern, it is possible to switch objects which are embedded in an object tree. If objects which generate an RT program are embedded when an object tree is created, it is possible to generate RT program automatically.

4.2 Automatic generation of RT program using Factory Method Pattern

Figure 6 shows the outline of automatic generation of RT control program using Factory Method Pattern, and Factory Method Pattern are applied where Parser creats object model.

Creator defines the creation method of an object model for Parser, and SimCreator creates objects which transform simulation program into platform independent language. RTCreator creates the object model which generates an RT program when an RT program is generated. Then, transformation of simulation program and automatic generation of RT program are able to execute by using object model of same structure. It is possible to add the kind of code generation, if a subclass of Creator and classes which are embedded in object tree is created.

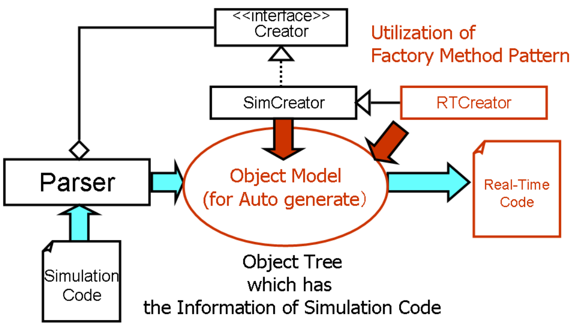

Fig. 6 Automatic generation of Real-Time contorl program using Factory Method Pattern

5. IMPLEMENTATION OF THE PROPOSED METHOD

By using the proposed method, RT control program for RT control program framework is automatically generated from numerical computation language. And, simulation and control experiment are executed at same machine by transformation of program written in numerical computation language to platform independent language. If only simulation program are created, it is possible to integrate simulation and control experiment by using the proposed method. Then, it is possible to raise the efficiency of a design process of control system.

This paper implements the proposed method using the C language and the Java language. We describe the example of the implementation in the following.

Figure 7 shows the architecture of the developed integrated environment, and RT control framework runs on RTOS and Java VM. Hotspots of the framework are divided into the parts which use the function of an RTOS and the parts which use the function of the Java VM. It is able to execute simulation, automatic generation of RT control program, and control experiment by using GUI which runs on the framework.

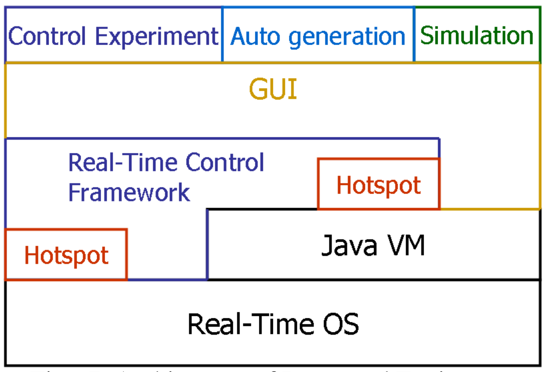

Fig. 7 Architectrue of Integrated Environment

5.1 RT control framework

We developed ReTiCoF(Real Time Control Framework)[10] which is the implementation of the Real-Time control framework proposed by this paper. ReTiCoF runs on RT-Linux, and it is implemented by the C language and the Java language.

We describe the architecture of MK-Task which is the execution parts of Real-Time processings in ReTiCoF. MK-Task is implemented by the C language, and use the function of RT-Linux.

The architecture of MK-Task when used by the Java by using JNI(Java Native Interface)[11] is shown in Fig. 8. JNI is the function to use a native code like the C language from the Java.

