'94 KACC (1994. 10. 17 ~ 20)

A Development of GEO Satellite Ground Control Softwares

H. J. Lee, J. M. Kim, and S. J. Chung

Satellite Communications Technology Division, ETRI
Yusong P. O. Box 106, Taejon 305-350, Korea
Tel. (042) 860-5769, Fax. (042) 860-6430

ABSTRACTS

To provide more instructive and a safer ground control operation environments for satellite operators, and subsequently to implement a better look-and-feel user interface and a structural mechanism to enhance the efficiency of control and monitoring facilities, we have developed a prototype(laboratory model) ground control softwares targeting for the first generation KOREASAT scheduled to be launched in 1995. As far as the functionality is concerned, the developed system is covering almost all the mission phase operational functions except for some functions like antenna tracking control that are necessary for real operation environments. Most of the functions of the system is realized in softwares but some hardwares needed for TM/TC RF communications are also included in it. The system is now being integrated and under the system test. The performance and functionality is to be evaluated by the end of this year by using the satellite software simulator. Next year, this system could be configured to be used as a workbench for a on-line/off-line analysis of the operating KOREASAT satellites.

I. INTRODUCTION

To provide more instructive and a safer ground control operation environments for satellite operators, and subsequently to implement a better look-and-feel user interface and a structural mechanism to enhance the efficiency of control and monitoring facilities, we have developed a prototype(laboratory model) ground control softwares targeting for the first generation KOREASAT scheduled to be launched in 1995. From the bare foundation, we designed and developed for ourselves all the major softwares based on the KOREASAT spacecraft specification. The system development was initiated as an R&D activity to prepare a technical foundation for the satellite ground control system to make our own control systems for the next KOREASATs and others. As far as the functionality is concerned, the developed system is covering almost all the mission phase operational functions except for some functions like antenna tracking control that are necessary for real operation environments.

Ground control and monitoring of a satellite in an orbit needs two kinds of processings; real-time data processing and flight dynamics processing. The telemetry from the spacecraft downstreams in the period of one second order(for KOREASAT 2 seconds) to the ground in the form of radio frequency signals. After some electromagnetic processing like demodulation, the data stream is recovered and fed to the real-time processors. Regardless of system workload or operating environments, it is required to process the input data stream completely before the next telemetry arrives. Of course, telecommands made on the ground should be transmitted in the meanwhile if needed within each processing time slot to the spacecraft to give effective actions. For this reason, a real-time processing mechanism is employed. The subsystem that does these functions which are characterized to be completed in a given time slot is called real-time processing system. On the other hand, the orbit and the attitude of the spacecraft should be evaluated and controlled periodically to be maintained in the state defined for a given mission. Even though the spacecraft is subject to disturbances by several external forces, its digress from the nominal situation is not so fast that the ground processing is not always required to respond as fast as the real-time processing, while this function needs more sophisticated calculation and more accurate results. This kind of functionalities requires a subsystem called flight dynamics processing subsystem. In some cases, however for instance, when the spacecraft loses its pitch-axis lock, a closed-loop ground control is inevitable to recover the spacecraft attitude. This function is required to be performed on a real-time basis nevertheless of its categorization. However, a time delay incurred in command signal transmission and in telemetry signal processing makes an accurate ground control difficult so that a real time loop control is either implemented on the basis of a simpler modeling considering a margin for compensating time delay or in a hardware configuration relying upon hardware signal synchronization. In this paper, we presents the system architecture and the processing structures first, then describes those developed subsystems briefly in view of features and functionalities.

II. SYSTEM ARCHITECTURE

The developed system has the hardware architecture shown in Figure 1. It consists of one main computer and three workstations. The real-time processing subsystem has one main computer, Vax 4000-300, for TM(telemetry) processing and display and TC(telecommand) transmission. Also it has two workstations, Vax 4000-60's, for operator control and monitoring respectively. The remaining workstation is allocated to the flight dynamics processor. Two color X-terminals are also assigned to the analysis facilities for TM data and flight data respectively. TM page display monitors consist of 3 color 14" monitors located in the console as well as 2 color 19" monitors hanging from the ceiling. The computers are connected to the LAN via Ethernet 802.2 with the DECnet network protocol. The satellite software simulator hosting workstation is used as control target which provides telemetry streams through RF links and responds to the telecommand came into it through RF links.

