Proceedings of the 11th KACC, October 1996

On-Line Tuning of Controllers with State Observer Using A Real-Time CAD of Control Systems

°Masanobu Koga*, Hiroaki Toriumi* and Mitsuji Sampei*

Department of Mechanical and Environmental Informatics
Graduate School of Information Science and Engineering
Tokyo Institute of Technology, 2-12-1, Oh-okayama, Meguro-ku, Tokyo 152, JAPAN
Tel: +81-3-5734-2328; Fax: +81-3-5734-2552; E-mail: koga@mei.titech.ac.jp

Abstracts

In this paper development of a CAD of control systems is introduced which enables us to do not only analysis of control systems, design of controllers but also real-time implementation of controllers. By utilizing this software, the control engineer is able to repeat the procedure of modification of controllers and experiments without recompile to attain better performance. The software also offers the facility to update the parameters of controllers without stopping real-time control, which helps on-line tuning of controllers. If some parameters of the controller is changed on-line, the control input may change discontinuously. It has serious effect on the control systems. A method for on-line tuning of state feedback controller with state observer is proposed and verified through the experiment with an inverted pendulum.

Keywords CADCS, On-line Tuning, Real-Time Control, Simulation

1. Introduction

During the last few years, there has been an increased interest in automating the process of implementing digital controllers. The driving forces behind this development has been the wish to have a better consistency between the designed and implemented controllers. This has led to a number of modules for design software packages. AutoCode for Matrixx[2] and the C-code generation toolbox for Simulink[1] are examples of such products. One of the problems of testing implemented controllers in a laboratory environment is that the iterative nature of the design process makes it necessary to perform ad hoc modifications and experiments in order to arrive at an optimal control performance. These experiments and changes should be possible to make in a efficient way without having to recompile.

In this paper development of a CAD of control systems is introduced which enables us to do not only analysis of control systems, design of controllers but also real-time implementation of controllers in one environment. By utilizing this software, the control engineer is able to repeat the procedure of modification of controllers and experiments without recompile to attain better performance. The software also offers the facility to update the parameters of controllers without stopping real-time control, which helps on-line tuning of controllers. If the parameter of a controller is changed on-line, the control input may change discontinuously. It has serious effect on the control systems. A method for on-line tuning of state feedback controller with state observer is proposed and verified through the experiment with an inverted pendulum.

The system has been used in research and education, and has reduced the implementation time considerably, e.g. when developing new laboratory exercises, or when a control algorithm is tested in a laboratory experiment.

2. Real-Time Implementation

There are a lot of softwares for real-time implementation of the designed controllers such as Real-Time Workshop (Mathworks Inc.), Realsim/AC-100 (ISI), Vissim (Visual Solutions), and so on. Vissim enables us to write the controller by using block diagram interface, however, the speed of computation is slow since it is an interpreter. Real-time Workshop and Realsim/AC-100 generate real-time control programs directly from block diagram of the control system and provide matrix calculation library. Though they are excellent softwares for real-time implementation we have to recompile the generated code as show in Fig. 1, if we change the parameters of the controller or at least the structure of the controller. In this work we developed a CAD (RTMATX) for real-time implementation, which provides real-time implementation facility as well as facility of analysis and synthesis of control system. We integrated real-time implementation facility to CACSD software MATX [4, 5, 3], which allows the definition and manipulation of matrices in a mathematical algebraic fashion. Since RTMATX programs can be compiled, the speed of computation is as fast as C programs.

When we use the commercial software above for real-time implementation, we have to stop real-time control in order to change parameters of controller. If the real-time control is not stopped, the computed control output may be incorrect since some parameters are

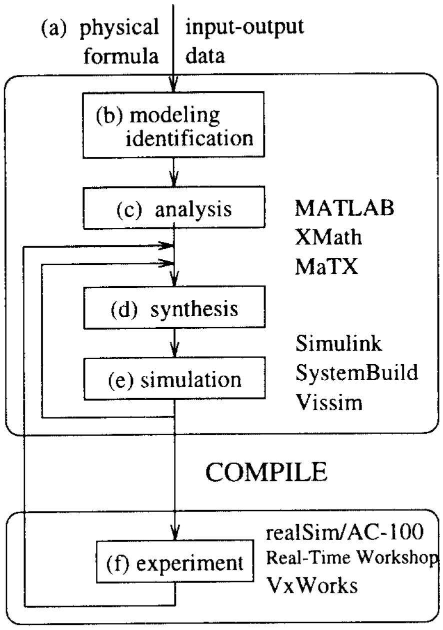

Figure 1: Real-time implementation with compile

updated and others are not at one moment. RtMaTX allows changing parameters of controller without stopping real-time control as shown in Fig. 2. This facility makes it possible to iterate the procedure of control system design efficiently.