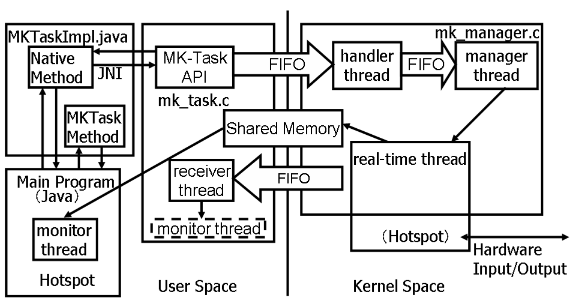

Fig. 8 Architecture of MK-Task

MK-Task consists of mk_task.c which runs on User space, and mk_manager.c which runs on Kernel space. mk_task.c consists of the APIs of MK-Task, and receiver thread which receive data from Kernel space. mk_manager.c has the handler thread which runs when a call from User space is sent, manager thread which manages an Real-time task, and Real-Time task as thread which has the Real-Time constraint. And MK-TaskImpl.java has native methods which call the APIs of MK-Task using JNI, and methods which are needed when MK-Task is used from Java.

5.2 Separation of platform dependent parts

We use the C language for the environment adapted language and the Java language for environment independent laguage. We use JNI to use both the C language and the Java language.

As an example of the transformation of simulation program by using the object model, we implement the transformation of MaTX programs to Java programs by using matj[12]. matj is a tool which converts MaTX programs into Java programs.

5.3 Automatic generation of Real-Time Program

We apply the Factory Method pattern to matj, and it is able to automatically generate RT control program from simulation program written in MaTX.

5.4 Advantage of using Java

The implemented integrated environment can execute remote control using distributed objects and local and remote compatible GUI is available as the advantages of using Java.

5.4.1 Remote control using distributed objects

It is possible to automatically deploy an RT control program by using remote processing.

The distributed object is an object which is able to call methods through network. RMI(Remote Method Invocation)[13] is the technology which implements distributed objects in Java.

Figure 9 shows the remote control system using Real-Time distributed objects. The server machine is installed RT-Linux, and can execute RT control program. Client machines are connected to server machine via network.

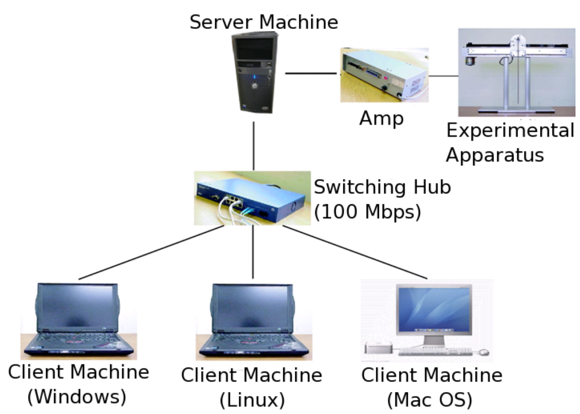

Fig. 9 Remote control system using Real-Time distributed objects

The users of the integrated environment execute simulation and Real-Time control using the integrated environment on the client machine. The client machine doesn't need the environment which runs Real-Time program, and it needs only the execution environment of the Java. So, the system is independent from platform. And, the problem of deploy is solved by sending the RT program from integrated environment on the client machine using RMI. Therefore, it is possible to integrate execution of simulation and Real-Time (remote) control.

5.4.2 Local and remote compatible GUI

Local control is executed on only the server machine. When local control is executed, the users of the integrated environment can use same GUI which executes remote control. Figure 10 shows the GUI which is provided by the integrated environment. The GUI has the menu and toolbar which are related to simulation, automatic generation of RT program, and control experiment. And it has also the parts of display of experiment result, property of Real-Time task, and messages for users.

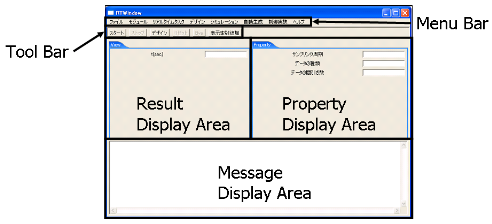

Fig. 10 Start-up screen of GUI

The Proxy pattern is used for the mechanism of local and remote compatible GUI. Figure 11 shows the class diagram of the Proxy pattern. The Proxy pattern is the design pattern which makes the Proxy class as the proxy of the RealSubject class which is the real actor, and it handles the process of the RealSubject class as much as possible.

