NIPPON STEEL TECHNICAL REPORT No. 67 OCTOBER 1995

UDC681.3:519.6:621.771.23

Development and Application of Dynamic System Simulator

Yuusuke Konno[*1] Masanori Shioya[*1]
Takatsugu Ueyama[*1]

Abstract:

By capitalizing on control and simulation technologies fostered in the steelmaking business, a general-purpose dynamic system simulator was developed as a process control system development tool for Nippon Steel's industrial system solution business. The simulator can analyze general dynamic systems represented by ordinary differential equations, difference equations, and algebraic equations. Equipped with a powerful numerical analysis function and an easy-to-understand user interface in block diagram format, it can also readily simulate dynamic systems. The characteristics and configuration of the dynamic system simulator, and a hot strip mill process simulator as an example of its application are described.

1. Introduction

The development of various process control systems requires the full understanding of the behavior of the processes concerned. Field experimentation is difficult to carry out due to such problems as operating conditions and cost. If an opportunity is afforded for experimentation with a pilot plant, the results of the pilot plant experiment can only be verified in limited cases.

Under these circumstances, a process simulator using a computer is essential to:

1) Understand the behavior of the process in question.
2) Verify the effectiveness of a method devised for controlling the process.
3) Carry out research and development by intoducing new concepts.
4) Nurture young engineers through simulation of process control.

Traditionally, simulation programs for specific processes were created in general-purpose programming languages, such as FORTRAN, but their maintenance and modification requiered large amounts of time and labor. Before tackling the simulation of processes, control engineers had to learn computer programming and could not simulate the processes as readily as they wished.

The authors developed an interactive dynamic system simulator as a powerful for the industrial system solution business tool for solving the problems of conventional simulators as noted previously.

This paper gives an overview of the dynamic system simulator and describes a hot strip mill process simulator as an example of its application.

2. Description of Dynamic System Simulator

2.1 Design philosophy

A good understanding of the mechanism of the process on the shop floor is the first step toward creating a new idea in the control system design phase. It is important to accurately grasp the physical phenomena taking place in the process, use one's imagination, then draw a dynamic image of the process in one's mind. Observation alone of the actual process involves such problems as:

(1) Few of the phenomena actually taking place in the process can be observed.
(2) The process cannot be manipulated as desired.

[*1] Technical Development Bureau

(3) The process cannot be experimented with as freely as desired.
(4) It is difficult to identify problems when the process is operated on a routine basis.

A simulator that dynamically analyzes the process and its control system on a computer is a powerful tool for solving these problems.

The transient behavior of the process can be microscopically known by imposing many conditions and operations on the process and repeating simulation experiments. This procedure allows us to imagine process control and develop our imagination to produce new ideas. When the developer builds a simulator for the process, he or she can rearrange the relations between the mathematical models used in the simulator. This greatly helps the developer to understand the process mechanism.

In this way, process simulation not only substitutes for field experimentation, but also promotes understanding of the process concerned, and encourages new control ideas for markedly improving the performance of the process. For the process simulator to accomplish these roles, it must meet the following conditions.

2.1.1 High visualization and simulation abilities

To image a phenomenon taking place in an actual process, it is necessary to experience it as if operating the process. To this end, the simulator should provide beautiful images so it is possible visually confirm the phenomenon, and allow free intervention.

2.1.2 Easy-to-understand simulator structure

When the simulator is designed for computer experimentation, it should be capable of simulating any conceived idea quickly and simply. To this end, its structure must be easy to understand, and it must be simple to modify. This is a sharp contrast with conventional software simulators that were complicated in structure and difficult to understand, and were left unused and unmaintained once their developers left the company.

2.1.3 Structure for easy control system design

The processes to be simulated are usually nonlinear and vary in characteristics over time. The control theory is usually built for linear time-invariable models. It is extremely difficult to immediately design a control system using simulators. Manual linearization requires much labor and time. As a result, the thought process is interrupted, and development efficiency is reduced. It is desirable that for a controller to realize an idea, it should be capable of being simply designed with a simulator. Furthermore, the simulator should be capable of immediately confirming the behavior of the process incorporating the controller.

2.2 Characteristics of dynamic system simulator

The characteristics of the dynamic system simulator developed to meet the above-mentioned goals follow.

2.2.1 Visual and interactive operation

Fig. 1 shows an example of a dynamic system simulator's simulation screen. Displayed here are the volume and graph. As soon as the volume changes, the resultant change in the system is reflected in the graph. From theses graphs and meters, the calculated results are displayed on the screen from time to time.