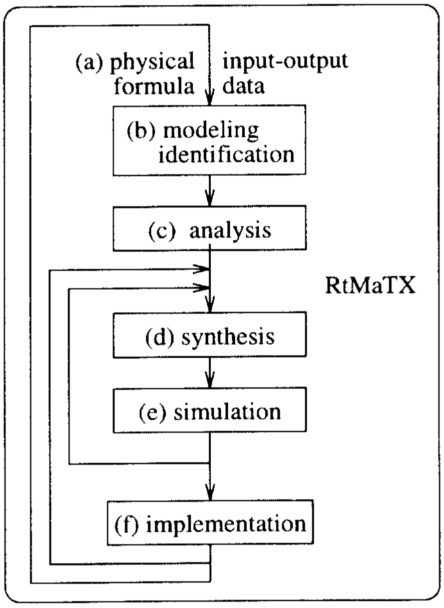

Figure 2: Real-time implementation without compile

3. Simulation and Real-Time Implementation

We deal with the simulation and the real-time implementation of the inverted pendulum shown in Fig. 3 to illustrate the facility of RtMaTX. Consider the lin-

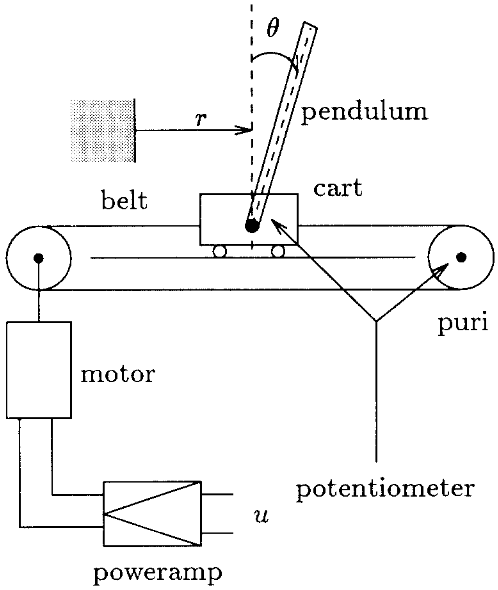

Figure 3: Inverted pendulum

earized model of the plant:

$$\dot{x} = Ax + Bu \quad (1)$$
$$y = Cx \quad (2)$$

where $x \in R^n$, $u \in R^m$ and $y \in R^p$. We design an LQ state feedback control law

$$u = -Fx \quad (3)$$

and an observer

$$\dot{z} = A_h z + B_h y + J_h u \quad (4)$$
$$x_h = C_h z + D_h y. \quad (5)$$

The observer is discretized for digital control such that

$$z[k+1] = A_{hd} z[k] + B_{hd} y[k] + J_{hd} u[k]$$
$$x_h[k] = C_h z[k] + D_h y[k].$$

3.1 Simulation

The simulation program consists of three functions, `main()`, `diff_eqs()`, and `link_eqs()`. `diff_eqs()` is the function which calculates the derivative of state vector $\dot{x}$ and `link_eqs()` is the function which calculates control inputs u. In `main()` the differential equation is integrated according to RKF45 algorithm automatically changing the step size to guarantee the specified computation error. The list of `link_eqs()` is shown in List 1. See [3] for other functions.

