

High Fidelity Real-time Railroad Simulation

-- Its necessity, advantages, and results

S. Yaskawa

Yaskawa Information Systems Corporation, Kawasaki, Japan

Abstract: There have been no major control problems during the first three years of the new Tokaido-Sanyo Shinkansen (bullet train) Programmed Route Control (PRC) computer system. Prior to its commissioning in December, 1999, the first two years of the previous PRC computer system's operation experienced 10 major problems. This significant software quality improvement has been achieved through the use of a high fidelity real-time railroad simulation built on the Parallel Inference Machine (PIM). This simulator also contributes to significant improvement of every phase of the PRC computer system development and operation. The high fidelity real-time railroad simulator models such things as track and facility data, train schedules, train performance, signal and safety control system, route (track switch) control system, and train drive and control system. The model is as true to the physical system as possible, simulates the operation of trains in real-time (faster than real-time if necessary), connects on-line with the control computer, reproduces or creates any system behavior of trains including schedule disruption for 24 continuous hours (or suspension and resumption if necessary). It tests new control schemes, tests the whole control system, trains rail dispatchers, verifies new train schedules, tests new facility plans, etc. This paper describes why the simulator on the PIM has been indispensable, its advantages, and its measured effects both technological and economical.

Keywords: *Real-time; Parallel; Simulation; Control; Railroad*

1. INTRODUCTION

There have been no major control problems during the first three years of the new Tokaido-Sanyo Shinkansen (bullet train) Programmed Route Control (PRC) computer system. Prior to its commissioning in December, 1999, the first two years of the previous PRC computer system's operation experienced 10 major problems. This significant software quality improvement has been achieved through the use of a high fidelity real-time railroad simulation built on the Parallel Inference Machine (PIM). This simulator also contributes to significant improvement of every phase of the PRC computer system development and operation. It tests new control schemes, tests the whole control system, trains rail dispatchers, verifies new train schedules, tests new facility plans, etc. This paper describes why the simulator on the PIM has been indispensable, its advantages, and its measured effects both technological and economical.

2. HIGH FIDELITY REAL-TIME RAILROAD SIMULATOR

This chapter describes what the high fidelity real-time railroad simulator is and its prerequisites. It models the things listed below, for 24 continuous hours, as true to the physical system as possible, simulates the operation of trains in real-time (double speed if necessary) at a fidelity of

realistic acceleration and deceleration, connects directly with the real-time on-line control computer, reproduces or creates any system behavior of trains including schedule disruption. The model can also be suspended and resumed as necessary. It tests new control schemes, tests the whole control system, trains rail dispatchers, verifies new train schedules, tests new facility plans, etc.

- Track and facility data
- Train schedules
- Train performance
- Signal and safety control scheme
- Route (track switch) control system
- Train control and drive system
- etc.

To realize the above requirements, the simulator has to:

- Simulate operation facility conditions as true to the physical system as possible,
- Simulate operation of all trains on the rail at a fidelity of run-curve (acceleration and deceleration) taking all train operation conditions into account based on a train schedule,
- Create easily any abnormal situation caused by operation facility trouble or train breakdown (malfunction),

- Simulate train operation in schedule disruption as true to real case as possible,
- Be able to do the above mentioned simulations in real-time, or double speed if necessary for better test efficiency, and
- Be capable to test the functions of new train tracking, track switch control, and schedule disruption monitoring for the next generation of the PRC system.

3. MULTIPLE NECESSITIES AT THE COMTRAC PROJECT

3.1. Introduction

The PRC (Programmed Route Control) computer is the central core of the Shinkansen (Bullet Train) Operation Control and Monitoring System COMTRAC. The simulator is assumed to play the roles listed below at various phases of development and operation of the PRC computer.