Operating the volume or switch with a mouse or keyboard, the user can manually intervene, and conduct controller switchovers and control system adjustments by successively observing the motion of the simulator. This interactive operating environment affords the user high visualization and simulation capabilities.

2.2.2 Modular structure

So that the internal structure of the simulator can be easily understood, each unit (model) of the simulator developed on this dynamic system simulator is represented by a block with input and output terminals and the relationship between individual units is represented by connecting lines. Two or more blocks can be denoted by a single module, and a large-scale and complex simulator can be built in a hierarchical and easy-to-understand structure. **Fig.2** shows that the blocks enclosed within the border can be denoted by a single module.

Since such a simulator is not written in a programming language but is represented by blocks (modules) and connecting lines in this way, the individual units and their relationships can be visually recognized, and their configuration can be readily understood by users other than its developers. Since individual blocks are independent of each other, the correction of a block or addition of a block does not lead to the correction of the whole simulator. It is also easy to partially modify the simulator or add functions to the simulator, and the simulator can be easily reused as a different simulator. Frequently used blocks, such as adders and integrators, are prepared as built-in blocks as shown in **Fig. 3**. This means that component parts are standardized.

2.2.3 Control system design function

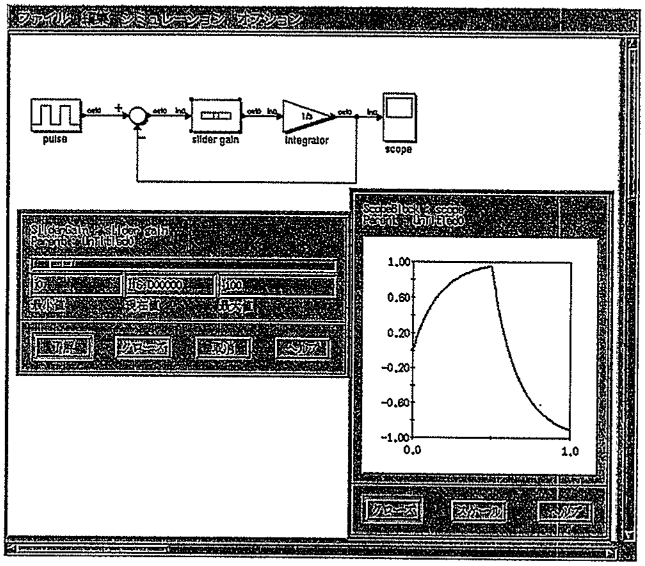


Fig. 1 Volume and graph

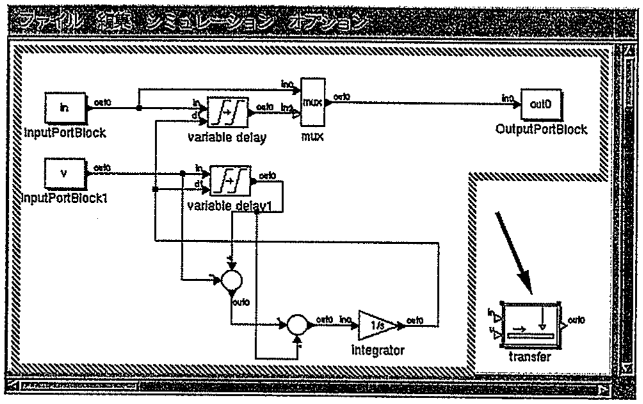


Fig. 2 Modularization function

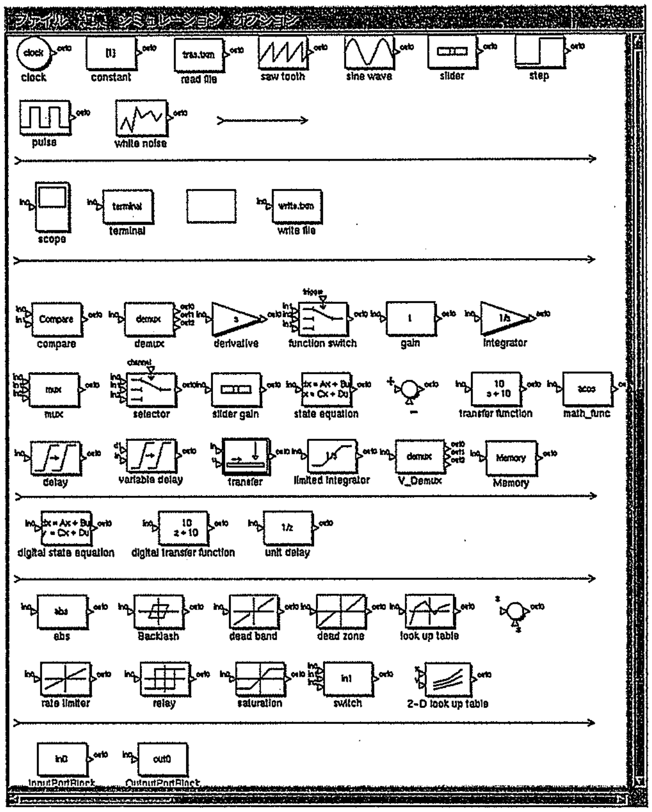


Fig. 3 List of built-in blocks

An automatic linear model generation function was developed so that the simulator can be stopped at any given point of time during simulation and so that a linear time-invariant model can be calculated in that condition. A controller can be easily designed by incorporating the model into the control system design tools developed by the authors' laboratory.

The designed controller can be automatically introduced into the simulator and executed. This function allows control system design and simulation to be performed without delay and computer experimentation to be continued without interrupting the thought process.

2.2.4 Expandability

If the built-in blocks of **Fig. 3** are functionally insufficient, the user can describe new blocks in the C or C++ language. The new blocks are compiled and dynamically connected as a dynamic link library when simulation is carried out. For this reason, the blocks need not be relinked as is done with ordinary programs, and the system is structured for simple block development. Since object-oriented software development technology is used for making the simulator, the software ensures high expandability and reusability of the simulator itself.

