

Telemetry Data Mining with SVM for Satellite Monitoring

Yosuke Fukushima

*Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency,
Japan*

1. Introduction

The aim of this chapter is to provide readers with a basic knowledge of satellite monitoring and data mining for anomaly detection using a support vector machine (SVM) technique. The author describes the design and implementation of an SVM and an example of the use of the satellite hardware anomaly detection to discover instability in the attitude rate bias of a gyro sensor. This anomaly is caused by a change in the characteristic parameter of the gyro hardware: a statistical parameter related to noise specifications. The detection is demonstrated using telemetry data that have been sent by an actual science satellite.

This chapter is divided into three sections: first, the author describes the target satellite and the basic mathematical modelling and formulation of attitude dynamics of the satellite. In the formulation, kinematics and model parameter estimation technique using Kalman filter method is described to provide readers the key parameter; the drift parameters of attitude rate gyro, which are to be dealt with in the following sections in detail. Estimation of unknown parameter of the formulation is also shown using actual telemetry data. This scheme called observers is most popular method for almost every satellite. Second, a brief introduction of the SVM technique is given and followed by a design and implementation of the SVM technique to the gyro bias instability detection. This analysis and calculation are performed using a set of real telemetry data are given. Finally, a software architecture is proposed that will make it easier to migrate SVM software into an onboard computer as a step toward realizing onboard autonomy.

Although the formulation developed in this chapter is concerned with attitude rate sensors of a particular satellite, this approach can be applied to other types of remote systems; a remote system that is designed to be operated by human operators in a distant site by communicating using telemetry systems. This type of onboard autonomous system monitoring seems to be promising not only in all remote systems that are working at server circumstances such as space or deep underwater but also in some consumer products such as cars and trains.

2. Onboard satellite health monitoring

In this section, knowledge of attitude determination of a satellite is given by modelling and analysis of an actual satellite attitude motion in detail. It is necessary to understand meaning

and mechanism of estimation of characteristic parameters of hardware as health indicators of satellite systems throughout operations. To dealing with change in value of such parameters, a complete set of analytical process of attitude determination is shown. The telemetry data used in this chapter was obtained in an actual satellite operation.

2.1 Aurora science small satellite REIMEI

The Institute of Space and Astronautical Science, Japan Exploration Agency (ISAS/JAXA) has launched a series of scientific satellites including planetary spacecraft as well as astronomical observation satellites. Although the missions have achieved fruitful scientific results, these satellites, including our own M-V launch vehicle, have cost nearly 160 million dollars and taken over eight years to develop. As a result, the launch frequency of scientific satellites has decreased significantly in the last decade. In addition to these large and expensive missions with a long development time, an inexpensive mission with a short development time involving a small piggyback satellite has been planned. This satellite should be an effective tool for demonstrating new technology and performing scientific observations.

A small satellite named "REIMEI" was developed from 2000 and was launched to a height of 610 Km by the Dnepr Launching Vehicle from Baikonur Cosmodrome launch site in Kazakhstan on August 24, 2005 (Saito et al., 2005). Since then it has followed a near Sun synchronous orbit. The Japanese word "REIMEI" means dawn, and the satellite name was chosen to celebrate the new era of high-performance small satellites developed in Japan. REIMEI's mission objective is to observe dynamic aurora phenomena using the three spectrum imagers (MAC) and two particle analysers (E/ISA) installed onboard. Observations are carried out with the aim of studying the small-scale dynamics of terrestrial aurora, namely, their spatial distributions and time variations, and their correspondence to the spectral properties and spatial distributions of charged particles. Figure 1 shows the flight configuration of REIMEI. There are two solar-concentrated deployable paddles on the top surface that can generate a power of 150 W, three camera lens holes for MAC in the black-kapton-covered front surface, and E/ISA are installed covered by the silver-Teflon-lined right surface.



Fig. 1. Flight configuration of REIMEI

2.2 Attitude control requirements

REIMEI has a bias-momentum three-axis attitude control subsystem (ACS) to meet the requirements: the attitude control accuracy should be less than 0.5 degree, and the attitude determination accuracy should be less than 0.05 degree. These requirements are specified for one of the most important observation modes, the image/particle simultaneous observation mode, in which E/ISA detect particles to count their number as well as to measure their energy, while MAC captures aurora phenomena emerging at a magnetic line foot-point, where the aurora is energized by the particles observed.

ACS inputs data from sensors such as the spin type sun sensor (SSAS), the two-dimensional sun sensor (NSAS), the star tracker (STT), the three-axis geomagnetic field aspect sensor (GAS), and three-axis fibre-optic gyroscopes (FOGs). On the other hand, ACS outputs data to actuators such as the small momentum wheel (WHL), which provides the satellite with bias momentum (0.5 Nms), and the three-axis magnetic torquers (MTQ). Figure 2 shows the flight models of FOG of REIMEI.

ACS can be divided into two function blocks: the attitude control and the attitude determination blocks. Since several papers have been published (Sakai et al., 2006a; Fukushima et al., 2006) dealing with the REIMEI attitude control block that refer to its algorithm, formulation, and flight results, here we will focus on the attitude determination block illustrated in Fig.3. The inputs for the attitude determination block are the angular increment angle data measured by FOG and the five star vectors measured by STT. The outputs are the attitude and the attitude rate of REIMEI, which are expressed in an inertial coordinate system. Attitude is expressed using quaternions in this paper and is propagated by the integration of an initial or current attitude quaternion using the attitude rate with respect to time. The attitude rate is also called the angular velocity. Note that the attitude rate is calculated from both FOG outputs and the FOG bias. The FOG bias is residual output when FOG is motionless in the inertial coordinate system.

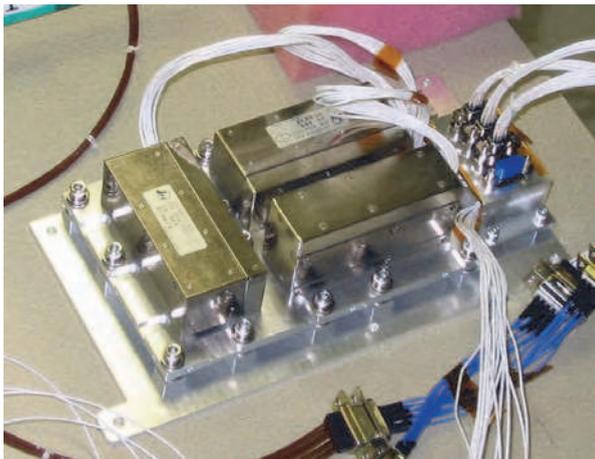


Fig. 2. Flight models of three FOGs assembled on one aluminium plate.

An error correction procedure must be installed into the attitude determination block since there will be a modeling error or observation error in the a priori initial attitude, the attitude

rate observed by FOG, and the star vectors obtained by STT. For example, if the FOG bias is not suitably taken into account in the attitude propagation, the attitude rate error will accumulate to a value that will corrupt the attitude propagation. In addition, the FOG bias fluctuates with respect to time and temperature of FOG. Thus, REIMEI has a Kalman filter programmed in the attitude determination block to simultaneously estimate the attitude estimation error and the FOG bias estimation error.

