Integrated Environment of Simulation and Real-Time Control Experiment for Control system

Kentaro Yano
Graduate School of Computer Science and Systems Engineering
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka
JAPAN
yano@mk.ces.kyutech.ac.jp

Masanobu Koga
Faculty of Computer Science and Systems Engineering
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka
JAPAN
koga@ces.kyutech.ac.jp

Abstract — This paper presents the methods which make a design process of control system efficient. The Real-Time control framework makes writing Real-Time control programs easy. Real-Time control programs are generated from simulation programs automatically. The idea of an object model and the Factory Method pattern are used for automatic generation of Real-Time control program. Separation of platform dependent parts makes non Real-Time parts can be independent from platform. We develop a platform integrated environment of the simulation and the Real-Time control for Control system using above methods.

I. 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 control experiment programs should be Real-Time programs(RT programs). An RT program is the program which assures the time limit of process beginning and process completion [1]. And, the control experiment program which is an RT control program is called an RT control program.

An RT control program is often written by using a library provided by an Real-Time OS(RTOS) like RT-Linux [2] [1]. At this time, it is necessary to find the parts which should be altered by the change of control plant from whole of the program. Since, the target-depend parts are scattering at large range of the program. Also, there is a high possibility that the miss which forget the partial change etc. get mixed in with the program.

A simulation is run to confirm the performance of controller, and 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. Therefore, it is impossible to execute design process of control system efficiently, because an individually creation of a simulation program and an RT control program is needed and smoothly change from simulation to control experiment is impossible.

To solve this problem, methods which create RT control programs using the information written at simulation programs are proposed. For example, by using R_TMqTX [5], it is able to create RT control programs by edit the function written in M_qTX [4] which is an numerical computation language. And, Real-Time Workshop(RTW) [6] generates RT control programs written in C language from block diagrams of Simulink [7]. A method which improves RTW to industrial applications and generates iFix [8] etc. from Matlab codes is also proposed [9].

A simulation is run on a common OS like Windows, but an RT control is often run on an RTOS. And, the platform(hardware or OS) for a simulation and the platform for an 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 a simulation and a control experiment are executed repeatedly, it is necessary to repeatedly change a simulation program, create an RT control program, and deploy it on an experiment machine.

To reduce a workload of a deployment of an RT control program, a method which uses a Web browser is proposed [10]. In this method, users can execute an automatic generation of an RT control program and control experiment by only selecting the simulation program written in Simulink on Web browser. However, simulation is executed by Simulink, and RT control is run by Web browser, so because of the difference of interface, users are easy to be confused by the operations. And, HTML client which is an application that runs on Web browser and uses UI parts provided by HTML, has problems like it is impossible to provide rich functions because the UI is poorer than an UI of common program.

The purpose of this paper is to propose methods to solve the issues and make it possible to efficiently execute the iteration of simulation and Real-Time control at a design process of control system. And, this paper describes about the integrated environment for simulation and Real-Time control which is the implementation of the proposed methods. It is able to automatically generate an RT control program from a block diagram of Jamox [11]. Jamox can run modeling and simulation, so it is possible to execute all parts of the design process of control system by using Jamox and the integrated environment.
In Section 2, Jamox is explained. In Section 3, we propose methods for the issues. In Section 4, developed integrated environment to actualize proposed methods is described. In Section 5, an illustrative example which is a stabilization control of inverted pendulum is explained. Finally, in section 6, conclusions are described.

II. JAMOX

Jamox(Java Agile MOdeling Tool for Control System) is a modeling and simulation tool for control system using an adjacency matrix, and it supports a linear system and nonlinear system modeled by continuous time, discrete time, and sampled data system(which is a mix of continuous time and discrete time).

An adjacency matrix is one of an expressive form of a graph used for graph theory. It is easy to simulate a control system represented by a block diagram drawn by Jamox. Jamox not only can simulate time response of system, but also can compute a state space representation of all system and frequency response(Bode diagram) easily, if that system is a linear.

Fig. 1 shows an execution screen of Jamox. It is able to allocate blocks(provided by the tool) by drag and drop from the library to the canvas. Each block is able to be connected by mouse action, and it is easy to create a block diagram on the screen displays.