2.3 Numerical analysis techniques

Systems addressed by the dynamic system simulator are generally described in the following form:

$$\frac{dx_c}{dt} = F(x_c, x_d, x_a, t) \quad \cdots\cdots(1)$$

$$x_d[i+1] = G(x_d[i], x_c, x_a, t) \quad \cdots\cdots(2)$$

$$H(x_c, x_d, x_a) = 0 \quad \cdots\cdots(3)$$

Eq. (1) is a differential equation, Eq. (2) is a difference equation, and Eq. (3) is an algebraic equation. They must be simultaneously solved. Of the three equations, the first and third must be approximately solved by some numerical scheme. Their solution methods are outlined below.

2.4 Differential equation

Various methods are available for solving the differential equation (1). They may be divided into explicit and implicit methods. The explicit and implicit methods each have the following characteristics.

Explicit method: Since the value at the next point of time can be immediately calculated from the value at a given point of time, programs can be easily created. Numerical instability is likely to occur at large time steps.

Example: Forward Euler method $x_{i+1} = x_i + hf(x_i)$

where x_i is the solution at present point of time; x_{i+1} is the solution at next point of time; h is the time step width; and $f(x)$ is the integrand.

Implicit method: Instability is unlikely to occur at large time steps. Since the equation must be solved at each time step, the amount of calculation required per time step increases.

Example: Backward Euler method $x_{i+1} = x_i + hf(x_{i+1})$

The dynamic system simulator allows the user to select an integration method from among three explicit schemes and three implicit schemes.

It is important that the time step width be appropriately selected for the explicit methods in view of the stability constraint, and for the implicit methods in view of the integration error. It is difficult for the user to determine an appropriate time step width for specific problems and this preferably should be determined automatically. The time step width can be large where the solution does not appreciably change, but should be smaller where the solution suddenly changes. For these reasons, the time step width is automatically adjusted for all integration methods. Once the user enters the minimum and maximum time step widths and the permissible error, the time step width is automatically adjusted to provide the maximum time step width that meets the permissible error.

As an example of automatic time step width adjustment, the solution of the time step width by the Van der Pol nonlinear differential equation[1] expressed by Eq. (4) follows.

$$\ddot{x} = 50(1-x^2)\dot{x} - x, \; x(0) = 1, \; \dot{x}(0) = 1 \quad \cdots\cdots(4)$$

This differential equation can only be solved by an explicit scheme in small time steps.

Of the available integration methods, the Runge Kutta Fhelberg biquadratic and quintic method (RKF45), as a representative explicit scheme, and the backward biquadratic differential formula (BDF4), as a representative implicit scheme, were compared at a maximum time step width of 2 seconds and permissible error of 0.001.