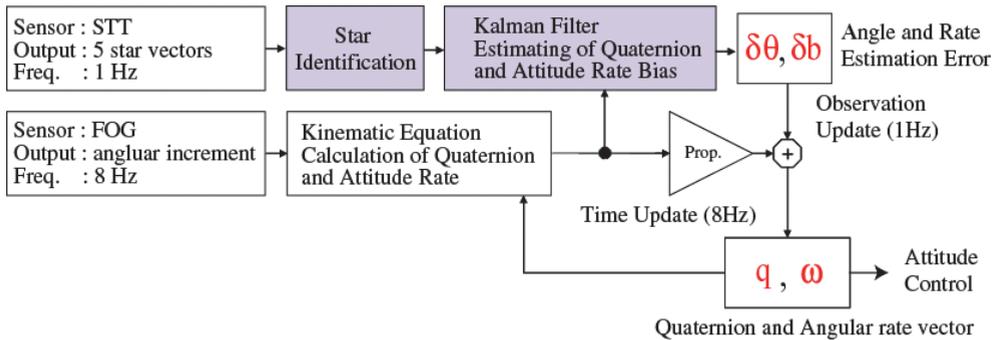


Fig. 3. Diagram of attitude control of REIMEI

2.3 Attitude kinematics and attitude determination

An extended Kalman filter (EKF) theory was employed to determine the attitude and attitude rate simultaneously. This filter algorithm is based on an attitude uncertainty model in rotating body coordinates that is discretized with intervals equal to the attitude update time interval. The algorithm estimates both the attitude estimation error angle theta and the attitude rate bias b. Bias b is also called FOG bias.

The satellite rotation equation and the relation between the attitude rate and the FOG bias are given by

$$\dot{q} = \frac{1}{2} \Omega(\omega)q \tag{1}$$

$$\omega = \omega_{FOG} - b \tag{2}$$

$$\dot{b} = 0, \tag{3}$$

where q is a quaternion vector, ω is an attitude rate vector, b is an rate bias vector, and ω_{FOG} is an attitude rate vector measured by FOG. $\Omega(\omega)$ is the following well-known 4x4 skew matrix

$$\Omega(\omega) = \begin{pmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{pmatrix} \tag{4}$$

If q is considered as reference, the state equations can be linearized around q . In other words, q can be regarded as an ideal determined attitude under the condition that there is no noise or uncertainties when Eqn. 1 is integrated with respect to time.

2.4 Estimation variables

First, it is necessary to define estimation variables and formulate state equations expressed in terms of these variables. In this paper, the attitude estimation error vector and FOG bias estimation error vector are used for this purpose. The attitude estimation error vector $\delta\theta = [\delta\theta_1 \ \delta\theta_2 \ \delta\theta_3]$ represents the difference between the estimated attitude and the actual attitude, while θ represents small rotation angles with respect to the X, Y, and Z axes. The FOG bias estimation error vector $\delta b = [\delta b_1 \ \delta b_2 \ \delta b_3]^T$ represents the difference between the estimated bias and the actual bias with respect to the X, Y, and Z axes. According to Farrenkopf's paper (Farrenkopf, 1978), the following equations hold.

$$\delta\dot{\theta} = \omega - \dot{\theta} - (b + n_v) \quad (5)$$

$$\delta\dot{b} = n_u \quad (6)$$

where n_v and n_u have the following statistics characteristics.

$$E[n_v(t)n_v(\tau)] = \sigma_v^2 \delta(t - \tau) \quad (7)$$

$$E[n_u(t)n_u(\tau)] = \sigma_u^2 \delta(t - \tau) \quad (8)$$

The constants σ_v and σ_u indicate the level of FOG bias drift and the random walk characteristics, respectively. They are usually estimated through experiments or given by the manufacturers.

Since we did not measure the precise alignment between FOG and STT, there is probably some misalignment that may cause the attitude rate of one axis to have a non zero projection onto the other axes. In addition, there are unobservable parameters of FOG in principle; thus, b should be regarded as the net equivalent bias vector.

2.5 Formulation of attitude determination

Figure 4 shows the update sequence in EKF. There are two types of update: the time update with an 8 Hz cycle and the observation update with a 1 Hz cycle. Since the state variables are updated in these cycles, their accuracy will vary during estimation. In the time update steps, the attitude is propagated by the integration of q using the small Euler angle approximation for the three axes $\Delta\phi = [\Delta\phi_1 \ \Delta\phi_2 \ \Delta\phi_3]$. $\Delta\phi$ is the incremental angle data including both the bias and the misalignment uncertainty. ω_{FOG} is obtained by dividing $\Delta\phi$ by the propagation time interval. In the observation update steps, the estimation is performed by subtracting $\delta\theta$ and δb from the calculated θ and b is obtained from the latest observation update using Eqns. 2 and 3. After that, θ is recalculated by integrating the sum of the $\Delta\phi$ sampled during the STT update time interval.

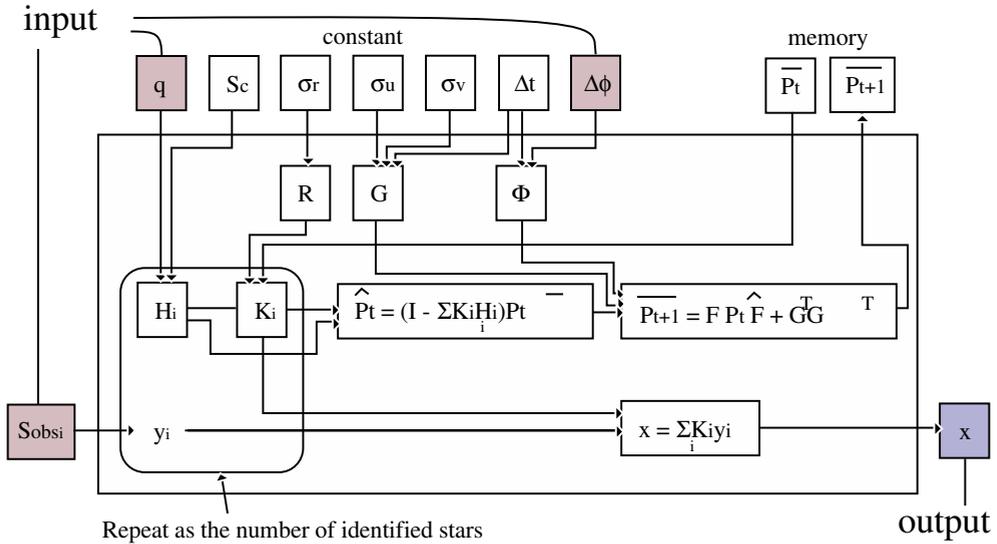


Fig. 4. EKF block diagram showing data-flow

Observations are made by obtaining a residual vector y expressed as

$$y = S_{obs} - (\delta\theta C S_c + n_r) \tag{9}$$

$$= S_{obs} + (C S_c) \delta\theta - n_r \tag{10}$$

$$\delta\theta = \begin{bmatrix} 0 & \delta\theta_3 & -\delta\theta_2 \\ -\delta\theta_3 & 0 & \delta\theta_1 \\ \delta\theta_2 & -\delta\theta_1 & 0 \end{bmatrix} \tag{11}$$

where “ \sim ” indicates the tilde operator used to form a 3x3 skew matrix from a 3x1 vector, S_{obs} is the observed star vector, S_c is the corresponding catalog star vector, and C is the direction cosine matrix of the satellite composed using q . The observation noise vector n_r can be expressed as follows using the delta function:

$$E[n_r(t)n_r(\tau)] = \sigma_r^2 \delta(t - \tau) \tag{12}$$

Figure 11 shows a block diagram of EKF (Farrenkopf, 1978). The symbols G , R , and K are described as follows. x is a state vector composed from $\delta\theta$ and δb . The state transition matrix of x can be written as

$$x(t + \Delta t) = \Phi x(t) = \begin{bmatrix} \Delta\phi & -\Delta t I \\ 0 & I \end{bmatrix} \begin{bmatrix} \delta\theta(t) \\ \delta b(t) \end{bmatrix} \tag{13}$$

Since we apply the reset-type Kalman filter (Ninomiya et al, 1994), x is the zero vector; in other words, x will be reset to zero after every observation update.

