Development of control simulation system in which 3-D CAD data is used

●Kenboku Arai ●Takao Iwasaki ●Tsutomu Goto

●Yuichi Sato ●Masayoshi Hashima ●Yosuke Senta

Abstract

Since 1997, we have adopted the usage of 3-D CAD (Computer Aided Design) for the development of vehicle-mounted deck mechanisms such as CD changers. Thereafter we have achieved considerable effects in optimizing the design especially for component parts, by taking advantage of 3-D data for the stress analysis based on CAE (Computer Aided Engineering), the verification of shapes, and the check of static interference between assembled parts. However, 3-D CAD data has not been fully utilized for the verification and review of the dynamic functions of the deck mechanisms.

We thus developed a simulation system in which a virtual model created on a PC by using 3-D CAD data, which can be controlled in the same way as the real deck mechanism.

Making use of this system has made it possible to carry out dynamic verification for the structure of the mechanism from the standpoint of "control of mechanism" effectively even at the early step of designing and evaluating the control software without the real mechanism.

This paper outlines the background of the development, functions of the simulation system, and effectiveness in its utilization.

1. Introduction

In order to respond to increasing demands for multi-featured, downsized, and slimmed-down systems, it has become necessary to improve the accuracy of component parts and increase their density. The same holds true for vehicle-mounted deck mechanisms, including CD changers, being developed by our company. In order to make the functions of the mechanisms reach their full potential even in the limited space of the product, it has become necessary to increase the density of the parts of the mechanism. The system configuration has also become complicated.

Aiming to cope with the growing complexity of design, shorten the development period, reduce costs, and improve the design quality, we have adopted 3-D CAD "Pro/ENGINEER" *1 for development of the deck mechanisms since 1997.

Since then, 3-D models have been used for verifying the shape and checking static interference between assembled parts. We have achieved considerable effects in optimizing the design especially for component parts, by widening the scope of application of 3-D data to the stress analysis, oscillation analysis, resin analysis, etc. based on CAE.

However, we have not fully utilized the 3-D data for verifying or reviewing dynamic functions of the deck mechanisms.

Items to verify and review dynamic functions of the deck mechanisms by using the 3-D data are roughly divided into the following two categories.

1) The movement characteristics of the mechanisms and functions that require strict analyses on dynamic structure

2) The structural movements, operability, assembling properties, geometric movability, and controllability that do not require strict analyses on dynamic factors

The simulation system to be explained in this paper was developed based on the 3-D mechanism simulator manufactured by Fujitsu Ltd., which exhibits excellent functionality especially for verification of the properties mentioned in "2)."

This is a simulation system in which a virtual model created on a PC by using the 3-D data can be operated and controlled as if it were a real deck mechanism.

This system has enabled us to effectively examine the controllability and control specifications of the deck mechanism even at the early step of design while reviewing and verifying the properties mentioned in "2)" as well as "control of the deck mechanism" to be explained in this paper. In addition, the control software that has already been designed can be verified without relying on the prototype of the deck mechanism.

Research and development of this simulation system was commissioned to Fujitsu Laboratories Ltd., and our requests have been extensively incorporated in the system.

First, to provide some background on the development of this system, we would like to refer to some problems that have arisen in the conventional development processes of the control software and the deck mechanisms.

2. Problems that have arisen during the development of control software

With the system configuration of the deck mechanisms increasing in complexity, the control logic of the built-in type software (firmware) to optimally control the mechanisms has become more complicated. In addition, with the system attaining greater sophistication, the number of software programs has been increasing year after year. Under these circumstances, it has become necessary for us to wrestle with the problem of how to develop the software with efficiency.

The basic flow of software development is shown in Figure1.

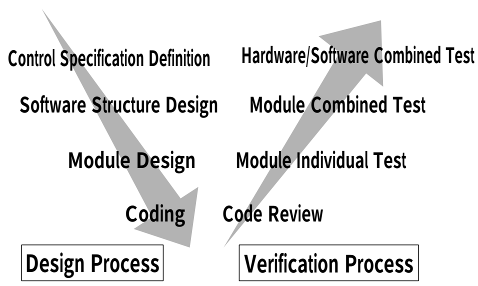


Fig.1 Software Development Process

The development process consists of the following two processes.