In **Fig. 4**, the solutions of the two methods agree. The number of calculation steps performed to obtain these results was 2,272 steps for RKF45 and 712 steps for BDF4. This means that BDF4 was much more efficient than RKF45.

This is also clear from **Fig. 5** in which the time step width is greater for BDF4. With both methods, the time step width decreases where the solution dramatically changes, indicating the effectiveness of the automatic time step width adjustment.

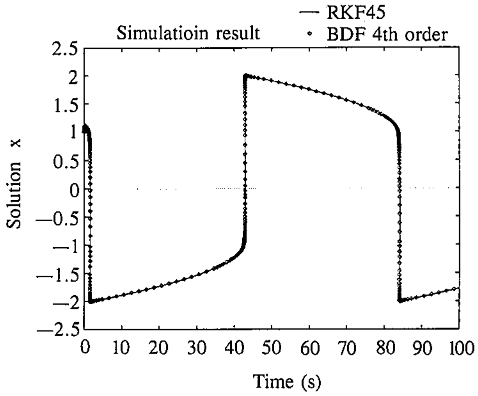


Fig. 4 Solution of Van der Pol equation

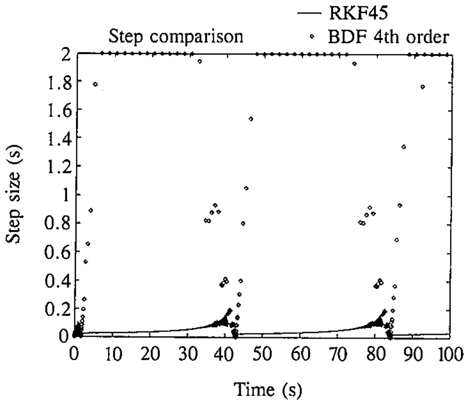
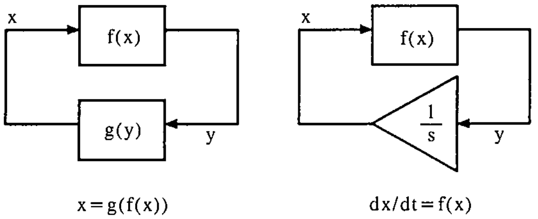


Fig. 5 Comparison of time step width



With algebraic loop

Without algebraic loop

Fig. 6 Algebraic loop

2.5 Algebraic equation

The algebraic equation to be solved during simulation may be contained in the simulation model (algebraic loop as shown in **Fig. 6**) or may accompany the implicit method for the differential equation described previously. The simple iterative method or Newton method is possible as a solution. Adopted here are the Newton method with good convergence and its variations. The simultaneous linear equations that appear in the iteration process of the Newton method are solved by the direct method (LU decomposition). The user can change the maximum number of iterations, convergence criterion, and other conditions.

3. Example of Process Simulator Construction (Hot Strip Mill Process Simulator)

To verify the performance of the dynamic system simulator, a tandem simulator for the finish rolling train of a hot strip mill was built and used to develop various control systems[2,3]. The configuration of the hot strip mill process simulator and the frequency-shaping AGC system as an example of control system development are described below.

3.1 Configuration of hot strip mill process simulator

The hot strip mill process simulator is composed of functional modules as shown in **Fig. 7**. The modules can be copied for easy building of multiple stands. The principal module functions follow:

- Mill model
- Looper model
- Looper control system
- Tension model
- Interstand creep model
- Thickness control system
- Transfer model
- Other control functions

As time elapses, the hot strip mill process simulator can verify at the midwidth of strip along its length the behavior of the following variables:

1) Strip thickness
2) Strip width
3) Strip crown and profile (elongation strain difference) at a given point from the edge
4) Unit interstand tension
5) Strip temperature
6) Strip speed
7) Roll speed
8) Screwdown position
9) Looper angle

Simulation can also be performed by incorporating operating data. The details of these models are discussed in another report[4].