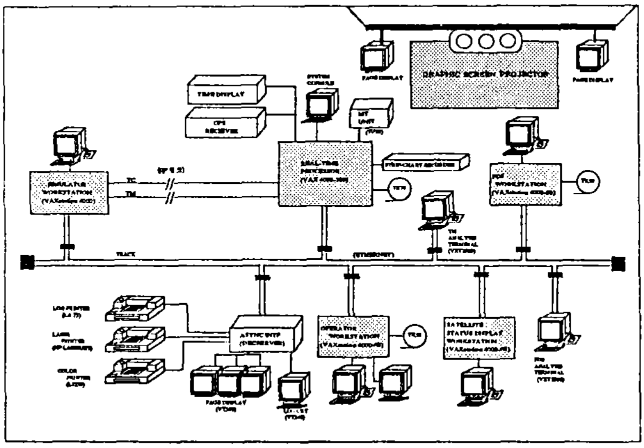


Figure 1 The overall hardware architecture of the system

(1) SYSTEM SOFTWARE STRUCTURE

The system employs a multiprocess software structure for both real-time data processing and flight dynamics processing so that a concurrent processing can be possible for several input data streams which leads to an enhancement of system throughput and processing capacity. Specially for the flight dynamics processing, contrast to the traditional batch-style processing, this structure enables real-time estimation functions and control functions to be incorporated into the system in a straightforward way. Furthermore the flight dynamics operator or analyst can access to several tasks at one time to observe each result generated from a number of different function windows. Another advantage of this scheme is that common functions, like thruster calibration or sun-moon ephemeris calculation, can be performed once and whose results can be saved to be used subsequently several times for other processings. This feature makes unnecessary the re-visits to those functions for each flight data calculation otherwise inevitable.

The system has also adopted a distributed processing structure so that it can perform data exchange and data processing over a computer network. System functions are so distributed to several CPU's that each computing resource has it own processes and functions specific to the role, and configuration of the each CPU. The interfaces between any of two processes in the system are messages exchanged either in a local way or over a network. There is no functional overlap between processes running in one CPU. Thus a system function, for example, telemetry display, is implemented in a way of a combination or sequential chaining of individual subtasks allocated to that function via message communication. Furthermore since we designed all the system functions to be partitioned and implemented in a modular way, it is very easy to reconfigure the functions in the system without giving any significant effects to the rest of the system. Therefore the system function can be easily expanded or reallocated to the CPU's by re-defining or re-configuring the interface messages according to the requirements. Additionally, owing to this feature, an independent and concurrent development of each process has been successful as well as efficient. The integration and the test also have been proved even easier and less time-consuming as predicted in [1].

In virtue of this distributed software and modular structure, the overall workload of the system can be easily distributed and shared, and subsequently the system capacity can be easily grown by duplicating the processes into the new CPU while making the new node do the same functions as the existing ones. Contrast to this expansion, the system can be also downsized and made more economical by re-populating the distributedly located processes in a fewer number of computers.

The real-time processing is based on the database-driven processing structure. TM and TC have their own internal structures and characteristics respectively and these information is realized in a form of database. Since display format and configuration, item layout, and graphic parameters can be also formulated, these are translated into a database as well. Moreover even in the flight dynamics processing subsystem, the spacecraft configuration, performance, and constants can be gathered to form a database. The separation of data from the application program enhances the portability and migration efficiency. Figure 2 shows the overall software processing structure where a distributed multiprocessing feature can be observed.

The system uses VMS as an operating system and the real-time property is supported by the VMS real-time scheduling mechanism. The real-time processes are assigned with higher priorities than other processes. As characterized by the VMS, the startup processes in each CPU of the system create several detached processes and some subprocesses.