G is the process error matrix written as follows:

$$G = \begin{bmatrix} \left(\sigma_v^2 + \frac{\sigma_u^2 \Delta t^2}{3} \right) \Delta t I & -\frac{\sigma_u^2 \Delta t^2}{2} I \\ -\frac{\sigma_u^2 \Delta t^2}{2} I & \sigma_u^2 \Delta t I \end{bmatrix} \quad (14)$$

H is the observation matrix written as follows:

$$H = \left[(CS_c) \quad 0_{2 \times 2} \right] \quad (15)$$

R is the following observation noise matrix:

$$R = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_r \end{bmatrix} \quad (16)$$

K is the following Kalman gain matrix:

$$K_i = \bar{P}(t) H_i^T \left(H_i \bar{P}(t) H_i^T + R \right)^{-1} \quad (17)$$

where suffix i is the sequence number of the identified stars.

The observation update shown in Fig.4 is performed up to i so that the inverse of a matrix larger than 3×3 does not appear in Eqn.17.

We have now defined all the matrices required for EKF. Even if the EKF process must be turned off for some reason, the time update process should be continued using Eqns.2 and 3 every 125 ms.

2.6 Flight results of attitude determination

The attitude determination system of REIMEI mainly depends on the STT, which has 3 arcmin accuracy. If the STT is not available, the FOGs take the role of the principal sensor to acquire the current satellite attitude by propagation. Two periods in which the STT is available and not available follow each other cyclically as illustrated in Fig. 5. In science operations, the earth enters the field of view (FOV) of the STT when the satellite attitude is controlled at a fixed value and pointing in a particular direction with respect to an inertial coordinate system. The durations of the periods when the STT is unavailable (SIT-OFF period) and available (SIT-ON period) are 67 min and 30 min, respectively.

This EKF has been operating as expected for more than five years (from Sep. 2005 to Mar. 2011) and no serious failures have occurred. The observed stars were scattered inside SIT-FOV, in other words, they were not gathered within a small area of SIT-FOV, resulting in attitude error correction being performed efficiently. Figure 6 shows a sample of EKF telemetry data including the determined quaternion vector q , the observed error angle vector $\delta\theta$, and the attitude rate bias vector b . REIMEI performed Z-axis maneuvers at 1:30 and 2:40 on Aug 5, 2006. There are four $\delta\theta$ data for the identified stars viewed by SIT. Note

that the $\delta\theta$ in Fig. 6 were calculated from y shown in Fig. 4. The observed error angle includes the effects of n_r .

The resultant accuracy derived from obtained telemetry are attitude determination is $0.04 \text{ deg} \pm 0.003(\sigma)$ and accuracy of rate bias is $0.1 \text{ deg/h} \pm 0.08(\sigma)$, respectively. The accuracy of b can be evaluated using the first several $\delta\theta$ data obtained shortly after beginning the STT-ON period. $\delta\theta$ for the first several data are equivalent to the angle given by the sum of the STT output errors, which can be modelled by a Gaussian, and the cumulative error angle of $\delta b \Delta t$. In other words, if δb is sufficiently larger than the STT noise, then we can regard the first $\delta\theta$ as $\delta b \Delta t$. Note that the accuracy may vary depending on the operation maneuverer plan, for example, how long STT has been turned off due to observation attitude requirements, or how rapidly the attitude is changing. However, the accuracy of estimation is typical for the most frequent observation operations using the telemetry data.

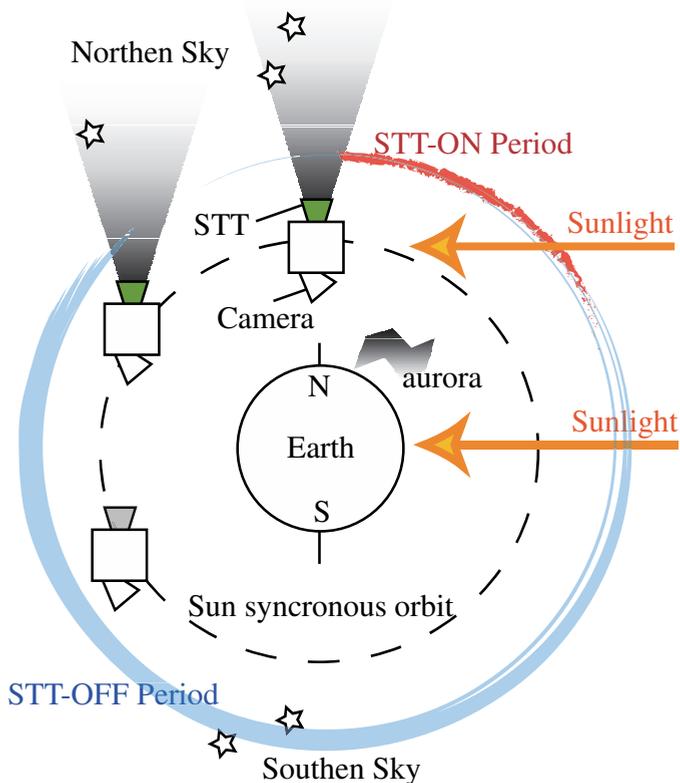


Fig. 5. STT-ON and STT-OFF operation: since the attitude of the satellite is fixed in an inertial coordinate system during the observation periods, STT must be turn on and off cyclically to prevent the Earth's albedo (reflection of the sunlight) from coming inside STT-FOV. This is one of a specific limitation of the use of STT in REIMEI system.