3.2 Development of control system

Frequency-shaping AGC[3,5] was developed to simultaneously reduce the effects on the strip thickness of skidmarks and roll

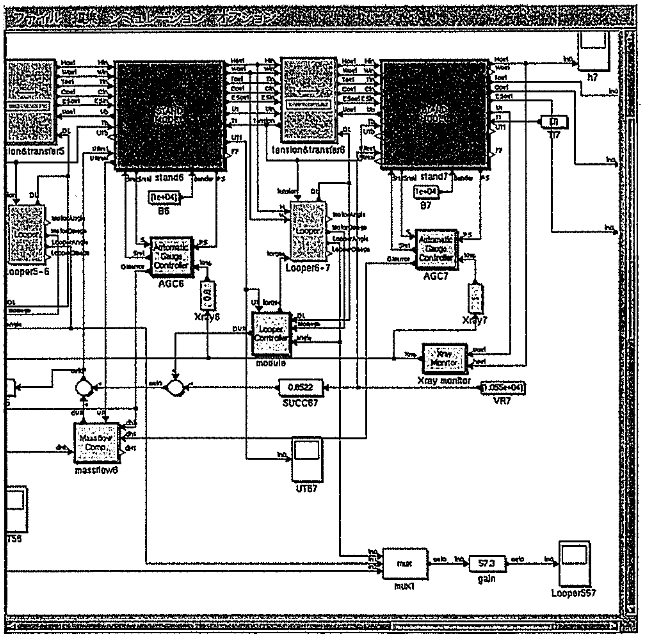


Fig. 7 Part of simulator for seven finishing stands at hot strip mill

eccentricity the two largest disturbances (see **Fig 8**) responsible for the deviation of strip thickness in the fishing train of the hot strip mill. As shown in **Fig. 9**, the conventional BISRA and gagemeter AGC systems controlled the strip thickness by adjusting a single parameter called the tuning factor. When the strip thickness deviation is detected and controlled on the basis of the rolling reaction force, the two disturbances act as follows:

- Skidmark disturbance increases strip thickness as the rolling reaction force increases.
- Roll eccentricity disturbance decreases strip thickness as the rolling reaction force increases.

When the effect of one of the two disturbances is reduced, the effect of the other increases.

The frequency band is mainly 10 rad/s and more for the roll eccentricity disturbance, and about 1 rad/s for the skidmark disturbance. That is, the frequency bands for the tow disturbances are separated.

Focusing on this point, the frequency-shaping AGC system controls strip thickness in each frequency band according to the frequency of each disturbance. The block diagram of the frequency-shaping AGC system is shown in **Fig. 10**. The fast roll eccentricity disturbance is eliminated by the loop α_1, while the slow skidmark disturbance is eliminated by the loop α_2.

Fig. 11 shows the frequency characteristics of the transfer function $G_1(s)$ from the skidmark disturbance ΔH to the exit strip thickness deviation Δh, and the transfer function $G_2(s)$ from the roll eccentricity disturbance ΔS_e to the exit strip thickness deviation Δh.

Fig. 12 shows the conventional BISRA AGC system in which the tuning factor α_1 is set at 1.0. The effect of roll eccentricity disturbance is pronounced.

Fig. 13 shows the frequency-shaping AGC system in which the tuning factors α_1 and α_2 are set at -1.0 and +1.0, respectively. The effect of roll eccentricity disturbance is reduced by about 60% in this case.