3.2 Real-Time Implementation

The real-time implementation program consists of three functions, main function `main()`, on-line function `on_task()`, and off-line function `off_task()`. Function `on_task()` is the function for calculation of control inputs and `off_task()` is the user-interface function for showing and changing parameters. Once the real-time control starts, `on_task()` is called every sampling period. The list of `on_task()` is shown in List 2. The function `sensor()` returns the output of the the plant measured by sensors, and `actuator()` operates the actuator. See [3] for other functions.

```
Func void link_eqs(u, t, x)
 Matrix u, x;
 Real t;
{
 Matrix y, xh;

 y  = C*x;              // output equation
 xh = Ch*xo + Dh*y;     // state estimation
 u  = - F*xh;           // state feedback
 z  = Ah*z + Bh*y + Jh*u; // observer
}
```

List 1: Function for control input

```
Realtime Matrix Ah,Bh,Ch,Dh,Jh,F;

Func void on_task()
{
  Matrix y, xh;

  y  = sensor();
  xh = Ch*z + Dh*Y;      // state estimation
  u  = - F*xh;           // state feedback
  z = Ah*z + Bh*y + Jh*u; // observer
  actuator(u);
}
```

List 2: Function for real-time control

4. On-Line Tuning

4.1 How to update parameters

We can utilize two tables of parameters in order to share parameters of the controller between non-real-time functions and real-time functions [6]. As shown in Fig. 4 the real-time function refers to parameters of one table for calculation of control output, while the non-real-time function changes parameters of another table. After all parameters are changed, two parameter tables are exchanged. Accordingly we can change parameters with consistency of all parameters.

4.1.1 Real-time variable

If we use the method proposed in [6], the table of parameters must be maintained by programmers. RTMATX introduces idea of **real-time variable** in order that the shared parameters can be changed without taking care of parameter tables. If we declare the variables which should be shared between real-time and

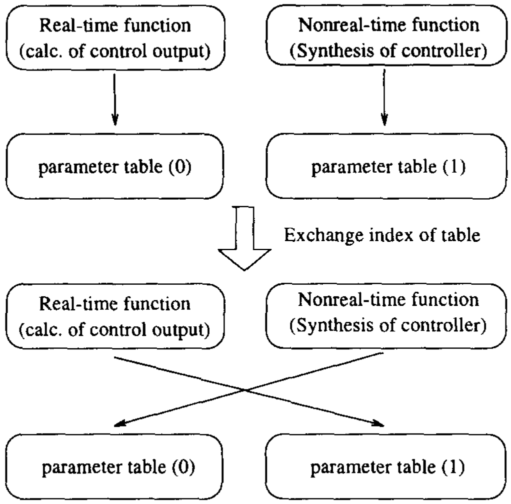

Figure 4: Exchange of parameter table

non-real-time functions as **real-time variables**, two parameter tables including these variables are prepared automatically, and the consistency of all parameters in the tables is guaranteed. Therefore we don't have to take care of parameter table to update the shared parameters.

We consider two implementations for real-time variables : double-array scheme and double-function scheme.

4.1.2 Double-array scheme

The double-array scheme prepares two arrays for parameter table and exchanges the tables by changing the index of arrays. A reference to real-time variable in the program corresponds to a reference to an element of an array in C program. It takes a little extra time for the double-array scheme to calculate the index of an array to access a real-time variable.

4.1.3 Double-function scheme

The double-function scheme prepares two functions for real-time control and exchanges the parameter tables by exchanging functions. The speed of computation is a little bit faster than double-array scheme since the calculation of the index of an array to access a real-time variable is not necessary. The size of program is bigger than that of the program generated by double-array scheme.

4.2 On-line tuning of state observer

We consider the effect of changing the parameter of the controller without stopping real-time control. The plant is the inverted pendulum whose state equation and output equation are given by (1) and (2). The controller consists of state feedback (3) and minimum order state observer (4) and (5). We assume that the parameters are changed at the operating point around the equilibrium point ($x = 0, u = 0$).

4.2.1 State Feedback

We found that the effect of change of state feedback is small through some simulations and experiments, since the state is around the equilibrium point $x = 0$.

4.2.2 State Observer

When the parameter of the observer is changed, the estimated state x_h may change discontinuously and control input u may change dramatically if the state of the old observer is still used for the initial state of the new observer. We propose the method to determine the initial state of the updated observer such that the estimated state of the new observer consistent with one of the old observer.