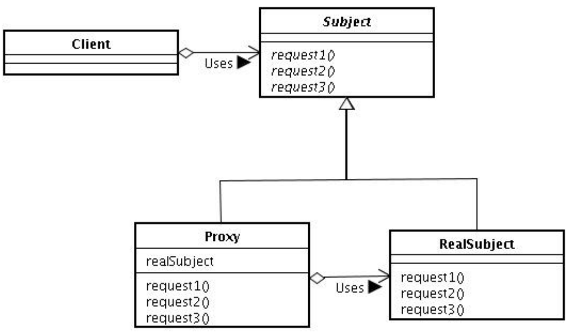

Fig. 11 Class Diagram of Proxy Pattern

The class diagram of Real-Time control system is shown as Fig. 12, and the mechanism of the local and remote compatible GUI are described below by using it. The GUI which is shown by Fig. 10 is provided by the RTWindow class. The RTWindow class has the classes which provide the parts of the GUI like the ViewPanel class and the class which implements MKTask interface which execute Real-Time control. If the user executes Real-Time control on the local machine, the MKTaskImpl class are used. If the user executes remote control, the RemoteMKTaskProxy class is used. RemoteMKtaskImpl which is on the server machine handles the remote control processes. The integrated environment uses the Proxy pattern, and uses the RemoteMKTaskProxy class which is the Proxy instead of RemoteMKTaskImpl which is the RealSubject. Therefore, the GUI is local and remote compatible GUI, because the class used by local control and the class used by remote control implements same interface.

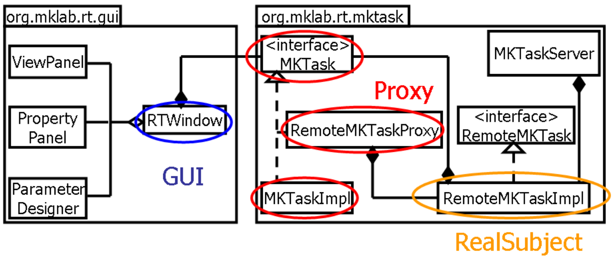

Fig. 12 Class Diagram of Real-Time control system

6. EXPERIMENT OF SWINGING UP A PENDULUM

To verify the availability of the method proposed by this paper, we execute the experiment of swinging up a pendulum[14] using developed integrated environment.

Simulation is executed by using the simulation program written in MATX. At this time, user writes the function of differential equation and the function which has information of controller. Figure. 13 shows the RTWindow which shows the results of simulation.

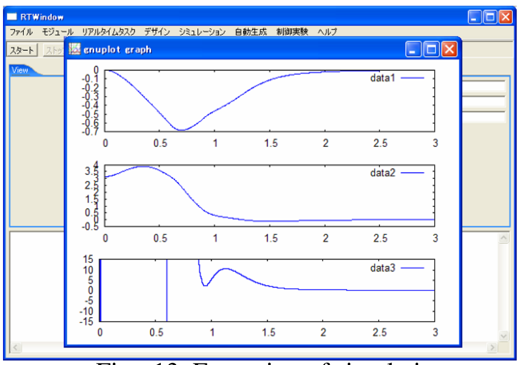

Fig. 13 Execution of simulation

The automatic generation of Real-Time program is executed by selecting the simulation program, and inputting the necessary information. After the above information is inputted, the Real-Time program is generated automatically. And the variable displayed on the GUI is added, the display of the GUI is changed for a pendulum.

An experiment is executed, after the setup of the generated RT control program, the module of the experimental apparatus, the parameter of the controller. Figure 14 shows the screen of the RTWindow which is executing a control experiment. Figure 15 shows the experiments results of swinging up a pendulm. The sampling time of the experiment is 5[ms]. As shown in Fig. 15, the angle of the pendulum is close to the 0[rad] after 4 seconds of the beginnig of the experiment, the experiment of swinging up a pendulum become successful.