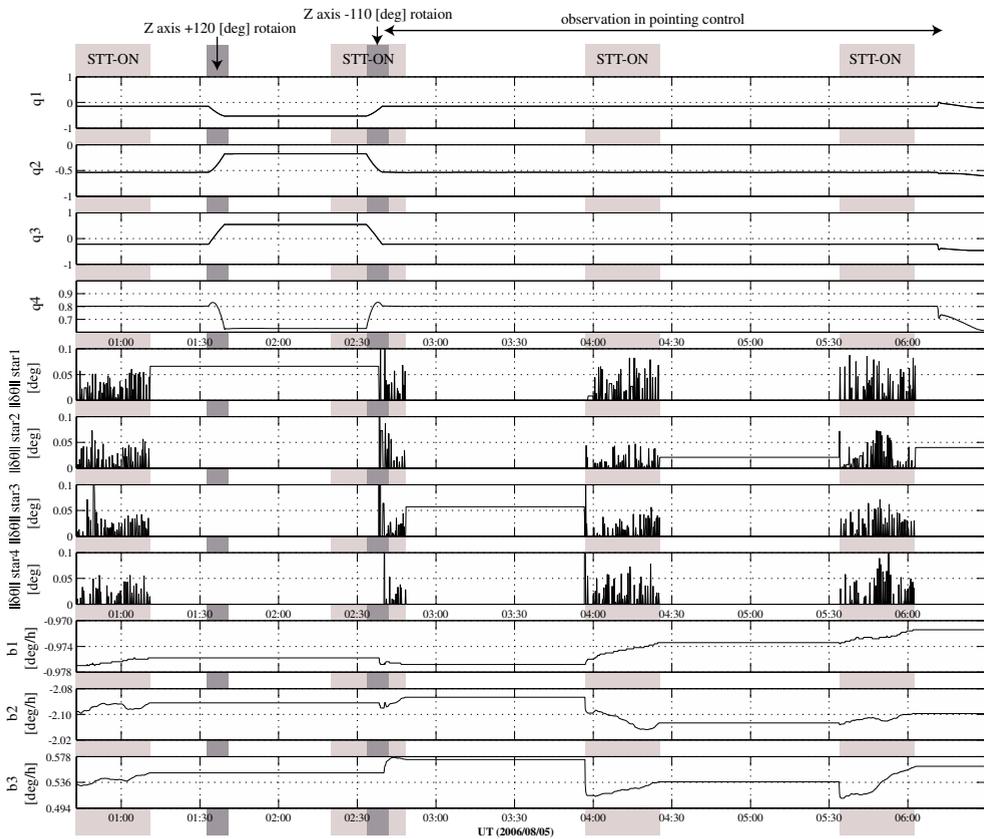


Fig. 6. Attitude and bias estimation results from actual telemetry data

3. FOG Bias instability problem

The accuracy of satellite attitude estimation depends only on the FOG data during the 30 min of STT-OFF periods; thus, the bias stability of the FOGs is a crucial factor in maintaining the accuracy. With several STTs onboard, such limitation to the accuracy of attitude would not exist. Owing to their small weight and volume, small satellites have little capacity for onboard components, and it is not unusual for a small satellite to have only one sensor or one actuator onboard, whereas most satellites have several sets onboard.

Although the attitude is estimated sufficiently accurately to meet the requirements of the mission during the STT-ON periods, this is not the case during the STT-OFF periods. The cumulative attitude error caused by the bias estimation error increases to a value exceeding the requirements.

There are two sources of bias estimation error: the Kalman filter tuning performance and the bias instability. In REIMEI system, the bias is modelled using Farrenkopf's gyro dynamic model shown in the section 2.4.

3.1 Accuracy of bias drift estimation

Since the parameters σ_v and σ_u were previously carefully tuned using a flight software simulator and actual flight telemetry data, the results of attitude estimation and FOG bias estimation for the REIMEI satellite were sufficiently good to ensure operation without any problems until about 18 months after the launch.

To maintain the estimation performance, it is necessary to continually monitor the temperature of the FOGs (Sakai et al., 2006b). Changes in temperature have a strong effect on the stability of the FOGs. Figure 7 shows an example of changes in bias observed in the telemetry data (Fukushima et al., 2006). Although these FOGs were well calibrated and their thermal environment has been controlled by heaters, some minor problems have arisen in bias estimation since Nov. 2007.

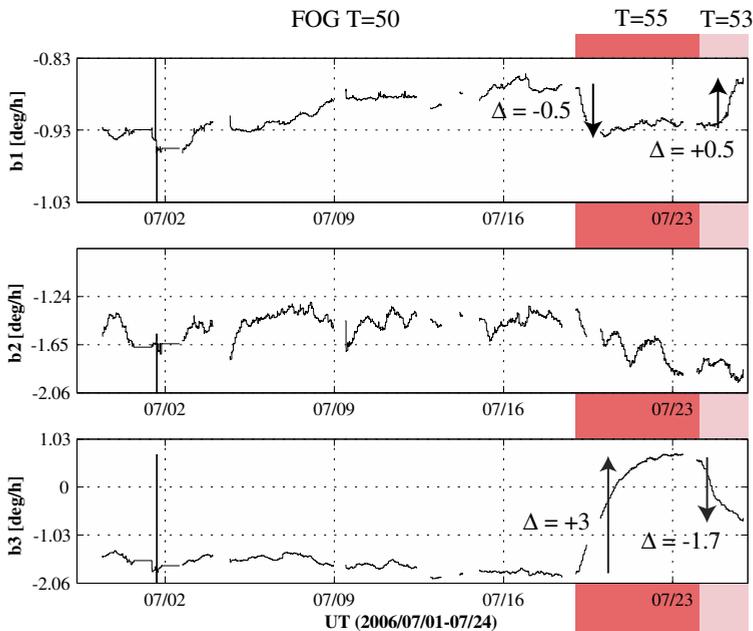


Fig. 7. FOG bias drift caused by change in FOG temperature

The accuracy of the estimated bias can be evaluated using the attitude error. This error can be observed at the moment when the STT is switched on. The attitude error is defined as the angular difference between the observed star vectors and the corresponding theoretical star vectors calculated using the onboard star catalog and the satellite attitude. This angular difference of propagation can be regarded as being equal to the cumulative error caused by the bias estimation error during the STT-OFF periods. Figure 8 is a schematic drawing of the cumulative propagation error, with its standard deviation σ_θ formulated using σ_v , σ_u , and the deviation of the STT observation data σ_n .

$$\sigma_\theta^2 = \sigma_n^2 + \sigma_v^2 T + \frac{1}{3} \sigma_u^2 T^2 \tag{18}$$

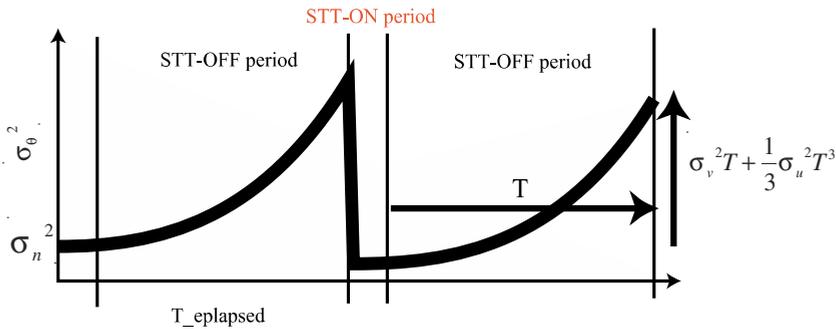


Fig. 8. Cumulative propagation error caused by bias drift of FOGs

3.2 Bias instability observed in telemetry data

The angular difference between the star vectors observed using the STT and the corresponding theoretical star vectors should be monitored during STT-ON periods to verify the accuracy of the STT data. If there is a problem with the STT, this angular difference will increase.

Figure 9 shows several examples of time history plots of the angular difference on the same axes. The horizontal axis is the time elapsed from the moment when the STT is switched on, while the vertical axis is the angular difference. We can observe two types of angular difference data on this plot. The data points plotted on or near the vertical axis (t=0) are the angular difference of propagation, and the data points near the center of this plot, with times of about 480 s to 1200 s, show the angular difference of the STT.

Figures 10 and 11 show time history plots of these two types of angular difference obtained over the last two years (2007 and 2008). From these plots, we can conclude that (1) the STT does not appear to be developing any signs of problems and (2) the bias estimation error has increased discontinuously several times, i.e., some unknown factors have affected the stability of the characteristic parameters, σ_v and σ_u , of the FOGs.