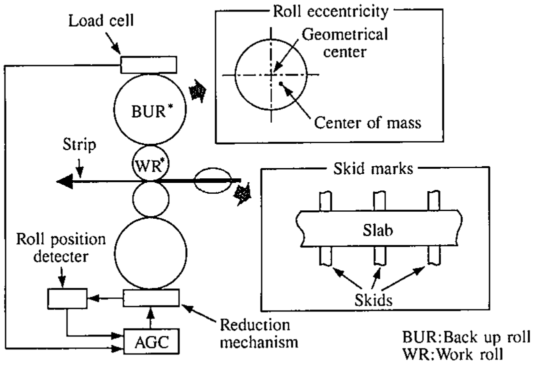


Fig. 8 Strip thickness disturbances in hot strip mill process

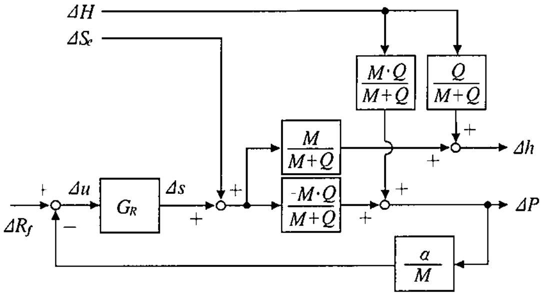


Fig. 9 BISRA AGC system

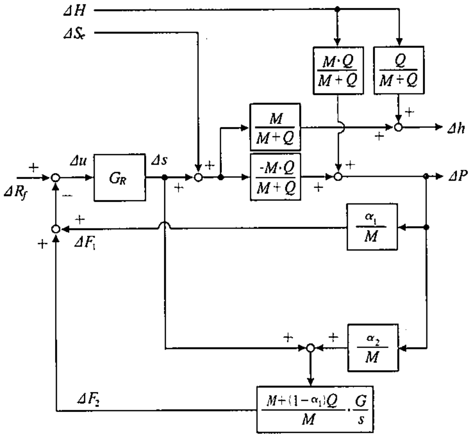


Fig. 10 Frequency-shaping AGC system

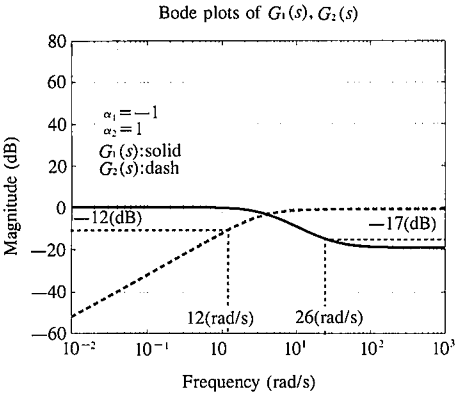


Fig. 11 Frequency characteristics of $G_1(s)$ and $G_2(s)$

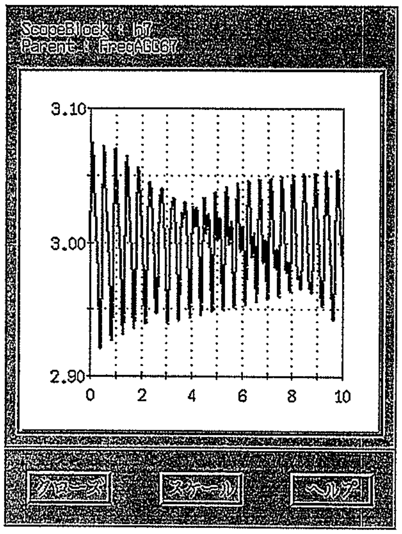


Fig. 12 Output of BISRA AGC system

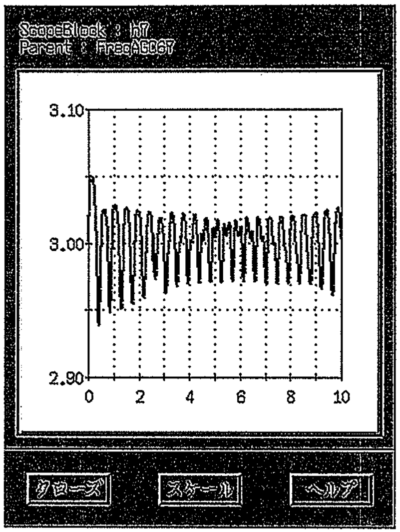


Fig. 13 Output of frequency-shaping AGC system

4. Conclusions

The dynamic system simulator is an extremely powerful tool for building easy-to-understand and easy-to-use process simulators. It is highly efficient from a numerical analysis standpoint. The development of the hot strip mill process simulator and frequency-shaping AGC system has verified the great effectiveness of the dynamic system simulator as a tool for designing the control systems of large processes.

The numerical analysis function of the dynamic system simulator will be increased in capacity and speed. Using the dynamic system simulator, process models adapted for the industrial system solution business will be built and accumulated as a library. With the addition of the discrete phenomenon system simulation function, the dynamic system simulator will evolve as a tool for simulating the entire plant from scheduling to process simulation.

References

1) Hairer, E., Norsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I. Springer-Verlag, 1987
2) Hot Strip Mill Looper Multivariable Noninteracting Control. Presented at 97th Meeting of Rolling Theory Committee, Joint Research Society, Iron and Steel Institute of Japan, November 1992
3) Hot Strip Mill AGC with Two Degrees of Freedom. Presented at 94th Meeting of Rolling Theory Committee, Joint Research Society, Iron and Steel Institute of Japan, September 1991
4) Development of Dynamic Rolling Simulator. Presented at 112th Meeting of Control Technology Committee, Joint Research Society, Iron and Steel Institute of Japan, November 1994
5) An Application of Two-Degree-of-Freedom Control to Automatic Gauge Control. American Control Conference, 1992, p. 806-810,