*1Pro/ENGINEER is a trade name of Parametric Technology in the U.S.

- Design process in which the software is produced while design specifications are being developed in a step-by-step manner after the control specifications are defined.
- Verification process in which quality of the software is verified in light of the design specifications drawn up according to each step

In this serialized development process, the following problems have so far arisen.

2.1 Problems that have arisen in the design process

In the design process, it took considerable time to determine the details of control specifications, causing difficulties in determining the final specifications of the software.

What kinds of component parts are to be employed to make up the deck mechanism and how each component part is to be activated so that functions of the deck mechanism will be fully taken advantage of depend on the design intention of the mechanical designer. Accordingly, the setting of the timing to drive the motor and to carrying out the control by detecting the sensor signal are to be defined by the mechanical designer and are to be specified in the primary control specifications.

Then, the software designer is to set about the software design, based on the specifications provided by the mechanical designer.

However, problems over the details, including ambiguities, inconsistencies, and omissions, are often found in the primary specifications. (These problems are not confined to the control of the mechanism.)

Accordingly, the primary procedure to be carried out by the software designer is to analyze the specifications and the mechanism to be controlled, in terms of "control." Then, inconsistencies, etc. are to be ironed out, and detailed specifications are to be redefined with ambiguities eliminated. In some cases, it will become necessary to propose a change in the system configuration of the deck mechanism.

These procedures, which are referred to as a design process review of control specifications and controllability of the mechanism to be controlled, are to be carried out in cooperation between the software designer and the mechanical designer.

However, it was extremely difficult to carry them out under the conventional development environment. This is because software designers had difficulties in understanding the system configuration of the deck mechanism (to be controlled) during the design phase only by the plane CAD drawings or timing charts. This tendency is being accelerated, with the mechanical configuration of the deck increasing in complexity.

Since it was practically impossible to conduct thorough analyses during the design phase, there were many cases where only basic control specifications were defined first, and detailed specifications were determined later based on the analyses made by using the prototype of the deck mechanism.

2.2 Problems arising in the verification process

As shown in Figure 1, the control software that has been designed is to be tested in a step-by-step manner. The software is to be evaluated individually first, and to be verified while it is built in the deck mechanism.

The following problems have arisen in this verification process.

1) The full-scale built-in test cannot be carried out until the deck mechanism is completed.

In general, it takes more time to build a prototype of the deck mechanism than to produce control software, because prototyping requires not only lead time to arrange manufacturing of parts but also adjustment work for assembling. As a result, in some cases, the prototype will not have been completed when the software stands ready to be evaluated. In other cases, the design program for the control software of the mechanism will be delayed (or given a lower priority) on purpose in order to adjust the timing in consideration of the completion time of the prototype. Whatever the case may be, it will take a con-

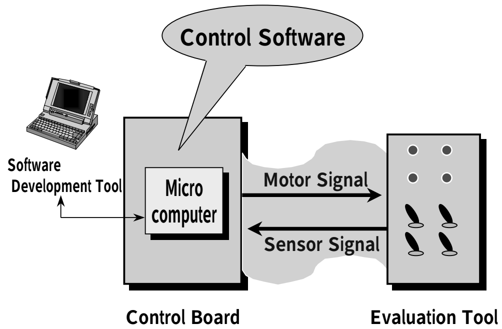


Fig.2 Software Evaluation by Evaluation Equipment

siderable time before problems are identified by the built-in test and measures are taken against them, thus delaying the overall development.