Fig. 1. Modeling and Simulation tool Jamox

1) Modeling: The block ① shown in Fig. 1 is a basic block registered by the library. Kinds or structure of the models are defined, and only parameter is customizable. The block ② is a customer-defined block which represents any dynamic systems and static systems. And it is related with a Java class which extends DynamicSystem class or StaticSystem class. It is able to create high reusable blocks efficiently based on grouping of control system and use of inheritance of class. And, it is easy to create fast computation code using NFC [12] which is a numerical computation language. The block ③ is a block which treats a system written in MATLAB syntax, and it is possible to reuse existing MATLAB codes.

2) Simulation computation: In simulation calculation, solution of differential equation is solved by RungeKutta class etc. which implements OdeSolver interface provided by Ode package of NFC. If an algebraic loop exists, a solution of a nonlinear simultaneous equation is solved by using Newton-Raphson method.

III. INTEGRATE METHODS OF SIMULAITON AND REAL-TIME CONTROL

A. RT Control Framework

In this section, we propose an RT control framework which provides a frame of the creation of RT control programs for target-dependent parts is scattering at large range of RT program.

1) Framework: In software engineering, a framework is used commonly in the sense of technique which provides a frame of the whole software [13]. 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 semi finished product. The parts which are changed or added by developers are called hotspots, and other parts are called frozen spots.

As shown in Fig. 2, a program written by users using a library only calls and uses codes in a library. On the other hand, in a framework, it calls and uses codes developed by users based on a specification of a framework.

Fig. 2. Framework and Library

2) Architecture: The architecture of the framework which we propose is shown in Fig. 3.

An RT control framework provides a frame of the creation of RT control programs. Target-dependent parts become apparent and it is able to control centrally by using an RT control framework. It is also possible to create an RT control program by only creating the target-dependent parts. Therefore, it makes writing RT control programs easy, and it is possible to raise the efficiency.

It is possible to write an RT control program only creating pieces of Real-Time task, experimental data viewer, command interpretation

B. Automatic generation of RT control program

This paper proposes a method that automatically generates an RT control program from a simulation program using an object model and the Factory Method pattern [14]. It is possible to raise the efficiency of the design process of control system.
1) Object Model: An object model is a data structure which is an object-oriented model of data used by a program, and it is an interface of given operations for that. An object model of a programming language is a set of classes to create the objects which have the same information of a source code that was written by the language. Fig. 4 shows the schematic of object modeling of a source code as an object tree which has the same information of the source code. Let a function which generates an RT control code be defined at each node. Then, it is possible to generate an RT control program by pursuing the object tree and calling the defined functions written at each node.

2) Automatic generation of program using object model: Fig. 5 shows the outline of automatic generation of an RT control program using object model, and the Factory Method Pattern are applied where Parser creates object model. And the Factory Method Pattern is applied where Parser creates an object model. The Factory Method pattern is a design pattern which decides how to create an instance of a class defined at a super class, and the concrete creation of instance is handled by each subclass [14]. The Factory Method Pattern defines that Creator creates Product. And ConcreteProduct which is a real instance is created by ConcreteCreator which is a subclass of Creator. Then, the class provides a frame of instantiation and the class of a real instance generation are separated.

In Fig. 5, Creator creates an object model for Parser(Product). And RTCreator(ConcreteCreator) creates objects(ConcreteProduct) which transform a simulation program into an RT control program for RT control framework. By using RT control framework, generated Real-Time program becomes simple and easy. And, by using the Factory Method pattern, it is possible to switch objects which are embedded in an object tree. If an implementation class of Creator and classes which are embedded in object tree are created, it is possible to add another kind of RT control program.

C. Separation of platform dependent parts
To solve the issue of RT control programs depend on platform, this paper proposes a method which separates platform dependent parts and independent parts.

1) Separation of platform dependent parts of RT control program: In general, an RT control program consists two parts. One is a Real-Time part which needs Real-Time processing, and another part is a non Real-Time part which doesn’t need Real-Time processing. If a Real-Time part is written in a platform correspond language, and a non Real-Time part is written in platform independent language, a non Real-Time part can be independent from platform. The outline about separation of platform dependent parts is shown by Fig. 6.

2) Server and client system: This section describes an improvement of portability of programs by combining of a method mentioned above section and server and client system. Fig. 7 shows the schematic of server and client system. A plant is connected to a server machine. A server machine is installed RTOS, and it can execute RT control programs. A client machine is connected to a server machine via network.