- System feasibility study and design (Test of new control schemes and investment decision)
- System test
 - Test of the complex structure and dynamic behavior of the PRC (Programmed Route Control) Computer – Central CTC (Centralized Traffic Control) – Local CTC – ATC (Automatic Train Control) – Track Segment Circuits – Trains (Figure 1)
 - All-inclusive test including schedule disruption; tests are more stringent than those with actual trains
- Training rail dispatchers
- Verification of new train schedules
- Test of additional facility plans

3.2. System Feasibility Study, Investment Decision and Design

The following new control schemes were planned to be incorporated into the next generation of the PRC (Programmed Route Control) computer.

- New train tracking logic
- Control logic of excess alarms inquiry displays
- Monitoring and judging schedule disruption
- Supporting operation monitoring job

A thorough test to find if the above functions are feasible was necessary for the decision of investment and system design. To this end a high fidelity real-time railroad system simulation was needed.

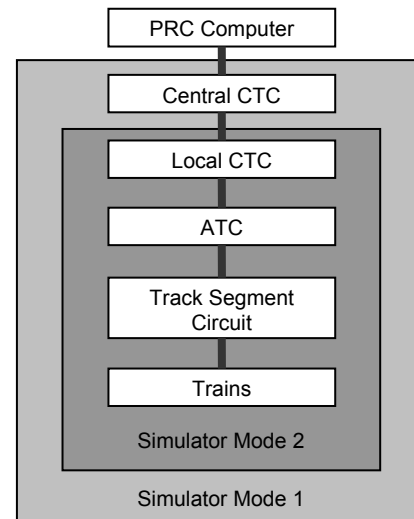


Figure 1. Structure of Train Control

3.3. System Test

Since the PRC computer is up and running the large-scale operation of bullet trains everyday, it is essential to give a thorough test to the new PRC computer before replacing the old one overnight. Those tests with actual trains like monitor-run and control-run are limited in coverage. Monitor-run tests the response of the new PRC to the inputs from the actual rail system so that its coverage is limited to normal operation. Control-run controls the actual rail system so that it can be applied to only a few specially scheduled trains in midnight.

The simulator was expected to cover the rest of train system behavior that monitor-run or control-run were unable to test. In other words the simulator should test the complex structure and dynamic behavior of the PRC (Programmed Route Control) Computer - Central CTC (Centralized Traffic Control) – Local CTC – ATC (Automatic Train Control) – Track Segment Circuits – Trains (Figure 1). There are two test modes: Simulator from trains up to Central CTC connects directly with the PRC computer; Simulator from trains up to Local CTC connects through Central CTC with the PRC computer. It is also possible to have an interface at ATC or track segment circuit.

The tests with the simulator were expected to be all-inclusive so that they are more stringent than those with actual trains. The simulator was to test actual or design-maximum number of trains, all actual combination of routes (track switches), and

any operation from planned schedule to disrupted schedule. See table 1 for the test coverage by the three types.

Since the tests with the simulator were so inclusive and realistic that the user would be able to give its approval to control computer system even at the manufacturer's site.

3.4. Training Rail Dispatchers

Training rail dispatchers has the following dilemma. Dispatchers are most needed in case of schedule disruption. It is almost impossible to give them on-the-job training since trains are running to the schedule most of the time. Training with real trains is possible only during test run of a new line. Occasional OJT is possible only when the schedule happens to be disrupted. A simulator that can reproduce or create any operation situation is able to solve the above dilemma.

3.5. Verification of New Train Schedule

It must be verified whether a new train schedule is actually operable and causes no conflict with the operation monitoring control system and local facilities. This used to be an intensive task by an expert. A simulator that produces train operation true to the new schedule would make it easy to identify any operational problem or conflict. If the user has few experts in this regard, the simulator is an indispensable tool.

3.6. Test of New Facility Plans

Plans such as additional stations, and tracks and new control scheme like the new ATC (Automatic Train Control: non-stop braking) must be tested with planned schedules before design decision, and subsequent investment in physical plant. A simulator that models those new facility plans is indispensable since it is impossible to test real facilities in advance.

