Keywords teleoperation, space robot, time delay, operator aid, user interface

Abstract
The existing teleoperation methods, which verify the command only as a single command not as a series of commands, have poor operability under the existence of communication time delay. This is because an inappropriate operation can only be stopped just before the execution and the following restarting operation becomes very difficult. This paper suggests a new on-line teleoperation method called Additional Time Delay Method and reports the results of its application to the satellite experiment on the Engineering Test Satellite 7.

This new method is equipped with the additional delay mechanism, which adds some additional time delay to the communication time delay to maintain a command buffer. Every verified command is temporarily stored in this buffer before transmission and, when a wrong command is detected later, part of the commands or all the commands in the buffer are automatically re-edited to make a new series of commands which smoothly avoid the wrong operation. The operation continues smoothly from the last command point that was left after the re-editing.

In our application for the satellite experiment, this method was applied in order to reduce the excessive force in manipulating the truss structure.

1. Introduction
The shortage of crew time on the International Space Station will become a serious problem in the near future. And large-scale space activities, such as the construction of on-orbit structure or lunar base, are expected in the not too distant future. Considering these factors, ground or on-orbit teleoperation by human operator can be a realistic solution to realize safe and flexible operation on a large scale. What will be needed is the practical operator support technology that enables more than one operator to work comfortably for many hours.

In teleoperation, operators need to be careful not to make wrong commands, because it is usually difficult to restore the operation in the remote site once wrong operations occur. This stresses the operator in greater or lesser degree and makes the long-hour operation almost impossible.

This is especially true in the ground teleoperation where communication time delay is inevitable. When objects in the low earth orbit are controlled from the ground, several seconds of communication time delay (around 5.5 seconds in the case of the Engineering Test Satellite 7 launched in 1997) exist. Since the confirmation of the executed command is always delayed in several seconds, the operation tends to adopt move-and-wait strategy, a repetition of small pieces of operation, for safety reason. Without a proper aid system, the operation gets extremely inefficient. Though many operation support systems, such as the predictive display, have been tried by now, the existing methods do not seem to aim at easing the restoration of wrong operation or reducing the operator’s stress regarding the wrong operation.

Our newly developed teleoperation method is equipped with an on-line command re-editing. It realizes the re-editing of the teleoperation commands by introducing the additional time delay to reserve every verified command for a while before transmission. This method can remove the improper commands without losing the smooth operability of the online commanding. This method does not stress the operator, allowing them to make wrong commands by accident. If an inappropriate command is detected, this method re-edits it together with the preceding commands into a series of proper commands. The additional time delay method realizes the smooth and continuous operation.

There could be many applications for this method, and the simplest one can be the collision avoidance in non-contact operation, like avoiding obstacles during an approach to a target object. But, the use of this method is not limited to the collision avoidance. This can be also used to decrease excessive force in contact operation. In this paper, we report the way we implemented this method to decrease excessive force and the results of the experiment conducted on the Engineering Test Satellite 7.
2. Additional Time Delay Method

In this section, the algorithm of the additional time delay method is summarized. For detail, see [1].

2.1 Outline

The existing teleoperation techniques, especially the on-line techniques with predictive display, are certainly capable of judging the suitability of a single command and stopping the execution of it, if necessary. Therefore, it can be said that they are effective to some extent as a safe operator support system in ordinary operation.

The existing systems, however, are not always sufficient in respect of operability in the off nominal operation, because they do not facilitate the restart operation required after being stopped due to wrong operation. When an improper command is detected, the operation can only be stopped just before the execution of the wrong command, because all the preceding commands have already been transmitted and executed. Thus, it will be a difficult work to restart the operation that was stopped at the last moment. For example, when a robot arm is stopped just before a collision in a narrow workspace, it is likely that the robot could not move in any direction for lack of enough clearance.

