Implementation of Bus-Type Architecture with Common Subsystem in Microsatellite ORBIS

By Hirohisa ASANO$^1$ and Hironori SAHARA$^1$

$^1$Department of Aerospace Engineering, Tokyo Metropolitan University, Tokyo, Japan

(Received June 23rd, 2017)

We propose a method that enables a satellite to decide its operation mode by using a distributed architecture with a common signboard subsystem, where each subsystem is regarded as an independent player in a game; we then simulate the operation mode and its transition in the satellite, based on game theory. The status of the satellite is expressed with the remaining-battery ratio and data-accumulation ratio on two axes, and we define two thresholds for each axis. By adopting a dynamic cooperative game model, in which the thresholds are determined dynamically, the mode transition was adjusted depending on the satellite status, and the time efficiency of the mission mode was improved.

Key Words: Microsatellite, Common Signboard Subsystem, ORBIS

1. Introduction

There are currently two methods of satellite development. Under one method, large and highly functional satellites are developed within budget restrictions to satisfy usage requirements. The other hand, microsatellites, which have masses under 100 kg, have increasingly been proposed by universities since 2000, because they have advantages in terms of cost and development time. As shown in a recent study, nearly 2,400 nano/microsatellites are set to be launched between 2017 and 2023.

Indeed, many microsatellites have been developed and launched, but the number of microsatellites that function correctly remains low. This is indisputably due to satellite systems becoming more complicated because of enhancements in requirements. Because even microsatellites face an increasing number of functional requirements, their systems become more sophisticated. Such sophistication disturbs their proper operation during the mission and significantly increases labor, thus negating the advantages offered by microsatellite development. In fact, a recent study found that the mission success rate of microsatellites since 2002 has been less than 50%.

In general, a satellite is expected to work autonomously in orbit, i.e., the operation mode of a satellite is uniquely dependent on a set of preliminarily configured branch conditions. However, such a pre-configuration is actually very difficult to predict accurate value in a complex system with massive branches, and imperfect implementation of the conditional branches impairs the success rate of microsatellites; therefore, for a complex system such as a microsatellite to work autonomously depending on the situation, a different method is required to increase the success rate of microsatellites with more mission time.

In a system, a set of members that have similar interests can be defined as a subsystem, for example, attitude determination and control subsystem, electric power subsystem, or communication subsystem in a satellite. As there is often a conflict of interest among the subsystems, judgments or condition adjustments of branches can be established by negotiation among the subsystems. Distributed architectures are also taking account of testability, which possibly has flexibility for environment adaptability.

Indeed, distributed architecture has been partially adopted in satellites with subsystems connected via SpaceWire to decrease work effort for system integration, but this has not resulted in a completely autonomous system.

Here, we are currently developing a microsatellite, ORbiting Binary black-hole Investigation Satellite (ORBIS), based on a distributed architecture with a common signboard subsystem (DACSS); we propose that it will decrease work effort and also secure system autonomy.

In this study, we evaluate the efficiency of an algorithm that enables a satellite to decide its operation mode and maximize its mission time by using a DACSS.

2. DACSS, Multiagent System and Game Theory

2.1. DACSS

Figure 1 shows a conceptual diagram of a DACSS, which is categorized as a function distribution system. The subsystems in a DACSS are connected to a bus line; each subsystem can access other subsystems via the common signboard subsystem in order to function autonomously. Therefore, the test items in the development of a satellite with DACSS are limited to the relations between each subsystem and the common signboard subsystem.
2.2. Multiagent system and game theory

In this study, we regard the architecture of a satellite as a multiagent system and each subsystem in it as an agent. Furthermore, we treat each subsystem as an independent player in a game. Therefore, we simulate the operation mode and its transition in a satellite based on game theory, to create a cooperative solution in a distributed architecture in the satellite.

We use ORBIS as the model for the simulation of operation mode and its transition in a satellite. The “mode” represents the controllable status of a satellite, and the “transition” indicates transition from one mode to another. The ORBIS system comprises the following subsystems: mission (MISN), attitude determination and control (ADCS), electric power (EPS), communication (COMM), and a common signboard (COMMON); however, only MISN, EPS, and COMM can generate a trigger for mode transition in the model of mode transition.

We set three modes in the ORBIS model as follows:
1. MISSION MODE, in which ORBIS observes a targeted astral body and obtains observation data.
2. STANDBY MODE, in which ORBIS recharges its batteries under power shortage.
3. DOWNLINK MODE, in which ORBIS transmits the observation data to the ground station.

2.3. Conditions for the simulation

Conditions for the simulation, based on the design specifications of ORBIS, are summarized in Table 1. Additionally, the power consumption under each mode and the specifications of EPS are presented in Tables 2 and 3, respectively. The total capacity of the batteries includes the depth of discharge, and the amount of power generated under sunshine and shade are both assumed constant. Details of the total data capacity and communication bit-rate are listed in Table 4. All the values given in Tables 1, 2, 3, and 4 are the designed values in ORBIS.