If the control software is designed to control the simple deck mechanism in which only the lever mechanism is to be positioned and controlled by one motor and several sensors (e.g.: CD player's mechanism for insertion/ejection of discs and open/close mechanism for the liquid crystal display panel of navigation system), it is possible to verify whether the software meets the control specifications, by carrying out the input/output test with an appropriate evaluation tool as shown in Figure 2 instead of the completed deck mechanism. However, since it is difficult to evaluate the complicated deck mechanisms made up of motors and sensors, including CD changers and MD changers, we have to prepare an evaluation tool specifically designed for the input/output test as occasion demands or have to use the completed deck mechanism. Worse yet, we are not able to verify, only by carrying out the input/output test in which the evaluation tool is used, whether the control specifications per se are appropriate or not.

2) If the prototype is broken or out of order, the verification work will be interrupted.

Since the levels of perfection of the deck mechanism and the control software are low at the early step of the development, the prototype will often be broken or out of order due to bug-ridden software or faulty design of the deck mechanism. If this situation occurs, it will be necessary to track down the cause and repair the prototype or make an arrangement for a substitute, thus causing loss of time. The verification of the control software could be interrupted for a while.

3) It is not easy to verify whether the control software works under abnormal conditions.

When the completed deck mechanism is used, it is relatively easy to check the basic functions under the assumption that the mechanism would work normally or would be manipulated normally, However, it is not easy to verify the functions under the assumption that the mechanism would be under abnormal conditions or would be manipulated in a wrong manner. Abnormal conditions of the deck mechanism include a breakdown of the motor, interference with the mechanism elements, breakdown or malfunction of the sensor, and stack of discs. In consideration of these abnormal conditions, the control software is designed to contain many recovery functions and fail-safe functions, which account for 70 to 80% of all the functions of the software.

In order to verify these functions of the control software, it is necessary to reproduce the abnormal conditions efficiently and repeatedly.

When using the completed deck mechanism for making the evaluation, we have to create the intended conditions of the mechanism by forcibly stopping the operation of the mechanism, manipulating the internal switches in an unexpected manner, rotating the gear with a finger, or disassembling and modifying the mechanism.

4) Regarding factors responsible for discrepancies, it is not easy to distinguish between those attributable to the deck mechanism and those attributable to the control software.

Since the operating state of the prototype tends to be unstable at the early step of development, in some cases it is difficult to determine whether the discrepancies are attributable to the deck mechanism or attributable to the software. For example, it is not easy to distinguish between a malfunction of the sensor and that of the detection system of the software. When the frequency of reproduction of a problem is low, we often find that discrepancies in fact have been caused by the deck mechanism, after spending considerable time analyzing the possible causes attributable to the software.

3. Problems arising during development of the deck mechanisms

In most cases of conventional development of the deck mechanisms, we have built a prototype before reviewing the functions. As previously mentioned, this is because it was difficult to analyze the system configuration of the deck mechanism in terms of "control" in cooperation between the mechanical designer and the software designer or it was difficult to review it, in advance, with other persons who are parties to the development.

As a result, problems arose during the review process after the prototype was built, thus bringing about the necessity of not only change in the design of the deck mechanism but also re-prototyping.

Consequently, it can be said that heavy dependence on the prototype has caused considerable loss in the development of the control software and the deck mechanism.

Examples of major problems related to the system configuration of the deck mechanism and its control are shown in Table 1.

While working toward widening the scope of application of 3-D CAD data, we found that utilization of the 3-D data would be effective in solving the previously-mentioned problems related to "control of the mechanism" and those arising during the design and verification processes of the control software. In consequence, we have devised and developed the following simulation system.

Table 1 Examples of Major Problems and Solutions

Example of a Problem	Solution
When in a particular mechanism state, it becomes impossible to return to a normal status or to initialize the mechanism status.	･Review mechanism structure ･Alteration of control specifications (addition of a failsafe control)
From a problem in sensor configurations, multiple mechanism states cannot be identified, and cannot be controlled.	･Review sensor configuration ･Alter control specifications

4. Simulation system and its utilization

The simulation system to be explained here has been developed based on the 3-D mechanism simulator, "FJVPS (Fujitsu Virtual Product Simulator)" manufactured by Fujitsu Ltd.

In the order for the development procedures in which the virtual model to be created by using this simulation system is used, functions of the system and effects brought about by its utilization are to be explained.