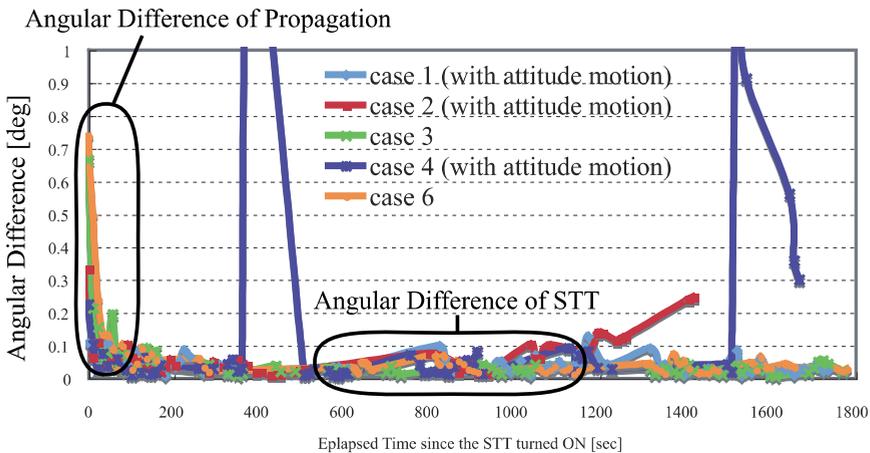


Fig. 9. Angular difference plots overlaid aligned for each start from STT-On event to the left

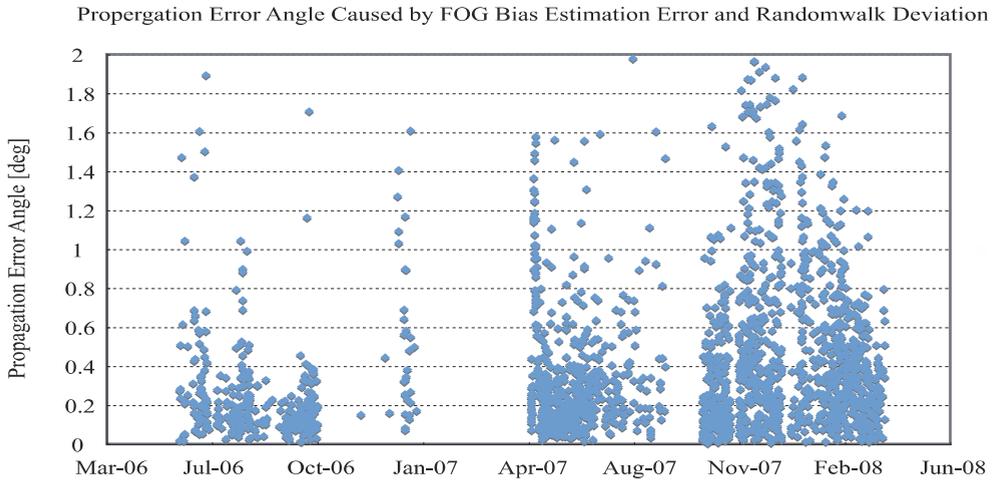


Fig. 10. Trend of angular difference of propagation

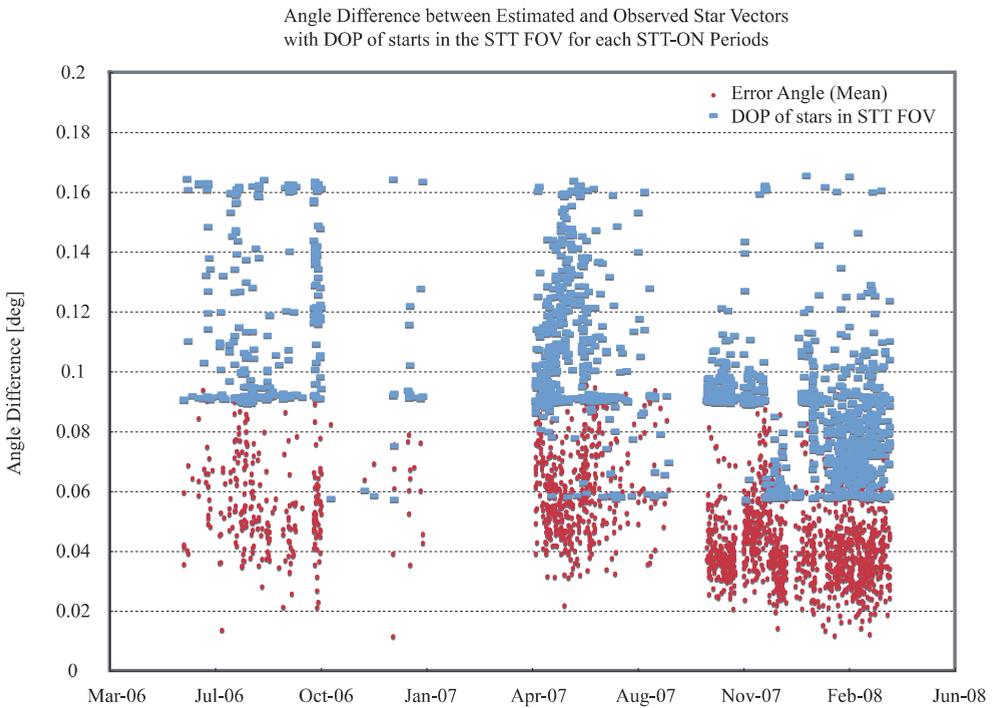


Fig. 11. Trend of angular difference of STT

We can see that the angular difference of the STT appeared to be larger during some periods in Fig. 11. This difference can be explained using the dilution of precision (DOP) of stars. The DOP of stars is defined in this paper as an index of how stars are aligned in the FOV of the STT; its formula is given as follows:

$$DOP_{stars} = \sqrt{\text{trace}(H^T H)^{-1}} \quad (19)$$

where H is the observation matrix used in the Kalman filter of this system. The larger, the value of DOP, the less accurate the attitude estimation, similarly to the DOP used in the accuracy analysis of the Global Positioning System. The DOP indicates how stars are seen in the FOV of the STT, which affects the accuracy of attitude determination. The degradation of accuracy of attitude estimation over time is inevitable for such a system with only one STT.

4. Detection of changes in FOG bias using SVM

It has been needed to find any weak sign of event that would link to some system troubles. Such demand called failure detection, a sub category of system monitoring, has been researched for ground systems as well as onboard systems in many fields. Having experience of working for years as an operator of REIMEL, the author also has felt such for failure detection only for system failure but also for detection of degrade of performance: signs of changes in specification that there is no error but something seems to be wrong. In this section, an example of such failure detection is shown using a problem of degrade of FOG bias drifting in REIMEL.

4.1 Signs of changes in bias

Figure 12 shows time history plots of the angular difference of propagation and the angular difference of the STT over 2 years, which are plotted over different time spans such as one month, ten days, and one day.

From the statistics for these time spans, it is easy to observe the level of fitness of the EKF parameters. By taking statistics over a longer time span, randomness can be reduced. For example, there is no sign of parameter mismatch before Oct. 2006, but an event occurred in Oct. 2006; the difference between the median and mode of the distribution of the angular difference of propagation has been increasing since Nov. 2006. The median indicates the accuracy of bias estimation, while the mode indicates the random-walk error of bias drift.