![Fig. 2. Status of ORBIS and the defined thresholds.](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Simulation time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative time between real time and simulation time</td>
<td>0.01</td>
</tr>
<tr>
<td>Experiment simulation time (10 days)</td>
<td>8640</td>
</tr>
<tr>
<td>DOWNLINK time per day</td>
<td>23.07</td>
</tr>
<tr>
<td>Sunlight cycle</td>
<td>57.3</td>
</tr>
<tr>
<td>Sunlight time per cycle</td>
<td>35.98</td>
</tr>
<tr>
<td>Shade time per cycle</td>
<td>21.32</td>
</tr>
</tbody>
</table>

Table 1. Conditions for the simulation.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>MISSION MODE</th>
<th>STANDBY MODE</th>
<th>DOWNLINK MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPS [W]</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>COMM [W]</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>MISN [W]</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ADCS [W]</td>
<td>40</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>COMMON [W]</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 2. Power consumption under each mode.

| Battery total capacity [J] | 416,793 |
| Housekeeping (HK) data communication [W] | 100 |

Table 3. Specifications of EPS.

Table 4. Total data capacity and communication bit-rate.

| Total data capacity [kbit] | 23,069 |
| Housekeeping (HK) data communication [bps] | 250 |
| Mission data communication bit-rate [during MISSION MODE] [bps] | 250 |
| DOWNLINK communication bit-rate [bps] | 10,000 |

2.4. Mode transition method

The status of ORBIS is expressed with the remaining-battery ratio and data-accumulation ratio on two axes, and we define two thresholds for each axis (i.e., there are four thresholds, C₁, C₂ and D₁, D₂), as shown in Fig. 2. The remaining-battery ratio is the ratio of the current battery capacity to the total capacity, and the data-accumulation ratio is the ratio of the currently accumulated data to the assumed total amount of observation data. The four thresholds on the two axes result in nine rectangular sections, as shown in Fig. 2.
2.5. Threshold configuration model with cooperative repeated game

We defined the EPS, COMM, and MISN subsystems as players in a cooperative game. Each subsystem follows its own rationality and arranges its respective thresholds to obtain the maximum reward as an area of the section, $S_1, ..., S_{33}$, as shown in Fig. 4.

The individual rewards of EPS ($R_{\text{EPS}}$), COMM ($R_{\text{COMM}}$), and MISN ($R_{\text{MISN}}$) are given as follows:

$$R_{\text{EPS}} = S_{13} + S_{21} + S_{22} + S_{23} + S_{31}$$  \hspace{1cm} (1)
$$R_{\text{COMM}} = S_{12} + S_{13} + S_{22} + S_{23}$$  \hspace{1cm} (2)
$$R_{\text{MISN}} = S_{11} + S_{12}$$  \hspace{1cm} (3)

Additionally, we set the group reward in the whole system, $R_{\text{ALL}}$, to maximize $R_{\text{MISN}}$:

$$R_{\text{ALL}} = S_{11} + S_{12}$$  \hspace{1cm} (4)

As a result, we describe the rewards of the subsystems $R'_{\text{EPS}}$, $R'_{\text{COMM}}$, and $R'_{\text{MISN}}$ as follows in the cooperative game:

$$R'_{\text{EPS}} = R_{\text{EPS}} + \lambda_{\text{EPS}}R_{\text{ALL}}$$  \hspace{1cm} (5)
$$R'_{\text{COMM}} = R_{\text{COMM}} + \lambda_{\text{COMM}}R_{\text{ALL}}$$  \hspace{1cm} (6)
$$R'_{\text{MISN}} = R_{\text{MISN}} + \lambda_{\text{MISN}}R_{\text{ALL}}$$  \hspace{1cm} (7)

where $\lambda_{\text{EPS}}$, $\lambda_{\text{COMM}}$, and $\lambda_{\text{MISN}}$ represent the relative importance factors between the individual and the group.6)

Then, we set the strategies of the subsystems, $X'_{\text{EPS}}$, $X'_{\text{COMM}}$, and $X'_{\text{MISN}}$, in the cooperative game by adopting the following rules for moving the thresholds $C_1$, $C_2$, $D_1$, and $D_2$:

1. Each agent selects all the thresholds it needs.
2. The difference between the ideal value for each agent and the current value for the selected thresholds is calculated.
3. The value calculated in the previous step is multiplied by 0.01, and the threshold is moved by this value.

Based on the above rules, the strategies of the subsystems become pure strategies depending on the relative importance factors, which are represented as follows:

$$X'_{\text{EPS}} : C_1 \rightarrow C_1 + 1 - 0.01(1 + \lambda_{\text{EPS}})C_1,$$
$$D_2 \rightarrow D_2 + \lambda_{\text{EPS}} - 0.01(1 + \lambda_{\text{EPS}})D_2,$$
$$X'_{\text{COMM}} : C_1 \rightarrow C_1 - 0.01\lambda_{\text{COMM}}C_1,$$
$$D_1 \rightarrow D_1 - 0.01D_1,$$
$$D_2 \rightarrow D_2 + \lambda_{\text{COMM}}(1 - 0.01D_2),$$
$$X'_{\text{MISN}} : C_1 \rightarrow C_1 - 0.01(1 + \lambda_{\text{MISN}})C_1,$$
$$D_2 \rightarrow D_2 + (1 + \lambda_{\text{MISN}})(1 - 0.01D_2).$$