In addition, X-based graphics and desktop-style look-and-feel user interfaces are implemented to enhance the readability of the various output and to minimize the possibility of erroneous input[2][3]. Telemetry displays, ground system control, configuration/layout control, and satellite commanding is using X-window graphics and input/output widgets wherever applicable. Page display and four kinds of graph-style display as well as alarm summary display are provided for telemetry monitoring. Trend graphs are also available for a historical trending analysis for satellite telemetry monitoring points and for satellite flight dynamics data.

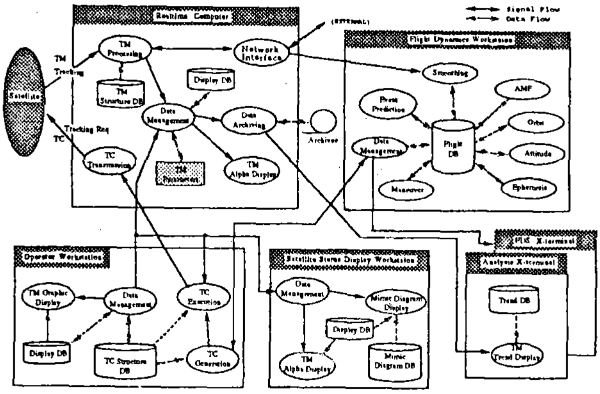


Figure 2 The overall software processing structure of the system

(2) PROCESSING STRUCTURE

As stated earlier, each process can perform several tasks upon request initiated from input messages. Each CPU constituting the system is classified as a software unit which characterizes the CPU populating a predefined number of processes that no network communication is needed among them and a physically separated from other CPU's. The term "signal" is used for referring to the message between processes. In each process there is an application signal handler which receives signals from either a local process or a remote process and activates the corresponding tasks according to the input signal. After the designated processing task is complete, the results can be displayed on the connected devices or can be formulated as a signal ready to be sent for next processing. Then the process returns to the waiting state for new input signals. Since each signal is mapped to the predefined processing routine, a system function can be represented a serial combination of signal chains and related processings. Due to this structure, system functions can be modified or expanded very easily only by either modifying or inserting both the signal and its corresponding routine while keeping the rest of the system unchanged. Thus this enables the system to have a great deal of flexibility and viability against any changes of the functional requirements.

All the signals are received and processed in an interrupt processing level or AST(Asynchronous System Trap) level in the VMS terminology. This implies that each process has its own base level scheduler that processes user input as shown in Figure 3. When there is no input signal, the program control remains in the base level. The processes residing in the workstation units have MOTIF graphic user interface. Some processes displaying the graphics objects like mimic diagram or graphs have additionally a third party SL-GMS interface which is an X-based application-level graphic interface. In these processes, application callback routines take charge of user input processing. For the processes that do not have any X-graphic user interface, the base level scheduler is reduced to a dummy endless loop. Application libraries provide the application programs with the system function interface such as signal reception and transmission.

The signal communication mechanism used in the system is basically the mailbox mechanism provided by VMS. The system has been designed for each process to have its own mailbox for local communication and to have network mailbox for network communication. As shown in Figure 4, process A can send the signals through VMS to the process B by designating the destination of ch. B mailbox. Reversely process B can respond to the received signals in the same way. However, since the network communication is supported by the DECnet software, at the initial time the required network channel is set up as a server-client model on the demand basis. This implies that if there is no need to have a network channel with the processes in the other unit, the process does not have one. The system also has another type of network communications, i.e. TCP/IP protocol communication for the external interface. This is for a data exchange with UNIX machines, but the implementation of it is limited to only one process in the system.