According to these time history plots over 2 years, it can be concluded that the EKF parameters have been mismatched with the actual state of the FOGs since Oct. 2006. This means that the characteristics of the FOG bias changed at that time. The most probable reason for the bias error is that the degradation of fibre transparency due to radiation became sufficiently large to cause bias instability. However, this degradation is still considered to have a limited effect according to the results of radiation tests performed before the launch; there must be another factor causing this degradation.

From the operator's viewpoint, one month is a too long period to take any measures to prevent the degradation of accuracy of attitude determination. Any parameter mismatch must be noticed as soon as possible rather from the analysis of monthly statistics.

4.2 Support vector machine

The easiest way to rapidly detect signs of bias change is to shorten the time span of statistics to ten days or one day. However, the shorter the time span, the less accurate the statistical results. Thus, it is necessary to define a threshold for determining whether or not the level of fitness of parameters is acceptable before calculating short-time-span statistics using the telemetry data.

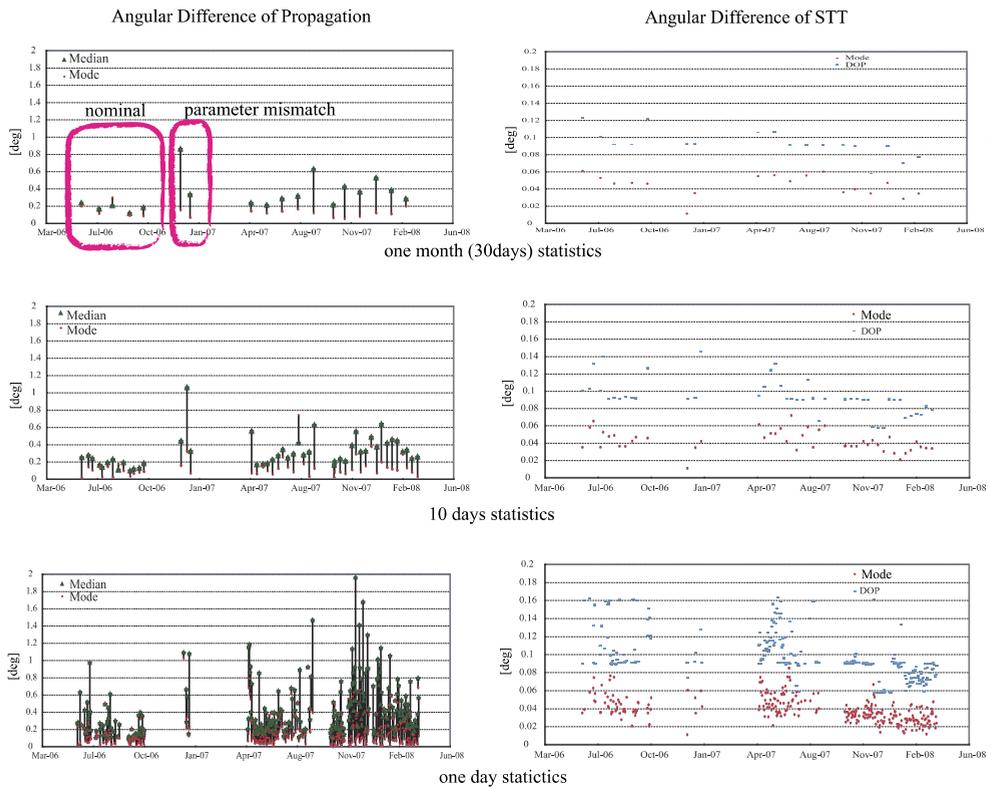


Fig. 12. Trend plots of the angular difference for two years time span

The support vector machine (SVM) technique is one of the most practical methods of detecting such signs from telemetry data. It is a popular discrimination scheme used in many fields. The SVM technique is a type of supervisory learning method (Cristianini & Shawer-Taylor, 2000). It is thus necessary to carry out a training process before using it for discrimination, which is shown in Fig. 13. The SVM technique is as follows: in Step 1, a telemetry data set that was obtained before the change in bias is prepared. This data set is treated as a no-error (OK) data set. In Step 2, a telemetry data set is prepared that contains some signs indicating that the parameters are unsuitable. This data set is treated as an erroneous (NG) data set. In Step 3, using these data sets as training data sets, support vectors are calculated. The kernel function is a simple linear function for this case. Using the

support vectors, the entire telemetry data set (test data set) can be classified as an OK data set or an NG data set, i.e., it can be established whether or not the current parameters are tuned appropriately.

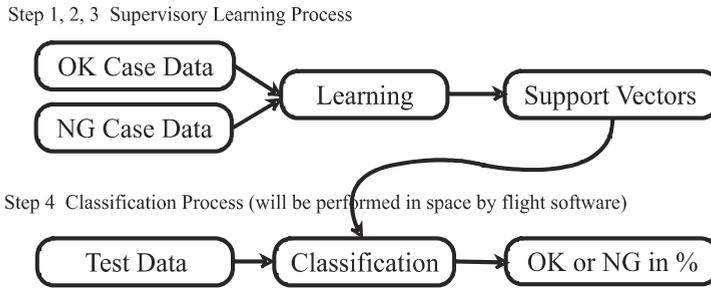


Fig. 13. The 4 steps to make up a SVM

The angular difference of propagation and the DOP of stars shown in Fig. 14 were selected as the training data set to test the performance of the SVM technique. Even though the SVM was generated using one month of statistical data, it detected a change in bias for ten days of statistical data as well as one month of statistical data. A summary of this detection is shown in Fig. 15. Since Mar. 2007, the parameters have been changed several times to reduce the angular difference of propagation to a sufficiently small value to meet the mission requirements. All these changes were detected in the results.

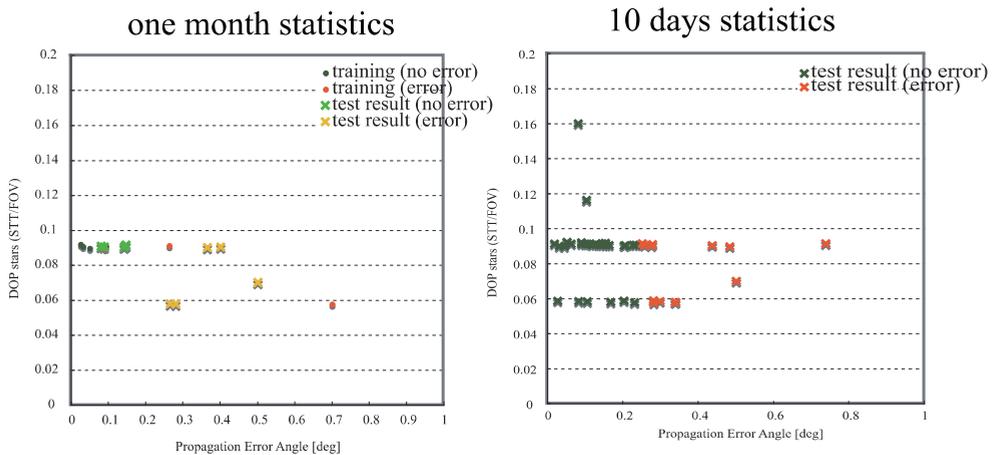


Fig. 14. Two plots of propagation error angle vs DOP stars calculated using deferent statistics. These plots show the discrimination performance how the SVM generated using training data can judge the state of FOG bias using test data.