4.1 Creation of a virtual model based on 3-D data

First, a virtual model of the deck mechanism is to be created using 3-D CAD data.

1) Part shape data designed by using 3-D CAD and assembly information is to be converted to polygon data, etc. by a conversion tool, and to be integrated into the system.

2) As shown in Figure 3, the movement of rotating parts and sliding parts, as well as the relation between parts and gears, cams, the groove mechanism, the clutch mechanism, etc., are to be set.

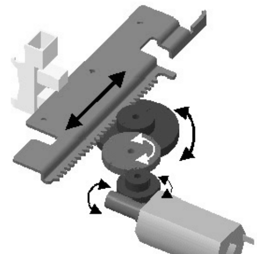


Fig.3 Movement Settings of Individual Parts

The settings mentioned above can be made by the basic functions of the mechanism simulator.

In the development of this system, we have incorporated new functions for extending the mechanism definition and carrying out the control simulation into the basic functions mentioned above.

3) Backlash of gears and the play of the groove mechanism are to be defined.

These definitions can be given by determining the relation between the settings mentioned in "2)" and hysteresis. (Figure 4)

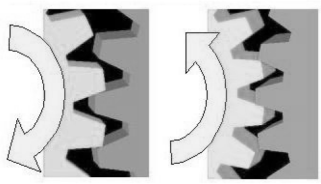


Fig.4 Gear Backlash

4) Movements of the model being transferred are to be defined.

Loading, ejection, and replacement of several discs are major operations of the CD changer mechanism. In order to model these movements, it is also necessary to model the movements of discs being transferred, as well

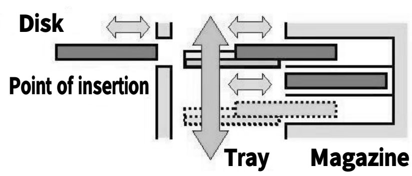


Fig.5 Movements of the Model Being Transferred

as the movements of the mechanism as a mechanism component part. As shown in Figure 5, the movements of the model being transferred can be defined by this system.

5) Definitions of a motor model and a sensor model, both of which serve as an interface with the control software, are to be established.

In order to "control" the virtual model of the deck mechanism, a motor model that drives the system according to the control signal output from the control software and a sensor model that inputs data into the control software are required. The sensor model is to output the sensor signal according to the operation of the deck mechanism. Two types of motors, a DC motor and a stepping motor, are provided as motor models for this system. Each model of ON/OFF switch, potentiometer, and encoder is provided as a sensor model.

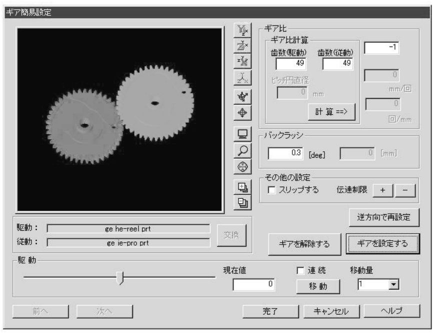


Fig.6 Example of Setting Dialog

a) CD Changer Mechanism

Required man-hours: Approximately 20 hours

Model Dimensions:

Number of parts	1,067
Number of Polygons	557,486
Number of joints(rotating/sliding)	103
Number of Motors	5
Number of Sensors	13

b) Cassette Player Mechanism

Required man-hours: Approximately 8 hours

Model Dimensions:

Number of parts	350
Number of Polygons	192,830
Number of joints(rotating/sliding)	46
Number of Motors	2
Number of Sensors	6

Fig.7 Example of Model Dimension and Time for Settings

At this point, creation of the virtual model is to be completed. These settings can be made without difficulty by inputting the necessary parameters into the dialog boxes (See Figure 6.) to be successively displayed according to each setting item. The dimensions of our deck mechanism models and actual man-hours needed for modeling are shown in Figure 7. The models can be created in a relatively short time.

In addition, it almost seems superfluous to say that these procedures can be carried out by the mechanical designer.