2.6. Dynamic threshold configuration model of cooperative game

The thresholds are determined when the relative importance factors are defined in a cooperative game. Utilizing this system, we examine the mode transition of ORBIS by adjusting the relative importance factors under dynamic circumstances. The relative importance factors are equated, $\lambda = \lambda_{\text{EPS}} = \lambda_{\text{COMM}} = \lambda_{\text{MISN}}$, and defined as

$$\lambda = 20 \times RB^2,$$

where $RB$ ($0 < RB < 1$) represents the remaining-battery ratio of ORBIS.

3. Results

The histories of $C_2$ and $D_1$ in the dynamic threshold configuration model of cooperative game are shown in Fig. 5 and 6, respectively. Notably, both thresholds converge to zero.

Fig. 5. Lower remaining-battery ratio threshold, $C_2$, in the dynamic cooperative game model.
Figure 6 shows the history of $C_1$ in the dynamic cooperative game model. This threshold does not converge to a constant value; it fluctuates between 10% and 50% 30 s after the start of the simulation.

Figure 8 shows the history of $D_2$ in the dynamic cooperative game model. Similar to $C_1$, it fluctuates between 50% and 90%.

The remaining-battery ratio from the dynamic cooperative game model is shown in Fig. 9. It rapidly decreases to 5% until 300 s and subsequently fluctuates from 5% to 35%.

The data-accumulation ratio from the dynamic cooperative game model is shown in Fig. 10. It varies between 0% and 70% with a period of 800 s.

4. Discussion

In the dynamic cooperative game model, the relative importance factor is changed dynamically according to the remaining-battery ratio. From Fig. 7 and 8, it can be seen that $C_1$ fluctuated between 10% and 50% and $D_2$ periodically transitioned from 50% to 90% after 200 s. These fluctuations occur because the relative importance factor changes such that MISN becomes dominant when the remaining-battery ratio is high, whereas EPS has priority when the remaining-battery ratio is low.

In the dynamic cooperative game model, the constant of 20 in Eq. (11) was intentionally selected to prevent the remaining-battery ratio from falling below zero. From Fig. 9, it can be seen that the remaining-battery ratio did not reach zero and fluctuated between 5% and 35% after 300 s.

Here, we evaluate the above algorithm for mode transition according to the following two indices:

1. Is the duration of MISSION MODE long?
2. Is the time efficiency of MISSION MODE high?

As for index 2, we first define an ideal mission time [s/day], $I_{mt}$:
where $Dt$ is the DOWNLINK time [s/day], $Dcs$ is the DOWNLINK communication bit-rate [bps], $Mct$ is the mission communication bit-rate [bps], and $Hcm$ is the HK data communication bit-rate [bps]. When using the conditions summarized in Table 1, the ideal mission time is obtained as $46,140$ s/day. We then calculate the time efficiency of MISSION MODE, $\psi$, as the ratio of the simulated mission time we have already obtained, $Mmt$, to the ideal mission time:

$$\psi = \frac{Mmt}{\text{Imt} \times 10 \text{ days}}.$$  \hspace{1cm} (13)

Table 5 shows the durations of each mode in the cases of the dynamic algorithm and a fixed importance factor. In addition, the result in the case of the conventional mode transition with the values of $C_1 = 40, C_2 = 20, D_1 = 20, D_2 = 80$ as a set of fixed thresholds under all circumstances is also included in Table 5. Only the duration of STANDBY MODE in the dynamic model is shorter than that in the cooperative game ($\lambda = 0.5$). However, the durations of DOWNLINK MODE and MISSION MODE are longer, increasing the time efficiency of MISSION MODE. This occurs because the remaining-battery ratio is relatively low; therefore, the operation time is used for MISSION MODE and not for battery recharge. MISSION time efficiency of the dynamic algorithm is 0.3% higher than that of the cooperative game. This implies that the cooperative game would prolong the mission time by 52.56 h. Therefore, the observed improvement to MISSION time efficiency suggests that the dynamic algorithm is more suitable for threshold determination under the considered circumstances.

5. Conclusion

In this study, we proposed a method for mode transition in a satellite by changing the thresholds between modes based on the satellite status by using DACSS. We evaluated the method with respect to the time efficiency of MISSION MODE and compared the dynamic algorithm to a purely cooperative game. The time efficiency using the dynamic algorithm showed marked improvement over the static mode transition method.

In an actual satellite in orbit, the satellite status should be defined by other factors in addition to the remaining-battery ratio and data-accumulation ratio, e.g., the combination of rotation speed of reaction wheels in the ADCS and the status of data transmission in the satellite. Therefore, we need to define more segmented modes in the satellite and introduce more adequate thresholds amongst the modes.

Acknowledgments

This study was supported by the Ministry of Education, Culture, Sports, Sciences, and Technology (MEXT) through the commissioning funds for promoting aerospace science and technology.

References