4. WHY SIMULATOR ON THE PIM?

The PIM, Parallel Inference Machine, is a real-time parallel computer equipped with Paracell,

Test Conditions	Related functions	Remarks	Simulator test	Monitor-run test	Control-run test	Total evaluation
(1) Number of trains	Route control Train tracking	How many trains can be tested to a maximum of 150 trains.	A	B	C	A
(2) Number of routes	Route control Train tracking	How many routes can be tested to the existing routes.	A	B	B	A
(3) Info input from real system	Train tracking	How wide the info input covers real facilities (Trains – ATC – interlocks – Local CTC – Central CTC)	□	A	A	A
(4) Control output to real system	Route control	How wide the info output covers real facilities (Central CTC – Local CTC – Interlocks – ATC – Trains)	□	□	A	A
(5) Actual train operation Planned schedule Disrupted schedule	Schedule file change Disruption monitor Train tracking	Actual train operation	□	A	A	A
(6) Planned train operation Planned schedule Disrupted schedule	Schedule file change Disruption monitor Train tracking	How freely the train operation is set up.	A	□	□	A

Table 1. Test Coverage by the Types

Note: A: full coverage; B: reasonable coverage; C: poor coverage

English-like parallel programming language, and the Navigator, intuitive easy-to-use development environment. It was chosen as the platform to build a simulator with such capabilities as described in the previous chapter. The reasons of the choice are described in this chapter.

The PIM with its Paracell language easily models any type of functioning object or agent such as train and signal. Neither special technique nor training is required for programming. Synchronization among objects or agents, which is a responsibility of the programmer on other systems, is incorporated in this system, and no application code for synchronization is necessary.

The fixed time scan operation of the PIM guarantees the real-time response of an application regardless of application system size or fidelity. Change of or addition to the models doesn't break the real-time guarantee.

A high-fidelity simulator is essential to test and verify a new computer system, but it is ironic that it is very difficult to verify the validity of the simulator itself. It is fairly easy to verify the validity of simulator application built on the PIM since realistic models are working in real-time on the PIM and it is easy to judge the appropriateness of the behavior.

Application system size and performance on the PIM is predictable since the PIM size can be configured to be proportionate to the number of system elements and the response time is fixed and guaranteed regardless of the size.

Completion of simulator before the control system to be tested is mandatory. To this end the distributed and parallel development capability of the PIM is important. The parallel and independent nature of Paracell statements makes quick development with a large number of developers possible.

One of the features of the PIM is on-line change, addition, and deletion of the program code as well as the data. There is no need for stopping and rebooting the whole application to fix bugs. This saves tens of minutes per bug compared with conventional computer systems.

Coding in Paracell is like describing system specifications in English. This allows the system developers a shortcut of the upper stream of the application development, and increase software productivity by the order of magnitude, along with the short turnaround time of fixing bugs.

The PIM is provided with a rich set of on-line debugging tools, which allows the user Paracell source code level debugging while the application is running. Those tools make debugging easy

along with the capability of on-line change, addition, and deletion of the program code.

The user can grow the application system incrementally in size, fidelity, and complexity thanks to the modular nature of the Paracell code and the capability of on-line change, addition, and deletion. Any prototyping work is not wasted.

Any engineer without programming experience can start Paracell coding on the first day of training, as opposed to one to two year training for the language C and other computer programming languages. The end user engineers can develop, enhance and maintain the application system.

The PIM is provided with industrial standard on-line interfaces such as Ethernet TCP/IP so that the simulator built on the PIM can test real-time control computers.

5. SYSTEM STRUCTURE AND CODE EXAMPLE

Figure 2 below shows the system structure of the railroad operation control and monitoring system and the simulator on the PIM

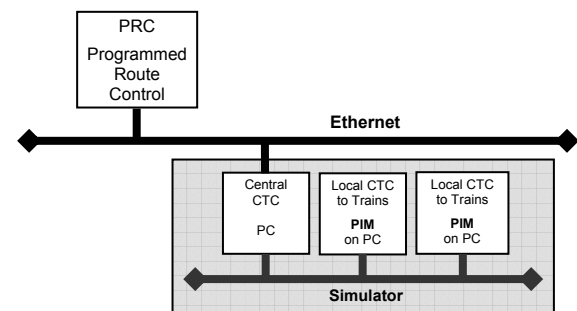


Figure 2. System Structure of the PRC and the Simulator on the PIM

A coding example in Paracell, the programming language of the PIM, is shown below.