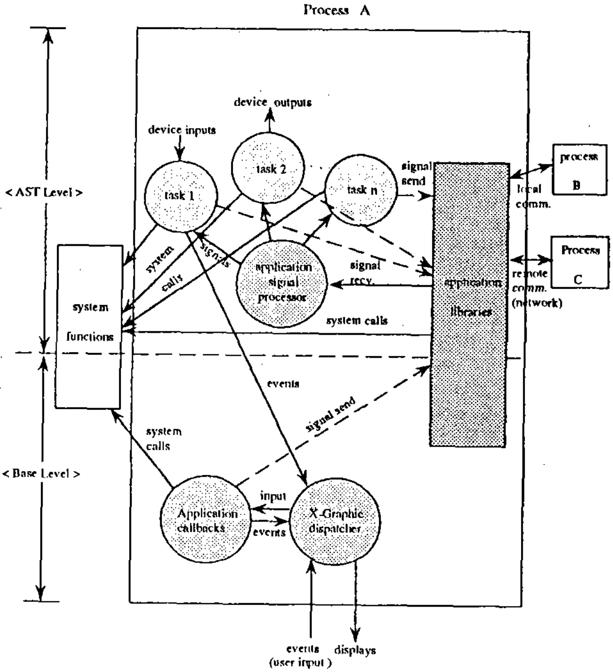


Figure 3 The signal and user input processing structure in a process

III. REAL-TIME PROCESSING SUBSYSTEM

The telemetry reflects the health status, operational configuration, and the dynamics state of the spacecraft. Since the housekeeping of the spacecraft greatly relies upon the operator's monitoring, the importance of the monitoring and a quicker detection of anomalies or failures have been stressed for many years. Moreover since the spacecraft control is activated through ground commands, the commanding environment as well as the anomaly monitoring have been

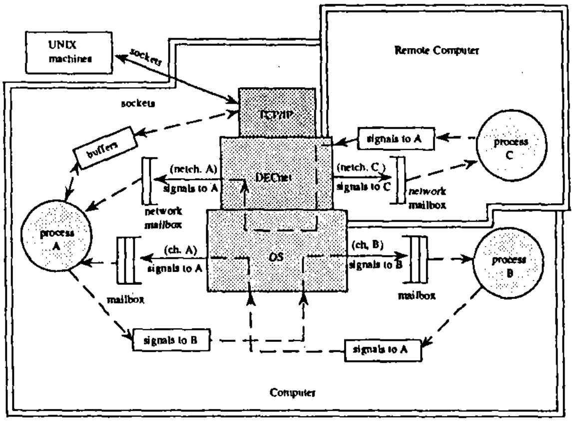


Figure 4 The inter-process communication mechanism

upgraded and evolved. Therefore the effort has been devoted to provide the operator with the graphic telemetry display and commanding information as much as possible, with a centralized and comprehensive monitoring environment for a correct and satisfactory operation.

The system has three large monitors in the main console for operators, providing graphic telemetry display, system/satellite control and alarm summary, and spacecraft mimic diagram display respectively. The system/satellite control monitor is shown in Figure 5. There supported in this facility system control function, satellite commanding, alarm summary display, alphanumeric display, data block handling, and VMS interface. All the telemetry page display layout, display format, display item, alarm value installation, and database update are performed through the system control graphical user interface. An additional X-based page display is also provided to support a centralized monitoring.

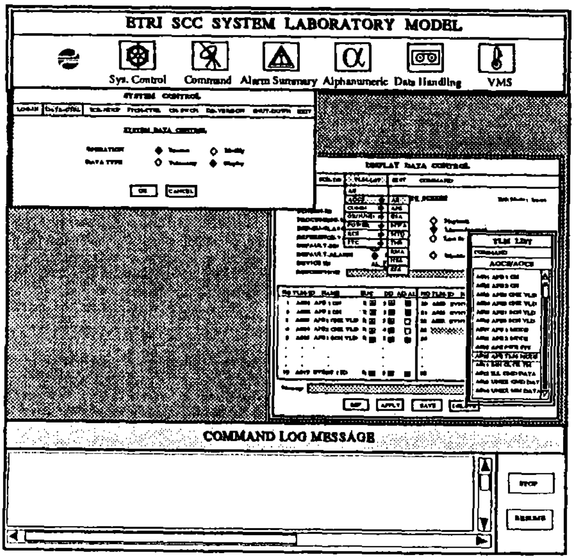


Figure 5 The view of the system/satellite control monitor in the operator workstation