In this system, Real-Time processing is executed by only a server machine, and a client machine runs only non Real-Time
processing. And, non Real-Time information is sent between a server machine and a client machine. So, a client machine doesn’t need the special environment for Real-Time programs. And, a client machine doesn’t depend on particular OS, since non Real-Time processing parts are platform independent. So, combining of separation of platform independent parts and server and client system makes improve of portability of program.

And, if a mechanism which automatically send an RT control program from a client machine to a server machine is implemented, an issue about deploy of an RT program is solved.

IV. IMPLEMENTATION OF THE PROPOSED METHOD

We implemented proposed methods, and developed the integrated environment for simulation and Real-Time control experiment using the C language and the Java language. Fig. 8 shows the schematic of the integrated environment. An RT control program using an RT control framework is generated automatically from a simulation program by the integrated environment. The integrated environment can raise the efficiency of the design process of the control system, since users can run integrated simulation and Real-Time control experiment.

Fig. 9 shows the architecture of the developed integrated environment. The RT control framework runs on an RTOS and the Java VM. The framework has two hotspots. Hotspot(1) use the function of an RTOS, and hotspots(2) use the function of the Java VM. It is able to execute a simulation and a control experiment by using GUI which runs on the framework. And automatic code generation of an RT control program is executed between a simulation and a control experiment.

A. RT control framework

We developed ReTiCoF(Real Time Control Framework) [15] which is the implementation of a Real-Time control framework proposed by this paper. ReTiCoF runs on RT-Linux [2] which is one of the RTOS, and it is implemented by the C language and the Java language.

The architecture of MK-Task [16] which is the execution parts of Real-Time processing in ReTiCoF is described here. MK-Task is implemented by the C language, and it provides mechanisms for easy use of functions of RT-Linux.

The architecture of MK-Task is shown in Fig. 10.

MK-Task consists of mk_task.c which runs on User space, and mk_manager.c which runs on Kernel space. mk_task.c has APIs of MK-Task, and codes of 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 MKTaskImpl.java has native methods which call APIs of MK-Task using JNI, and methods which are needed when MK-Task is used from Java. JNI(Java Native Interface) [17] is the technique to use a native code like the C language from the Java.

B. Automatic generation of RT control program

The developed integrated environment can generate an RT control program(hotspots of the RT control framework) from simulation program written in MqX [4]. The integrated environment creates an object model of MqX from source code of MqX using matj [18] which is an interpreter of
MATX implemented by the Java. The Factory Method pattern is applied to the Parser of matj. And, we created classes which generates a Real-Time processing code (matrix calculations, trigonometric functions etc.) and, a class which embeds those classes to an object tree. Then, it is possible to generate an RT control program by pursuing the created object model.

The integrated environment is also able to generate an RT control program from a block diagram created by Jamox [11]. The block diagram which is supported by the integrated environment has a controller created by a block of MATX or cascade connections of linear systems.

C. Separation of platform dependent parts

We use the C language for the platform correspond language and the Java language for platform independent language. We use JNI to use the C language and the Java language together.

In this integrated environment, server and client system is implemented by using RMI (Remote Method Invocation) [19] which is the implementation of distributed objects in the Java. The distributed object is an object which is able to call methods through network. And, we implement a mechanism which sends an RT control program from a client machine to server machine using RMI.

D. Local and remote compatible GUI

Fig. 11 shows the GUI provided by the integrated environment. It has a menu bar and a toolbar which are related to simulation, automatic generation, and control experiment. It has also display part of experimental data, a property of Real-Time task, and messages for user.

![GUI of integrated environment](image)

In this implemented integrated environment, it is possible to use same GUI between a local control using only a server machine, and a remote control using a server machine and a client machine. The Proxy pattern [14] is used for the mechanism of local and remote compatible GUI. The Proxy pattern is the design pattern which makes Proxy class as the proxy of RealSubject class which is the real actor, and it handles processes of RealSubject class as much as possible.

Fig. 12 shows the class diagram of the integrated environment using the Proxy pattern. The GUI which is shown by Fig. 11 is provided by RTWindow class. RTWindow class has the classes which provide the GUI parts like ViewPanel class, and the implementation class of MKTask interface which executes Real-Time control. MKTaskImpl class is used for local control, RemoteMKTaskProxy(Proxy) class is used for remote control instead of RemoteMKTaskImpl(RealSubject).