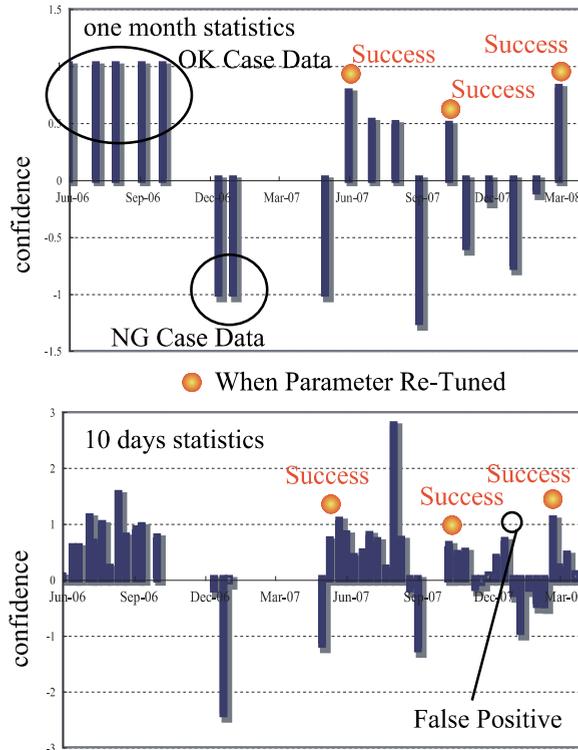


Fig. 15. Classification results by the SVM generated using OK and NG data sets. The result using one month statistics case successfully detects all the changes in FOG bias drifts, while the 10 days statistics case have one false positive detection (miss detection of change in FOG bias).

5. Onboard software architecture

If onboard system monitor is implemented on flight software, it is necessary to consider what software architecture should be employed. Most hardware control tasks must run in a fixed time interval or by triggered as a response to an event while satisfying strict timing requirements. In other words, such a control task must run as a real-time task. On the other hand, system monitoring shown in previous sections does not have such limitations. Since timing requirements for hardware control and system monitoring are different, corresponding software should be implemented in different level of tasks.

To implement flight software clearly structured as multiple tasks running in parallel without unexpected mutual interventions, layered programming approach is suitable. Although there are many researches on this topic, the three-layer structure programming is considered in this chapter as basic software architecture (Rabideau et al., 1999). because such structure is simple, popular, and has a variety of applications even in space systems (Balzano & Isaacs, 2006). The architecture consists of a deliberator layer, an executor layer, and a driver layer as shown in Fig. 16. The driver layer manages data interfaces with hardware in a real-time manner by performing tasks, such as serial data port handling and interruption signal handling. This layer should be implemented to perform real-time tasks

on a real-time OS. The deliberator layer processes all the input data such as environment measurement data, hardware status data, and operation commands and determines the next action to be taken. The executor layer connects input and output data between the executor and driver layers by interpreting the stream of abstract data such as operator commands (for examples GO or STOP) into the hardware data stream (ON or OFF) and vice versa. It also manages the issue of commands, i.e., when and where to send the data.

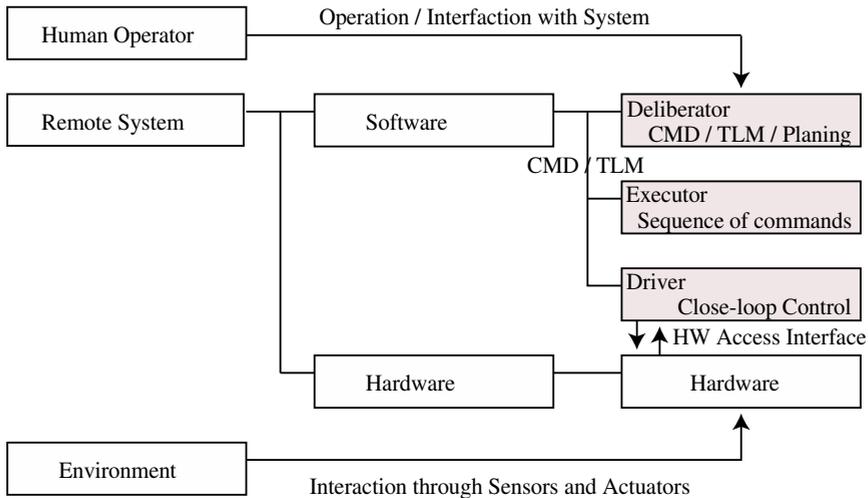


Fig. 16. The three-layer architecture for remote system software consisting of Deliberator, Executor, and Driver functions.

A human operator at a remote station communicates with a remote system by telecommands (CMD) and telemetry (TLM). The operator reads TLM data to determine the event that has occurred at the remote system and then sends CMDs so that the system can operate as planned to achieve its mission goals, resulting in a human-in-the-loop control system. To make the remote system autonomous, the deliberator layer should behave in the same way as the human operator. If this can be achieved, the lower layers will act in the same way under both a human operator and autonomous control. If this situation can be realized, the system can be considered to have the ability to make its own decisions.

Here is an example to show how to implement such architecture in Fig. 17. The driver and executor layers is implemented as real-time tasks running on a real-time OS and the deliberator is implemented as a script engine running on the same OS so that it is easy to communicate between the three layers by an intertask communication mechanism provided by the OS kernel. The reason why a script engine is used on the deliberator layer is that it is expected that algorithms for system monitoring such as SVM should change its target according to operation purposes and circumstances of the system: many algorithms should be loaded to detect many potential failures throughout its mission. FOG bias drift detection was explained in this chapter, but it is not only possible trouble on onboard hardware. Using usual tasks programmed in C/C++, it is hard to change onboard algorithm for system monitoring after starting up the system. However, script engines seem to be an only option to change algorithm frequently. By changing the script, the algorithms can be changed without affecting other running tasks such as hardware controls.

The requirements of time and memory for a script engine to run with multiple real-time tasks have been too great for its realization on an actual onboard computer so far. However, owing to recent technical advances, such a script engine can be installed in a spacecraft project. The author recommends T-Kernel as the OS kernel, which is expected to become a standard real-time OS, and JavaScript as the script engine to run on T-Kernel (Fukushima & Mita, 2011). Figure 16 depicts the proposed software architecture.

Most space systems can accept such commands that a software internal variable should be substituted by some value like "a_hardware_switch = 1". With the proposed script engine, it is possible to send commands by another form: an expression of a logic such as "IF a variable = value THEN do something", i.e., a program fragment. If space systems can accept a logic-type command, the system will have the capability to self-update dynamically; a script code sent by a logic-type command can modify the prescribed behaviours of the system. So far, software updates have only been achieved by stopping and reloading the entire software or by applying memory data patches on the running software code, despite the concern that these methods of updating have some risk of corrupting the system. Thus, such logic-type commands are clearly advantageous for our proposed software architecture.