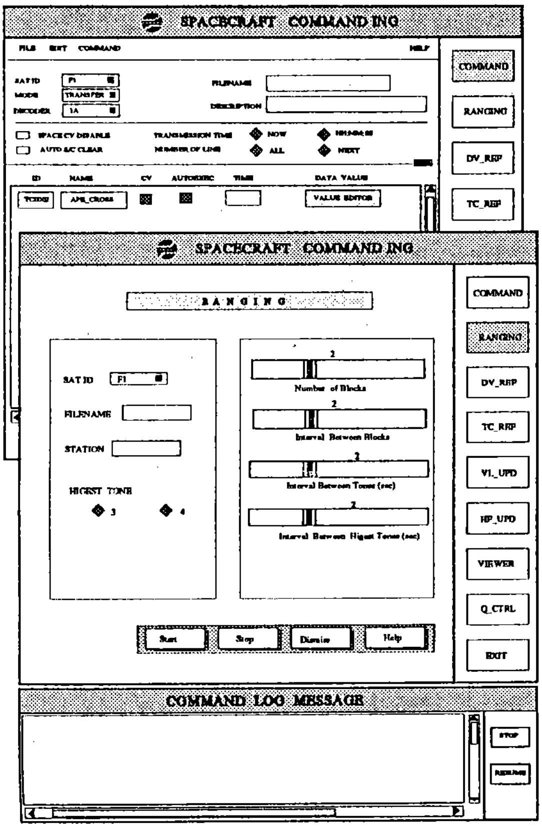
When the telemetry is processed, it is also archived at the same time both in a raw format and in a processed format. The attitude sensor data as well as tank pressure data are also gathered on demand for flight dynamics processing. Short-term and long-term trending of telemetry items as well as playback processing of telemetry from the archives enable the analyst to review the anomalies in detail and to evaluate the performance degradation of the spacecraft. All the commanding activities are supported to be performed on the graphical user interface. Each command is string-format coded so that it shows by itself its full meaning as appeared. Since all commands are characterized and implemented respectively as a selective item and are classified in a subsystem-wise unit, the operator who prepares the command is able to select the appropriate one easily only with a understanding of the control context. Moreover all the input data for commanding are guided to be typed correctly and in an appropriate format. The chance of command input error or misuse of command syntax is kept so narrow. The system also provides the functions for off-line command generation, ranging, transparent command stack handling, time-based transmission control, operator privilege check, and hazardous command authentication. These functions enable the operator to control the satellite with confirmation steps to ensure a sufficient depth of authorization and proper handling. All the activities are logged in the log window and saved into a file that can be redisplayed later for a trace or detailed examination of command history if needed. Flight control command requests and external payload user command requests can be streamed in the command generation facility through a network. The commanding windows samples are appeared in Figure 6.

IV. FLIGHT DYNAMICS PROCESSING SUBSYSTEM

This subsystem takes charge of the determination, the prediction, and the maneuvers of the attitude and the orbit for the satellite. Partly because of a soft time constraint of processing and partly because of a lot of independence among the functions in it, the blocks or processes are somewhat loosely coupled in the structural sense. This means that almost every function is allotted to only one process in contrast to the real-time processing subsystem. A centralized database and data files are shared for each processing and a new run of a process updates the data stack for an access to the latest result. Since the attributes of the functions are in essence analysis and planning, the functions are normally scheduled to be performed beforehand or initiated if required.

There support 12 separate functions in this subsystem and all these are invoked to be ready to process at the start-up time;

o Orbit Determination
o Attitude Determination
o Orbit Prediction
o Sun & Moon Ephemeris Generation
o Event(Eclipses, Sensor Intrusion, Communication Sun Outage)
o Flight Data Pre-processing
o Station Acquisition



SAT ID F1
MODE TRANSFER
DECODER 1A
TCIDNE APE_CROSS

SAT ID F1
Number of Blocks 2
Interval Between Blocks 2
Interval Between Tones (sec) 2
Interval Between Higest Tones (sec) 2

Figure 6 The commanding windows samples

- o Flight Dynamics Data Management
- o Attitude Maneuver
- o Apogee Motor Firing Planning
- o Station Keeping
- o Flight Data Trend Analysis

As for the processes in the real-time processing subsystem, all the processes have their own graphic user interfaces based on MOTIF. Several types of output results are provided; summary reports, data files, and graphs. Figure 7 shows the subsystem processing structure and interactions for the functions.