We design the minimum state observer for the linear system (1) and (2) by Gopinath's algorithm. The state transformation $x = T\bar{x}$ yields the equivalent system

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u & (6) \\ y &= \bar{C}\bar{x}, \quad C = [O \; I_p] & (7) \end{aligned}$$

Let the minimum order state observer for the equivalent system is described as

$$\begin{aligned} \dot{\bar{z}} &= \hat{\bar{A}}\bar{z} + \hat{\bar{B}}y + \hat{\bar{J}}u & (8) \\ \hat{\bar{x}} &= \hat{\bar{C}}\bar{z} + \hat{\bar{D}}y & (9) \end{aligned}$$

where

$$\begin{bmatrix} \hat{\bar{C}} & \hat{\bar{D}} \end{bmatrix} = \begin{bmatrix} I_{n-p} & L \\ O & I_p \end{bmatrix} \quad (10)$$

, L is a constant matrix which determines the poles of the observer. We obtain the state of the observer $\bar{z}$ from (9) and (10) as follows.

$$\begin{aligned} \bar{z} &= \begin{bmatrix} I_{n-p} & O \end{bmatrix} \hat{\bar{x}} - Ly \\ &= \begin{bmatrix} I_{n-p} & O \end{bmatrix} T^{-1}\hat{x} - Ly & (11) \end{aligned}$$

If we choose the initial state of the new observer as (11), the estimated state of the new observer consistent with one of the old observer.

4.2.3 Experiment

The Fig. 5 and Fig. 6 shows the experimental results of the inverted pendulum. We changed the poles of the observer from multiple double root -70 to multiple double root -20 at 3 second.

5. Summary

A new CAD of control systems, which enables us to do not only analysis of control systems, design of controllers but also real-time implementation of controllers, was introduced. The software offers the facility to update the parameters of controllers without stopping real-time control, which helps on-line tuning of controllers. A method for on-line tuning of state feedback controller with state observer was proposed and verified through the experiment.

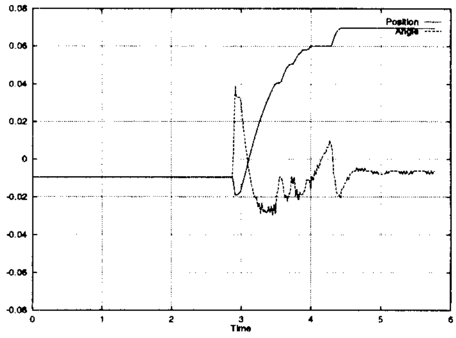

Figure 5: The initial state of observer is unchanged

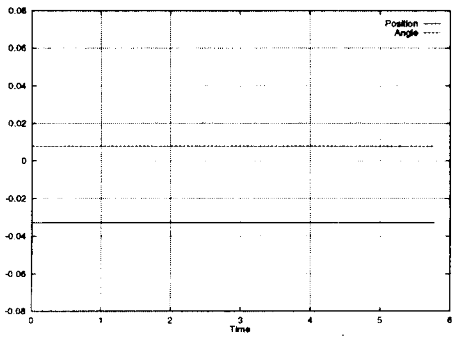

Figure 6: The initial state of observer is changed according to the proposed method

References

[1] The Math Works Inc. *SIMULINK – User's Guide*. 24 Prime Park Way, Natick, Mass. 01760-1500, USA, first edition, 1992.

[2] Inc. Integrated Systems. *MATRIXx Core*. 3260 Jay Street Santa Clara, California 95054, USA, 8 edition, 1991.

[3] Masanobu Koga. An interactive environment for simulation and real-time implementation of control systems. *Proc. of KACC'95, Seoul, Korea*, pages 336–339, 1995.

[4] Masanobu Koga and Katsuhisa Furuta. Programming language MaTX for scientific and engineering computation. In Derek A. Linkens, editor, *CAD for Control Systems*, chapter 12, pages 287–317. Marcel Dekker, Inc., July 1993.

[5] Masanobu Koga, Mitsuji Sampei, and Katsuhisa Furuta. A compiler of matlab to MaTX: Compiling and linking of m-files to an executable program. *Proc. of CACSD'94, Tucson, Arizona, U.S.A.*, pages 137–142, 1994.

[6] Suzumu Sekino. Integration of real-time kernel to cad. Bachelor thesis, Tokyo Institute of Technology, Department of Control Engineering, 1995.