But such situations are due to not only the improper command itself but also a series of the preceding commands. In other words, the existing techniques do not sufficiently verify the commands as a series of commands, and thus they do not re-edit them into a proper series of commands in the continuous operation. It is necessary to cope with an improper command in the continuous operation, as well as just to stop the execution of it. If every teleoperation command, which is already verified as a single command, is stowed in a command buffer before transmission for a while, all the commands or part of the commands in the buffer can be re-edited to make the smooth and appropriate commands, when a wrong command is detected.

The existing methods have some ways to prevent wrong operation itself. For example, some use restraint planes in the vicinity of objects to avoid getting too close to them. But there will be a limit for the use of these kinds of guiding, because they assume pre-defined tasks in a known environment. They not only require enormous database as the tasks diversify but also restrict the task execution. Some other method forces the operator to carefully carry out the operation by making the tasks intentionally difficult. But this method only increases the psychological burden and is not practical for long hour operation.

Figure 1(a) shows a general teleoperation system. The commands are transmitted to the on-orbit object through the control and communication system on the ground, and the telemetry data is also returned through the ground system. One cycle of operation, from generating commands to receiving the telemetry, usually takes several seconds, and this time lag is the communication time delay. Thus, the continuous operation becomes difficult, because it has always to wait for several seconds until the execution result is obtained. Several time-delay compensation methods have been developed to cope with this problem.

![Fig.1 Teleoperation System](image-url)

The teleoperation system with the additional time delay method is shown in Fig.1(b). The on-line revision of command is realized by inserting the re-editing mechanism between the operator and the ground control system.

The basic idea is realizing the re-editing of a series of commands, not just the elimination of a single
wrong command, by storing every verified command in the command buffer for a defined time period before transmission. The time period during which a command is stored in the buffer is the additional time delay. The addition of this additional time delay to the inevitable communication time delay lengthens the total time delay. Each command is transmitted after the additional time delay passes. In other words, a fixed number of the latest commands are always stored in the buffer. In the range of these stored commands, the deletion and revision of commands become possible.

The addition of the re-editing mechanism realizes the smooth avoidance of wrong operation in continuous operation. For example, in case of the collision avoidance, it is possible not only to stop the collision but also to regenerate a smooth avoiding trajectory on-line by revising the past commands.

2.2 The Structure of the Re-editing Mechanism

Figure 2 is the structure of the on-line re-editing mechanism. It shows the flow of command and information between the operator and the operated object.

The ordinary operation, where no wrong operation happens, is carried out along the bold line. The operator generates a command on a simulator equipped with some operation aid system like predictive display. Here, one unit of command can be defined arbitrarily. It can be continuous commands with several msec cycle, a set of command programs, or else. The generated command and its simulation result are first sent to the operation control mechanism to verify it as a single command. If the command is judged appropriate, it is next sent to the store-and-delay mechanism to be stored in the buffer for the additional time delay. After the additional time delay passes, the command is then transmitted to the operated object and executed. The telemetry data is returned to the operator and the operation control mechanism.

If a new command is verified inappropriate at the operation control mechanism, the re-editing starts and carries out the operation shown in the fine line. Though there can be various ways for re-editing, one fundamental method is introduced here. This method consists of two stages. In the first stage, a defined number of commands in the buffer are eliminated. Then, in the second stage, the deletion is compensated by refilling the buffer with new commands smoothly connected to the last command left in the buffer.

2.3 Example of Re-editing

Figure 3 shows an example flow of the re-editing. In this figure, the rectangular box represents that command buffer which functions as a shift register. Generated commands enter the buffer one by one from the left side. The buffer can store defined number of commands. As a command enters the buffer, the commands are pushed in order to the right side and, if they reach the right end, they are transmitted one by one. The time during which one command moves in the buffer corresponds to the additional time delay. The inevitable communication time delay of the system is not described in Fig. 3, because it occurs after the transmission.