This subsystem covers both the LEOP(Launch and Early Orbit Phase) mission and on-station mission. For the KOREASAT, in LEOP it is spinning in an inertial frame. Using Shuster's batch method and differential correction method, the initial attitude is determined by right ascension and declination angles based on the telemetered sensor data. Also spin up/down and attitude reorientation/slew maneuvers are planned to be performed by thrusting. Once the command data for maneuvers are generated and reviewed by operators, a request to make a command list is sent to the real-time processing subsystem. The operator at the real-time system is then notified of the arrival of the request, he or she is ready to perpare the commanding. In this phase orbit determination and prediction is done by gathering the tracking data coming from the real-time processing subsystem. Of course ephemeris generation and raw tracking data preprocessing, for example wild point editing or smoothing, are executed by the operator before the orbit determination and ground track/visibility calculation. The apogee motor firing is also planned by iterating the simulations for candidates to find an optimal

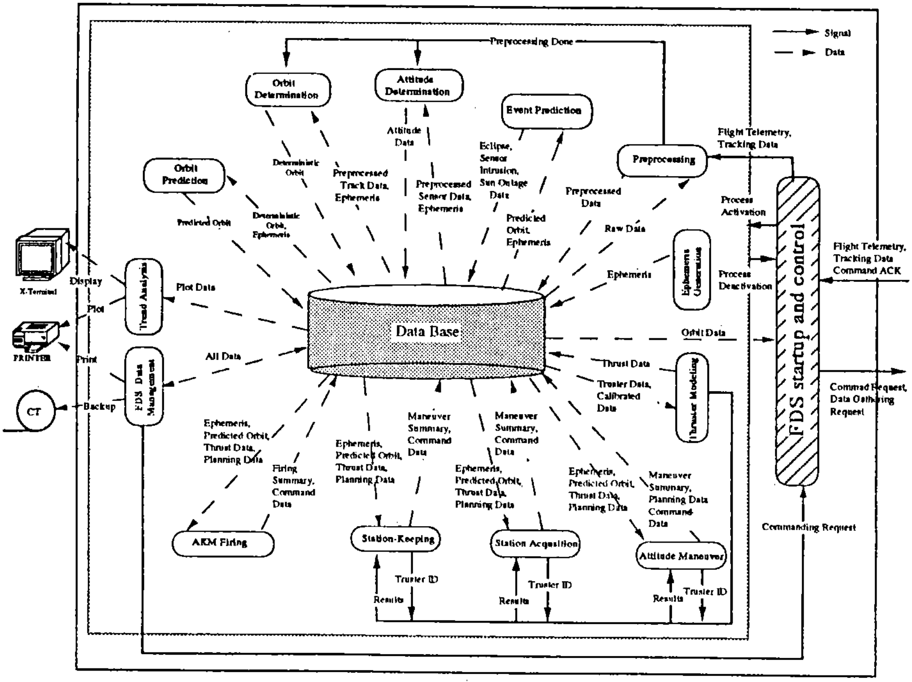


Figure 7 The processing structures and internal interactions in the FDS subsystem