![Class Diagram of Real-Time control system](image)

V. EXPERIMENT OF STABILIZATION CONTROL OF AN INVERTED PENDULUM

To verify the availability of the method proposed by this paper, we execute the experiment of stabilization control of a pendulum using developed integrated environment.

First, modeling a pendulum using Jamox and design a controller using MATX. Fig. 13 shows the result of modeling and design of controller.

![Modeling of pendulum using Jamox](image)

Simulation is executed by using Jamox. At this time, a user selects a solver of differential equation, and input a start/end time of simulation, and initial state. Fig. 14 shows the results of simulation.

![Result of simulation](image)
In the automatic generation of Real-Time program, user selects a simulation program (created by using Jamox), a function name to extract information, a name of an output variable, and input display variables name/unit for GUI. Fig. 15 shows the dialog provided by GUI of integrated environment, and it is used to input display variables name/unit for GUI.

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 select the generated RT control program, the module of the experimental apparatus, and input the parameter of the controller. Fig. 16 shows the screen of the RTWindow which is executing a control experiment.

Fig. 17 shows the experiment results, and abscissa axis is time [sec], ordinate axis is angle of pendulum [rad]. And, the sampling time of the experiment is 5 [ms]. As shown in Fig. 17, the angle of the pendulum is close to the 0 [rad], the experiment of stabilization control of a pendulum become successful.

VI. CONCLUSIONS

This paper proposed the methods which make execution of iterative design process of control system efficiently. In particular, this paper proposed a method which is based on the RT control framework, automatic generation of RT program, and separation of platform dependent parts. And, we developed the integrated environment for simulation and Real-Time control experiment which is implementation of proposed methods. Availability of proposed methods was shown by the stabilization of a pendulum.

REFERENCES

[1] M. Funaki and S. Ra, Guide book for Real-Time sensing and control by Linux. Shuwa system, 1999.
[2] “RTLinux,” http://www.fsmlabs.com/products.html.
[3] “The MathWorks Matlab,” http://www.mathworks.com/products/matlab/.
[4] M. Koga, MaTX for control/numerical analysis. Tokyo Denki University Press, 2000.
[5] M. Koga, H. Toriumi, and M. 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] “The MathWorks Simulink,” http://www.mathworks.com/products/simulink/.
[8] “GE Fanuc Automation: Proficy HMI/SCADA - iFIX,” http://www.gefanuc.com/en/ProductServices/AutomationSoftware/Hmi_Scada/iFIX/.
[9] W. GREGA and K. KOLEK, “Simulation and Real-Time Control from Simulink to Industrial Applications,” 2002, 2002 IEEE International Symposium on Computer Aided Control System Design Proceedings pp.104-109.
[10] M. Basso and G. Bangi, “ARTIST-A Real-Time Interactive Simulink-based Telelab,” 2004, proceedings of the 2004 IEEE Conference on Computer Aided Control Systems Design pp.196-201.
[11] M. Koga, Y. Tatsui, and J. Yabuchi, “Java Simulation Platform for Control System based on Block Diagram,” 2006, IEEE 2006 CCA/CACSD, pp.2304–2308.
[12] M. Koga and T. Matsui, “Development of OS-Neutral Numerical Foundation Class Library and its Application to Control System,” 2003, SICE 6th Annual Conference on Control Systems.
[13] T. Nakano, “Understanding the framework,” Java WORLD, vol. 6, no. 62, pp. 54–67, 2002.
[14] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesle Pub, 1995.
[15] K. Yano and M. Koga, “Platform Independent Integrated Environment for Simulation and Real-Time Control Experiment,” 2006, SICE-ICASE International Joint Conference 2006.
[16] M. Koga and S. Kawakami, “Development of Portable Real-Time Remote Control System in Java and its Application to Swing Up Control of Inverted Pendulum,” 2003, SICE Annual Conference 2003 in Fukui.
[17] R. Gordon, Essential JNI: Java Native Interface. Prentice Hall Pr., 1998.
[18] M. Koga, T. Matsuki, and H. Sada, “Development of Environment of Numerical Computation in Java based on Object Model of Programming Language,” 2005, 49th Annual Conference of the Institute of Systems, Control and Information Engineers.
[19] T. Ueta, Java’s Karakuri 2(practice version). Ltd. IDG Japan, 2001.