4.2 Review of the deck mechanism by the virtual model

Virtual models created on a PC can be manipulated without restraint as if they were real deck mechanisms. For example, any gear or lever mechanism can be moved in real time by dragging a mouse. Furthermore, change in the location of the eye-point, display of the sectional view, disassembly or replacement of parts, etc. can be carried out without restraint.

Introduction of high-speed graphics has made it possible to manipulate even a large-scale model quite normally. Thanks to the high-speed interference checking function, we can check the dynamic interference and the clearance with the deck mechanism in operation.

Since even persons other than mechanical designers can master the manipulation method with ease, review of the functions can also be effectively carried out by those in charge of manufacturing, sales, service, etc.

4.3 Software designer's analyses on the deck mechanism to be controlled

Software designers can also manipulate the simulation system quite easily. They can analyze deficiency or correctness of the control specifications according to the primary control specifications presented by the mechanical designer while "controlling" each part with a mouse, etc.

They can also conduct the analysis while observing the internal conditions, with the sectional view of the system configuration of the deck mechanism displayed or with external components displayed in a state of transparency/semi-transparency. This kind of work cannot be carried out with a real deck mechanism.

By conducting work like this, thorough analyses on the system configuration and control specifications can be made from the standpoint of "control."

Problems tracked down during the analyses are to be fed back to the control specifications and the design of the deck mechanisms. They can be corrected without difficulty because the mechanisms are still in the designing stage. While a cycle of verification and change is being repeated by using the virtual models, the design for details of the system configuration and the control specifications can be brought near to perfection.

4.4 Preparations for control simulation

The software designer is to design the software and perform the coding, on the basis of the control specifications determined under the procedures mentioned above. The functions of the control software can be verified by the control simulation function to be provided by this simulation system.

In order to carry out the control simulation, the following settings and preparations, as well as the previously-mentioned establishment of the definitions of motor models and sensor models, are required.

1) Preparations for synchronization with the simulator

During our experiments, the simulator's response presented no problem when the review was carried out while the virtual model was being manipulated by a mouse. However, the slow processing speed of the PC turned into a problem when the virtual model was connected to the software running in real time.

To solve this problem, a method of bringing the control software into synchronization with the simulator has been adopted for this simulation system. Under this method, the control software is to perform control processing in tune with the processing speed of the simulator. While sending/receiving a synchronization signal, the control software and the simulator perform the processing alternately. For example, after the simulator carries out calculations for 10msec, the control software is to perform the processing for 10msec. (See Figure 8.) By repeating this cycle, the simulation can be achieved on an accurate time-scale, though it will be carried out at a slow pace. To cite one example, a simulation with a cycle of 10msec was achieved, though it required about four times as much time as the real model.

Although it cannot be ensured that this simulation system will provide a simulation perfectly in real time, the system will considerably help us to thoroughly check the control exercised by software and the detailed behaviors of each model.

A software logic circuit is required to be incorporated into the control software in order to maintain synchronization with the simulator, but the size is diminutive. It is also necessary to program the conditions for synchronization into the simulator.

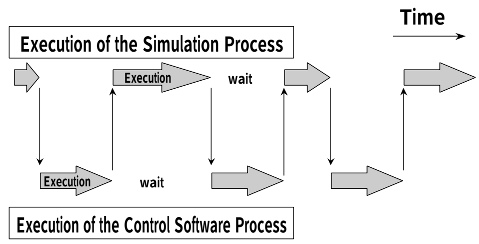


Fig.8 Synchronization of Simulator with Control Software

2) Connection between simulator and control software

A virtual model created on a PC is to be connected to the control software via the general purpose I/O board installed in the PC. Signals to be input/output are motor control signals, sensor signals, and the above-mentioned synchronization signals. Signals other than the synchronization signal are to be input in the same way as done to link the control software and the deck mechanism in completed products.

Meanwhile, the control software is to be operated on the microprocessor installed in the control board or on the software development tool connected to the board.