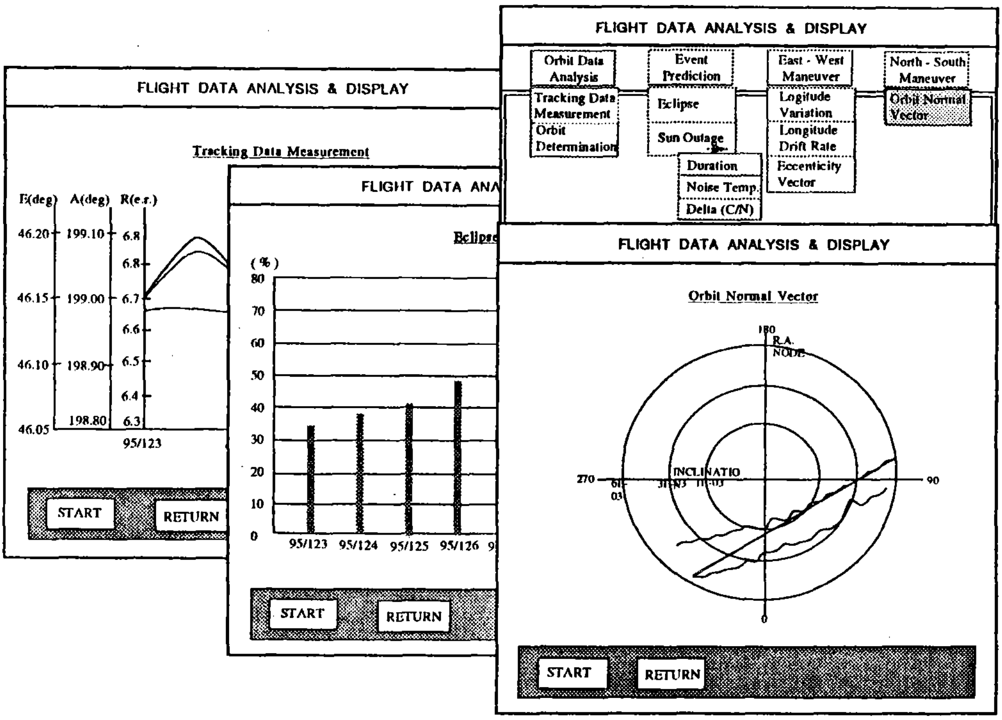


Figure 8 The flight dynamics data trend analysis windows

solution, which can give an efficient station acquisition for the satellite from the drift orbit to final orbital position. After the satellite finally gets its nominal attitude, orbit, and position, all the deployments are done and then it is in the situation of primary missioning commissioned. Station-keeping as well as orbit determination, prediction, and event prediction are normally and periodically performed. However, for some on-board functions like three axis attitude control, there is no specific backup function, e.g. ground loop control, as is often the case with the commercial s/w packages. The various outputs as well as historical data of flight dynamics are used for graphical display for trend analysis and subsequent planning. Some examples are appeared in Figure 8.

V. CONCLUSIONS

A functional model of satellite control software system has been for the first time domestically developed targeting for the first generation KOREASAT. All the activities and processes involved in the development including system design have been performed by ETRI engineers. The internal system function test was successful in conjunction with the system integration. The performance and test results of the flight dynamics processing were evaluated and proved to be satisfactory in comparison with those of commercial software. With the connection of the satellite software simulator, which is also developed in ETRI, the overall system test is scheduled to start in September this year. This system can be tested in a real operation configuration and thereafter used as an analysis workbench for the KOREASAT provided that the real telemetry data stream received from the KOREASAT TT&C backup station in Taejon is plugged into. Also this can serve as a domestic prototype system model for a practical satellite ground control.

ACKNOWLEDGMENT

This project is funded by the Korea Telecom(KT) and associated with the KOREASAT -related R&D program.

REFERENCES

[1] P. Vielcanet, et. al., "MERCATOR: A Distributed Ground Control System For Geostationary Satellite Positioning", *Proc. ESA Symp.:'Ground data systems for spacecraft control'*, June 1990, pp 81-86

[2] R. Thompson, "Man-Machine Interface(MMI) Aspects of Workstations For Use in Spacecraft Operations", *Proc. ESA Symp.:'Ground data systems for spacecraft control'*, June 1990, pp 577-582

[3] J. Garner, et. al., *Satellite Operations: Systems Approach to Design and Control, Series in Space Technology, Ellis Horwood*, 1990, pp 99-106