In this example, the unconditional elimination is used as the elimination method and the suspension of transmission as the restoration method [1]. In the ordinary operation, the buffer is always full (1). When a succeeding command is judged inappropriate, the elimination method is executed (2).
The defined number of commands is unconditionally eliminated and the operation point is automatically returned to the point of the latest command left in the buffer in a moment. The operation continues and the transmission is suspended until the buffer becomes refilled (3). And when the buffer is refilled, the operation returns to the ordinary operation (4). If this method is used for collision avoidance, for example, the operator is allowed to make collisions on the simulator and do not need to go back themselves to avoid the obstacle. When collisions occur, the command point will be automatically move back to a certain past point and the operator can regenerate the smoothly avoiding trajectory again.

Assuming the task of peg-in-hole as an example. If you define the contact between the peg tip and the hole edge on the predictive simulator as the trigger of re-editing, the peg looks like bouncing back to a past command point every time it collides with the hole edge. The operator can try again without making actual collision in the remote site. The operator can even try to find the most stable point of insertion by intentionally colliding the peg with the hole edge. Being able to do this kind of trial-and-error operation is an advantage of the additional time delay method.

This method is independent on the quantity of the communication time delay and able to keep the continuous operation by eliminating the wrong commands naturally. It is even possible to add some optimizing methods for re-editing. For precise operation, setting the adequate quantity of the additional time delay is necessary.

3. Deployable Truss Experiment on ETS-7

3.1 Engineering Test Satellite 7 (ETS-7)

Engineering Test Satellite 7 (ETS-7), launched by NASDA Nov. 28, 1997 in Japan, had been developed to demonstrate two major missions, the rendezvous docking and the space robotics (Fig. 4). For the space robotics, it is the world first robotic-arm teleoperation experiment satellite [2].

3.2 Truss Structure Experiment (TSE)

The TSE unit had three components, the launch lock, the deployable truss, and the truss joint. The deployable truss (DT) had 10 degrees of freedom, and was a part of a triangle truss structure that was statically determinate and was able to be deployed and folded (Fig. 5)[3].

Fig. 4  ETS-7 (Engineering Test Satellite 7)

Fig. 5  TSE Deployable Truss
a) Deployed Position, b) Stowed Position, c) Deployment Trajectory in World Coordinates (x, y and z)
The arm grappled the grapple fixture installed on DT and deploys it along a three dimensional spline curve under closed link arm control. The technical difficulty is to move the arm along the trajectory within suitable tip force and torque. The proper deployment required synchronized input of translation and rotation. The closed link movement along a strictly defined trajectory is the first operation for ETS-7.

3.3 Visual Aid System

We developed and tested a new visual aid system for deploying the truss that calculated the appropriate joystick input on-line and showed it by indicating the directions the joysticks should be moved to. These input directions were converted into the joystick coordinate and shown to the operator to enable the operator to follow the direction. In this way, the operator easily handled the complex operation without doing move-and-wait.

![Fig. 6 Aid Display for TSE DT Operation](image)

Fig. 6 shows the aid display we used for ETS-7 truss deploying experiment. The upper part of the display shows the overview of the TSE and the lower part shows the operation information. The window in lower right shows the force-torque sensor value and the image processing result. The lower left window is the down link image of the on-board hand camera. The two windows in the lower center are the direct input aid for joystick. The left window corresponds to the translation input and the right window to rotation input.

Fig. 7 shows the detail of the input aid. The left window is the translation aid. The bold bar is to indicate the input direction parallel to the trajectory and the red box is to indicate the force-release direction. The thin bar indicates the current operator input. The display matches the physical coordinate system of the input device, joystick in this case, in order for the operator to quickly follow the direction. The joystick axes are assumed to be vertical to the windows. If the bold bar points to the left, the operator is supposed to push the joystick to the left to execute deploying force. If the box is not around the window’s center, the operator is supposed to push the joystick in the red box’s direction to release excessive force. Thus, the ideal input should be the direction between the bold bar and the box like the one shown in Fig. 7. The vertical line in the right side indicates the third axis that is vertical to the window.