4.5 Control simulation

After completion of the settings mentioned above, it will become possible to carry out the control simulation by using the virtual model. Although a completed deck mechanism was required for the conventional built-in test (Figure 9), it has become possible to conduct the verification without using a real mechanism, by controlling the virtual model created on this simulation system. (Figure 10) This control simulation in which the virtual model is connected to the real control circuit is referred to as HIL (Hardware In the Loop) simulation.

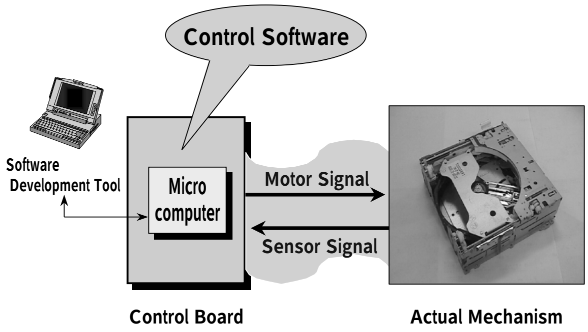


Fig.9 Combination Test Using Completed Mechanism

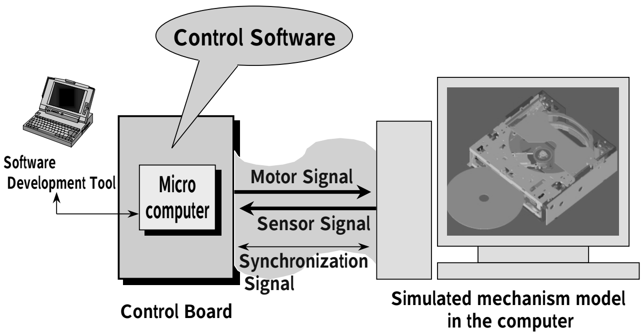


Fig.10 Combination Test Using Virtual Mechanism Model

4.6 Verification of control timing

In the control simulation, the control timing is defined as one of the important evaluation items. As previously mentioned, it cannot be ensured that the control simulation will be carried out perfectly in real time and on a linear time scale by this simulation system. Since measurement instruments "currently in use" cannot be used, logic analyzer functions have been incorporated into this simulation system so that the system will capture, on an accurate time scale, the signals to be input/output during the simulation. This has made it possible to determine the timing of input/output signals on an accurate time scale. (Figure 11)

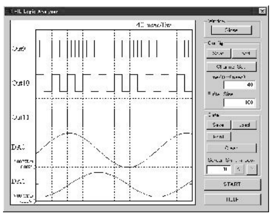


Fig.11 Logic Analyzer Function

4.7 Reproduction of abnormal conditions, etc.

Abnormal conditions of the deck mechanism, which are difficult to be reproduced by the completed model, can be reproduced easily by this simulation system. Failures of the motor or sensor (not in operation) can be reproduced without difficulty by using the dialog boxes for setting the motor model and sensor model. Chattering, which is expected to create problems for ON/OFF switches, can also be generated in the sensor model under any given condition. (Figure 12)

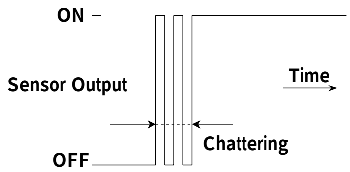


Fig.12 Chattering Generation

Interference or bounce in the mechanism, manipulation intended for causing abnormal conditions, etc. can be reproduced with ease. Retry processing of the control software and the behavior of the system caused at the time when the movement of a disc is forcibly restrained are shown in Figure 13 as an example.

Reproduction of abnormal conditions has facilitated the verification of the recovery control of the control software.

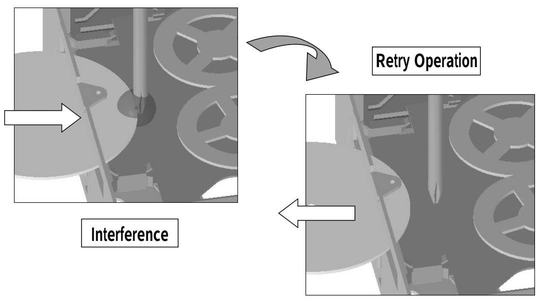


Fig.13 Retry by Control Software

By manipulating parts, any conditions of the deck mechanism can be created in this simulation system. In addition, these conditions can be registered and invoked, facilitating reproduction of the conditions of the mechanism.