//You can read the following program code without programming experience.

//Update the average speed

Average_speed_Vave is a local paracell number initially 0.

$Average_speed_Vave = (Next_station_distance_Pp - Current_distance_Pt) / (Next_station_time_Tp - Current_time_Tt).$

//Update the target speed

Current_target_speed_Vp is a local paracell number initially 0.

$Maximum_speed_Vp = \min (Maximum_speed_Vmax, ATC_signal_speed_Vatc, Average_speed_Vave * Driving_parameter_UP).$

//Update the train speed for the next frame

```

Local_acceleration_IAt and dv are local paracell
numbers initially 0.

Local_acceleration_IAt = Acceleration_At *
Simulation_speed.

dv = Current_target_speed_Vp - Actual_speed_Vt.

Increment Actual_speed_Vt by
if dv > 0
then Local_acceleration_IAt
else dv.

//Update the train distance

Run_step is a local paracell number initially 0.

Run_step = Current_speed_Vt * Frame_period
if 0 < Train_direction
then increment Current_distance_Pt by Run_step
else if Train_direction < 0
then decrement Current_distance_Pt by Run_step.

```

6. RESULTS FROM THE COMTRAC PHASE 7 DEVELOPMENT AND OPERATION

6.1. Introduction

A high-fidelity train operation simulator built on the PIM played multiple roles in the development and operation of the COMTRAC Phase 7 as described in the following sections.

6.2. Test of new control schemes

The simulator helped the design decision and subsequent investment in the COMTRAC phase 7 development. It could be very difficult decision without the high fidelity railroad simulator. The simulator minimized the investment risk of the new PRC system that could cost US\$5M to 50M. Risk saving is estimated as US\$500K to 5M.

6.3. Test of new control computer

Using the simulator achieved significant reduction in test cost, software quality improvement, and on-time commissioning.

It took only 24 times (days) of monitor-run test as opposed to 48 times for the previous phase 5. Monitor-run is a test with actual trains controlled by the current PRC, real input to the new PRC to be tested, and display output only. Each monitor-run requires 10 to 30 personnel. Cost saving is estimated as US\$720K, which is half of the previous monitor-run test: 30 man * 48 times * 3 days (preparation, test, and wrap-up) * US\$10K / 30 days.

The simulator helped significant software quality improvement of both the PRC computer and the central CTC with the two modes of test: direct connection with the PRC and connection through the central CTC with the PRC as shown in Figure 1.

No major control problems for the first three years of the new PRC computer. Prior to its commissioning in December, 1999, the first two years of the previous PRC computer system's operation experienced 10 major problems. The simulator contributed to keep the commissioning dead line and no problems during the start-up period

The simulator realized all-inclusive test including train schedule disruption. The tests are more stringent than those with actual trains. The simulator tested design-maximum number of trains, as opposed to the limited number of trains for control-run. Control-run is a test with actual trains, real input to and output from the new PRC to be tested. The simulator tested all possible combinations of route, i.e., track switch. Neither monitor-run nor control-run covers all combinations. It tested all possible operations from planned schedules to disrupted ones. Neither monitor-run nor control-run can include disrupted schedules. If everything must be tested by control-run, the cost could be as high as 100 man * 30 days * US\$10K / 30 days = \$1M. It saved the risk of US\$1M.

The simulator was also rented out to other subsystem manufacturers to test their systems and improve their software quality.

There was little going backward thanks to very few problems at monitor-run and control-run after all-inclusive test with the simulator. The simulator contributed to make the deadline, avoiding delay penalty.

6.4. Better quality of train schedules

Whenever the train schedule is revised, usually once every year and sometimes half a year, it used to be verified by an intensive work-month of an expert, and now is verified by simply running the simulator for one day to identify problems and another day to verify fixed ones. The expert would still leave quite a few excess alarms for the start-up period of the new schedule, and the simulator leaves none at all.