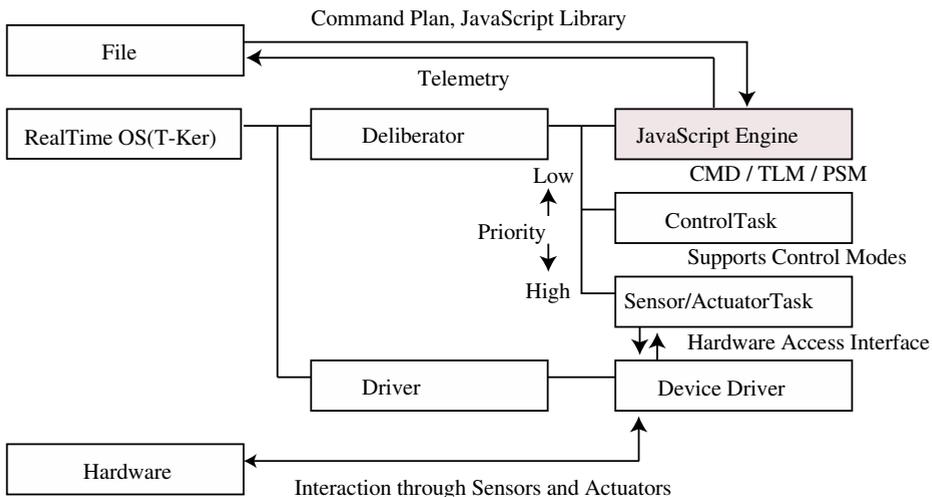


Fig. 17. A sample recommended implementation of the three-layer architecture for a remote system using a real-time OS with the deliberator that is realized by the help of a text-script engine, such as JavaScript, to provide onboard dynamic and self-update mechanism for users.

6. Conclusion

This chapter shows an experimental implementation of telemetry data processing to obtain a hidden event using a data mining technique. As a concrete example, detection of signs of actual satellite attitude sensor hardware failure was considered.

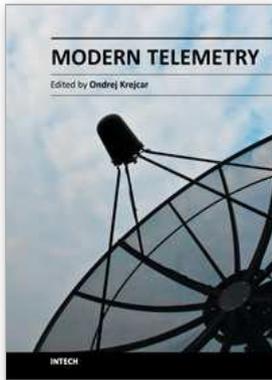
As preparation of treatment of practical data analysis commonly performed in satellite operations, a basic concept and practice of quantitative telemetry data analysis was demonstrated using REIMEI satellite data. By this preparation, the reader understood what the

FOG bias drift means and how it can be estimated from telemetry data of attitude motion. Then, an SVM was designed and tested to monitor the instability of the FOGs in the REIMEI satellite using 2 years of telemetry data. The result shows that the SVM can detect changes in bias with a simple linear kernel. Thus, the classification part of the SVM logic can be implemented on the flight software without difficulty. As a hint for actual implementation onboard software, the three-layer architecture for onboard software was explained. The flight software version of the SVM using a script engine has not yet been tested in the REIMEI satellite, and the author is waiting for an opportunity to carry out an in-flight experiment. It appears that SVMs can be used as a standard autonomous software component not only for onboard software even in small satellite systems but also for monitoring telemetry data. Thus, the author conducted an experiment to export this concept of system monitoring onto an autonomous underwater vehicle as concept verification activities and obtain sound results as it was expected. However, the most fruitful result expected area seems to space systems and it is the final destination of this research.

7. References

- Balzano, V. & Zak, D. (2006). Event-driven James Space Webb Space Telescope operations using on-board JavaScripts, *Proceedings of SPIE*, 6274, 62740A, Orlando, FL, USA, May 24, 2006
- Cristianini, R. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*, Cambridge University Press, ISBN 978-052-870-19-3, Cambridge, UK
- Farrenkopf, R. L. (1978). Analytic steady-state accuracy solutions for two common spacecraft attitude estimators, *Journal of Guidance, Control, and Dynamics*, Vol.1, No.4, pp. 282-284, ISSN 0731-5090
- Fukushima, Y. & Mita, M. (2011). A New Approach to Onboard Mission Replanning Using Orthogonal Arrays, *Proceedings of 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Budapest, Hungary, July, 2011
- Fukushima, Y. (2008). Onboard Sensor and Actuator Failure Detection using SVM for Autonomy of Small Satellite Systems. *Proceedings of the 9th International Symposium on Artificial Intelligence and Robotic Automation in Space*, Los Angeles, CA, USA, January, 2008
- Fukushima, Y.; Sakai, S. & Saito, K. (2006). Flight Performance of the REIMEI Microsatellite Attitude Determination System, *Proceedings of the Small Satellite Systems and Services*, ESA-SP-625, Sardinia, Italy, September, 2006
- Ninomiya, K.; Hasimoto, T.; Kii, T.; Muranaka, N.; Uo, M.; Maeda, K. & Saitoh, T. (1994). In-Orbit Performance of ASCA Satellite Attitude Control System, *Proceedings of 17th Annual AAS Rocky Mountain*, Keystone, CO, USA, February 2 - 6, 1994
- Rabideau, G. ; Knight, R. ; Chien, S. ; Fukunaga, A. & Govindjee, A. (1999). Iterative Repair Planning for Spacecraft Operations in the ASPEN System, *Proceedings of 5th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Noordwijk, The Netherlands, June, 1999
- Saito, H.; Mizuno, T.; Tanaka, K.; Sone, Y.; Fukuda, S.; Sakai, S.; Okuizumi, N.; Mita, M.; Fukushima, Y.; Hirahara, M.; Asamura, K.; Sakanoi, T.; Miura, A.; Ikenaga, T. & Masumoto, Y. (2005). AN-OVERVIEW AND INITIAL IN-ORBIT STATUS OF "INDEX" SATELLITE, *56th International Astronautical Conference*, IAC-05-B5.6.B.05, Fukuoka, Japan, 2005

- Sakai, S.; Fukushima, Y. & Saito, K. (2006). In-orbit Performance Evaluation of Temperature Controlled Small Fiber Optical Gyro on Microsatellite REIMEI, *18th International Conference on Optical Fiber Sensors Topical Meeting, TuE3*, Cancun, Mexico, October 23, 2006
- Sakai, S.; Fukushima, Y.; Saito, H. & Kaneda, R. (2006). Studies on Magnetic Attitude Control System for the REIMEI Microsatellite, *AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2006-6450*, Keystone, CO, August 21 - 24, 2006



Modern Telemetry

Edited by Dr. Ondrej Krejcar

ISBN 978-953-307-415-3

Hard cover, 470 pages

Publisher InTech

Published online 05, October, 2011

Published in print edition October, 2011

Telemetry is based on knowledge of various disciplines like Electronics, Measurement, Control and Communication along with their combination. This fact leads to a need of studying and understanding of these principles before the usage of Telemetry on selected problem solving. Spending time is however many times returned in form of obtained data or knowledge which telemetry system can provide. Usage of telemetry can be found in many areas from military through biomedical to real medical applications. Modern way to create a wireless sensors remotely connected to central system with artificial intelligence provide many new, sometimes unusual ways to get a knowledge about remote objects behaviour. This book is intended to present some new up to date accesses to telemetry problems solving by use of new sensors conceptions, new wireless transfer or communication techniques, data collection or processing techniques as well as several real use case scenarios describing model examples. Most of book chapters deals with many real cases of telemetry issues which can be used as a cookbooks for your own telemetry related problems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yosuke Fukushima (2011). Telemetry Data Mining with SVM for Satellite Monitoring, Modern Telemetry, Dr. Ondrej Krejcar (Ed.), ISBN: 978-953-307-415-3, InTech, Available from:

<http://www.intechopen.com/books/modern-telemetry/telemetry-data-mining-with-svm-for-satellite-monitoring>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.