Consequently, it has been proven that the control software can be verified by making use of functions of this simulation system, without relying on the prototype of the deck mechanism.

4.8 Adjustment and final check by the completed deck mechanism

Since this simulation system does not handle dynamic properties in the strict sense, the operation timing of the virtual model is different from that of the completed mechanism.

In some cases of control of the mechanism, braking time and driving time are to be determined in consideration of characteristics of the mechanism. However, it is necessary to make the final adjustment for control parameters of the software by using the completed deck mechanism in order to check the operating state.

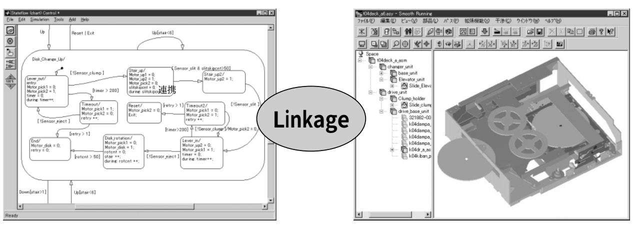
5. Linkage with status transition analysis/Design tool

Furthermore, this simulation system is equipped with the following functions.

Carrying out simulations in the early parts of the development process is effective in improving the quality of control software even at the early development stage and minimizing loss during development.

With the development of software increasing in complexity and in scale, designing software while analyzing and defining the behaviors of each software program by using the status transition chart/table is effective in development. This simulation system provides an interface function that has a linkage with the status transition analysis/design tool, MATLAB/Stateflow *2 or ZIPC *3, which is highly expected to support these design procedures. (Figure 14)

Utilization of this function will make it possible to carry out the connection test by using the virtual model even at the software design stage and to ensure the quality at a much earlier stage.



Status Transition Analysis/Design Tool　　Current Simulation System

Fig.14 Linkage to State Transition Analysis/Design Tools

6. Effects of this simulation system on the development

The conventional development process that has relied on the prototype of the deck mechanism is shown in Figure 15, and the characteristics and effects of the development process based on this simulation (shown in Figure 16) are as follows.

1) Each design procedure can be carried out in collaboration between the software designer and the mechanical designer even at the early design phase.
2) Despite the complexity of the system configuration, control specifications and detailed behaviors of the deck mechanism (to be controlled) can be analyzed by the software designer. As a result, problems related to control can be tracked down earlier and fed back to the design.
3) Even before the prototype of the deck mechanism is completed, the control software can be verified, and the mechanism and the control software can be concurrently developed.
4) Verification of the software will not be interrupted due to breakdown or failure of the deck mechanism.
5) As opposed to the completed deck mechanism, abnormal conditions can be reproduced with ease, and verification of the operating state can be carried out efficiently.
6) High reproducibility makes it easy to track down the cause of problems.
7) Regarding the development of the deck mechanism, since the operating state in relation to the control system can be checked before the release of the drawing, the loss cost resulting from returning and re-prototyping to take measures against problems can be minimized.

The simulation system developed this time has made it possible to design the deck mechanism in step with the control software, develop them concurrently, and carry out the test in nearly real time with the virtual model of the deck mechanism and its control software connected.

---

*2 MATLAB/Stateflow is a trade name of MathWorks.
*3 ZIPC is a trade name of CATS INC.