The verification of a new train schedule with the simulator takes only 2 (man * day), a significant reduction from 1 (expert * month) intensive manual schedule debugging. This is a cost reduction of 1 expert * month = US\$20K * month * time / year = US\$20K / year.

6.5. Better dispatcher training

The better the system quality gets, the less opportunity of schedule disruption, hence the less opportunity of dispatcher OJT. The simulator is able to create any situation to get rail dispatchers

ready for most difficult times. It also reduces training standby time.

The dual quality improvement of software and training doubly reduced the stress on dispatchers, and prevented economical loss.

Zero major trouble for the three years since the new PRC computer was put into operation, as opposed to 10 problems for the first two years of the previous PRC, prevented possible stress on personnel, man-month waste, and economical loss caused by delay.

It is difficult to quantify the effect of the above mentioned reduction of training standby time and training quality improvement, but the saving can be roughly estimated US\$1M / year. The control software improvement prevents possible fare refund for delay caused by poor quality of the control system, estimated as US\$1M * 5 times / year = US\$5M / year.

6.6. Test of New Facility Plans

Plans to introduce the Asa station, the Shinagawa station, and the new performance 500 series and 700 series bullet trains were tested on the simulator with planned schedules before design decision, and subsequent investment in physical facilities. It is also difficult to quantify the economical effect of those advanced tests, but the risk saving is estimated at least US\$1M.

Note: All of the above money figures are the estimation of the author, not authorized by the user, Central JR.

7. CONCLUSIONS

Train operation simulators are progressing to have more capabilities and better fidelity, but the one built on the PIM is still unmatched by others in the regard of fidelity, system size, and real-time.

The high fidelity bullet train operation simulator built on the PIM cost US\$2M-3M, but the user was able to save more than that and thoroughly test new control schemes, which was impossible without the simulator.

8. ACKNOWLEDGEMENTS

The author wishes to thank the staffs at the Tokyo information system development department of Central JR for their kind collaboration and providing me with all available data.

9. REFERENCES

- Honda, S., Y. Nishida, High Fidelity CTC Simulator Development, *JREA*, Vol. 41, No.5, 25398-25401, 1998 (in Japanese).
- Kon, N., CTC Simulator Development on a Parallel Processing System, *Cybernetics*, Vol. 4, No. 3, 1999 (in Japanese).
- Matsumoto, M., *Electric Railway*, Morikita Shuppan Co. Ltd., 88-113, 200-229, 243-264, 1999 (in Japanese).
- Mochinaga, Y., and IEEJ Education Survey Special Committee on Electric Railway, *Latest Technologies of Electric Railway*, Corona Publishing Co., Ltd., 78-93, 209-257, Tokyo, Japan, 2000 (in Japanese).
- Nozue, N., Simulations Applied to Railway, *Railroad and Electrical Technology Japan*, 7(6), 5-11, 1996 (in Japanese).
- Parunak, H. V. D., Industrial and Practical Applications of DAI, *G. Weiss, Editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 337-421, Cambridge, MA, 1999.
- Saito, M., and S. Suzuki, PRC Simulator Development for Tokaido-Sanyo Shinkansen, *Traffic and Electric Railway Workshop*, IEEJ, TER-96-13, 1996 (in Japanese).
- Yaskawa, S., Complexity and the Central Japan Railway, *Embracing Complexity*, The Cap Gemini Ernst & Young Center for Business Innovation, 59-66, 1997a.
- Yaskawa, S., and S. Yamamoto, A Real-time Parallel Processing System, PIM, and its Application to a Large-Scale Railroad Operation Simulator, *International Congress on Modelling and Simulation Proceedings, MODSIM 97*, 1997b.
- Yaskawa, S., and A. Sakata, The Application of Intelligent Agent Technology to Simulation, *International Congress on Modelling and Simulation Proceedings, MODSIM 99*, 1999.
- Yaskawa, S., Bullet Train Operation Simulator including Electric Power Consumption, *International Congress on Modelling and Simulation Proceedings, MODSIM 2001*, 2001.