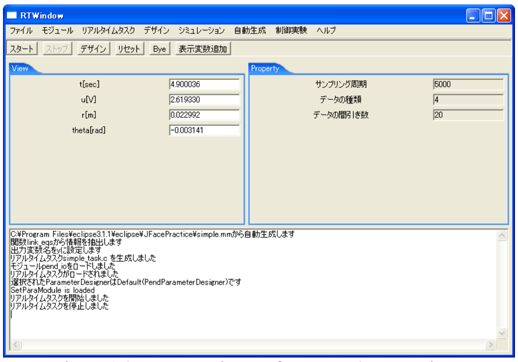

Fig. 14 Execution of control experiment

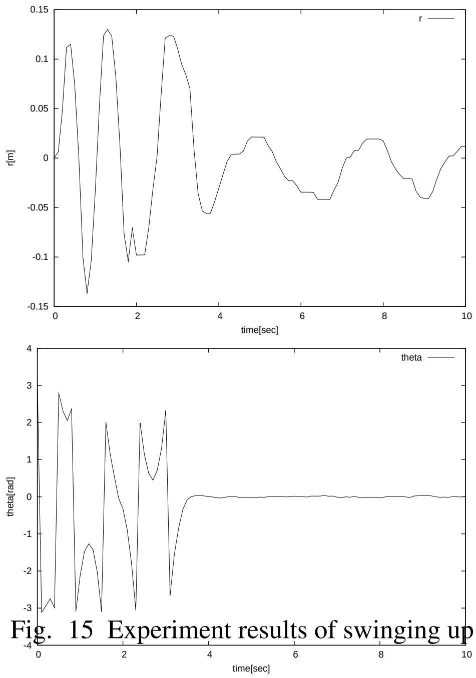

Fig. 15 Experiment results of swinging up a pendulum

7. CONCLUSIONS

This paper proposed the methods which make a design process of control system easy. This paper proposed the RT control framework, separation of platform dependent parts, automatic generation of RT program. By using the method proposed by this paper, simulation and Real-Time control experiment are able to integrate. We implemented the proposed methods, and availability was shown by the swinging up a pendulum.

REFERENCES

[1] RTLinux. http://www.fsmlabs.com/products.html

[2] Michitaka Funaki and Shouka Ra, "Guide book for Real-Time sensing and control by Linux," Shuwa system, 1999.

[3] Masanobu Koga. "MaTX for control/numerical analysis," Tokyo Denki University Press, 2000.

[4] The MathWorks Matlab. http://www.mathworks.com/products/matlab/

[5] Masanobu Koga, Hiroaki Toriumi, and Mitsuji Sampei "An integrated software environment for the design and real-time implementation of control systems," Control Engineering Practice, Vol. 6, pp.1287-1293, 1998.

[6] The MathWorks Real-Time Workshop. http://www.mathworks.com/products/rtw/

[7] Michele Basso and Gianni Bangi, "ARTIST:A Real-Time Interactive Simulink-based Telelab," Proceedings of the 2004 IEEE Conference on Computer Aided Control Systems Design pp.196-201, 2004.

[8] Teppei Nakano, "Understanding the framework," Java WORLD, Vol. 6, No. 62, pp. 54-67, 2002.

[9] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesle Pub, 1995.

[10] Masanobu Koga and Kentaro Yano, "Automatic generation of Real−Time code for remote control system," FIT2004 3rd Forum on Infrmation Technology, pp. 433-434, 2004.

[11] Rob Gordon, "Essential JNI: Java Native Interface," Prentice Hall Ptr, 1998.

[12] Takeshi Matsuki, "Development of Java Numerical Computing Engine based on Objectmodel of Programming Language," Master's thesis, Kyushu Institute of Techonology, 2005.

[13] Tatsuo Ueta, "Java's Karakuri 2(practive version)," Ltd. IDG Japan, 2001.

[14] Astrom and K.Furuta. "Swinging up a pendulum by energy control," *Automatica*, Vol. 36, pp. 287-295, 2000.