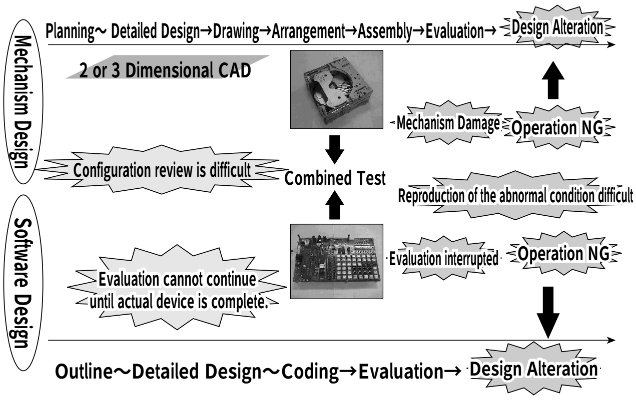


Fig.15 Conventional Development Process

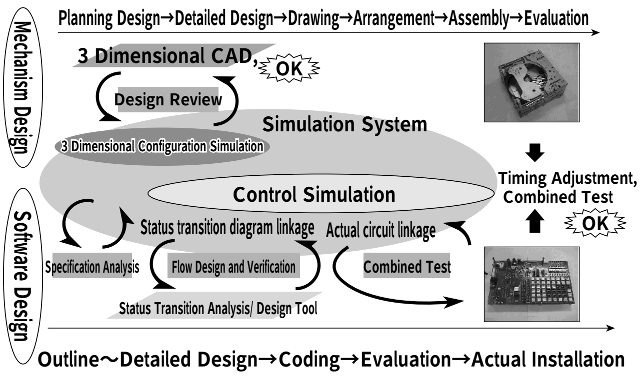


Fig.16 Development Process Based on This System

It is also expected that utilization of this system will help to reduce man-hours by 30 to 40%.

7. Conclusion

After partially introducing this simulation system into verification of the deck mechanism under development, we came to have the prospect of significantly enhancing efficiency in development of the deck mechanism by adopting this system. From now on, we would like to work toward further improving the quality of the deck mechanism and enhancing efficiency in development, by pressing ahead with digital engineering in which this simulation system and other simulation systems (for stress analysis, oscillation analysis, etc.) will be fully taken advantage of.

The simulation system explained in this paper is to be introduced on the market by Fujitsu Ltd.

Acknowledgement

We would like to extend our sincere gratitude to all who are parties to Fujitsu Ltd. and Fujitsu Laboratories Ltd. for their generous support and cooperation given to the development of this simulation system.

References

1) Built-in software development solution, FJVPS/HIL, ＦＩＮＤ, Vol.19,Nov.1,2001 (Compiled by Fujitsu Ltd.)
2) Construction of the built-in software development support system (VPS/HIL) (Written by Hashima and others) and collection of research papers published at the 10th lecture meeting of the design engineering/system section of Japan Society of Mechanical Engineers(pp148-149,2001)

Profiles of Writers

Kenboku Arai

Short History: Entered Fujitsu TEN Ltd. in 1981, working in development of software for devices using microcomputers, cassette and CD player software development. Currently in the System Testing Section Manager of Deck Mechanism Engineering Department at A.V.C. Products Group.

Takao Iwasaki

Short History: Entered Fujitsu TEN Ltd. in 1986, working in development of firmware for car audio systems. Currently in the DF Project of Deck Mechanism Engineering Department at A.V.C. Products Group.

Tsutomu Goto

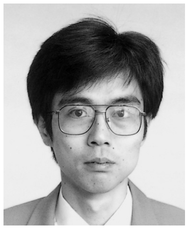
Short History: Entered Fujitsu TEN Ltd. in 1986, thereafter working in the development of cassette players. Currently in the System Testing Section of Deck Mechanism Engineering Department at A.V.C. Products Group.

Yuichi Sato

Short History: Entered Fujitsu Laboratories Ltd. in 1986, working in R&D of control and simulation technologies. Member of Robotics Society of Japan (RSJ), Information Processing Society of Japan. Currently in the Autonomous Systems LAB.

Masayoshi Hashima

Short History: Entered Fujitsu Laboratories Ltd. in 1991, working in R&D of robotic visual sensory / measurement systems, and simulator technologies. Member of the Robotics Society of Japan (RSJ). Currently in the Autonomous Systems LAB.



Yousuke Senta

Short History: Entered Fujitsu Laboratories Ltd. in 1997, working with development of HIL simulators for VPS and magnetic disks. Member of Japan's Society of Mechanical Engineers and Society of Instrument and Control Engineers. Currently in the Autonomous Systems LAB.