![Fig. 7 Joystick Input Aid](image)

The left window is for rotation input. The fan-shaped object shows the roll rotation allowance, because the DT’s grapple fixture is not completely fixed and can rotate to latch or unlatch. If the two thin bars, one for the current roll position and the other for the command roll position, are within the fan, the excessive roll torque does not appear. The center box just shows the window center. The other box, almost in the center, shows the pitch and yaw input. They are calculated based on the difference between the current value and the command value, which corresponds to the theoretical torque. The pitch and yaw input is expected to reduce the difference.

These aid bars and boxes kept moving during the operation, and the operator followed them.

For the calculation of the appropriate input, we developed a new method called the predictive force method [4] that used only the past trajectory and the present status, without using designed data of the trajectory, to theoretically estimate the current force (predictive force) executed to the truss. Predictive force is an alternate for the force sensor value under time delay, calculated based on the force control algorithm using the current command position and the delayed telemetry position.

4. Excessive Force Reduction by Additional Time Delay Method

4.1 Re-editing Method

We applied the additional time delay method
to the deployable truss operation in order to prevent excessive force and ease the operation for the operator. Here, excessive force means unnecessary force. It does not necessarily mean large force that leads to destruction of hardware. We tried to realize joystick operation with lower force, because the inaccuracy of the human operation tends to produce larger force than required.

Instead of starting re-editing at some physical contact, we defined the magnitude of the predictive force as the trigger of the re-editing. When the predictive force exceeded a defined value, the re-editing was started. The commands in the buffer were eliminated until the predictive force got lower than the defined value, or until the buffer became empty. For example, the re-editing was started at the predictive force of 6N and the commands were eliminated until it became less than 5N. In the actual experiment, additional time delay was set at 10 seconds, and the trigger forces for the re-editing was set at 4N, 5N, 6N for stowing operation, and at 8N, 9N, 10N for deploying operation, because different force level was required at the different point of the trajectory due to the truss dynamics. Re-editing was carried out until the predictive force decreased by 1N.

4.2 Comparison of Trajectories

Fig.8 compares the re-edited command trajectory with the actual trajectory on the x-y plane with the origin at the initial grappling point of stowing. The coordinate system is shown in Fig.10(a).

Since completely following the large curvature was difficult, two re-editings were executed, when the command trajectory deviated and predictive force exceeded the threshold. The effect of re-editing was much clearer in deploying the truss (Fig.9). The coordinate system is shown in Fig.10(b). Truss deployment needed bigger force than stowing, because a spring was put in the truss joint that automatically stowed the truss in emergency. This meant that the command trajectory needed to exceed the actual trajectory in order to overcome the spring force and deploy the truss under compliance control. But, if the command exceeded too much, it would cause big excessive force. Fig.9 shows that the command trajectory was restricted by the re-editing in order not to exceed the trajectory too much.

4.3 Comparison of Sensor Force

Fig.11 shows the transition of forces with the progress of time in stowing and deploying the truss sequentially. The first part of the operation, which ended in around nine minutes, was the truss stowing, which needed relatively small force, under 4N. The latter part of the operation was the truss deployment, which required bigger force, about 10N. The predictive force predicted the sensor force precisely enough to substitute the sensor force. The threshold force value for re-editing was set 4N in stowing, because the required force to stow the truss was expected constant. But, the threshold force value for deployment was increased in three steps during the operation, because the needed to force to overcome the spring force got bigger as the truss deployed.

Fig.12 compares the force sensor telemetries of program control, joystick operation without the additional time delay method, and joystick operation with the additional time delay method in stowing operation. Without the additional delay method, joystick operation generated larger force, but with the
additional delay method, joystick operation performed with as low force as the program control did. In the figures, the additional time delay method is called ADM for short, and the predictive force method is called PFM for short.

Fig. 11 compares the force sensor telemetries of program control, program control with the predictive force method and joystick operation with the additional delay method in deploying operation, though the latter two graphs almost completely overlap. Since program control with the predictive force method did not depend on the designed model of the trajectory, it did not produce excessive force as the ordinal program control did, and, for deploying operation, there was not a big difference between the force of program control and that of joystick operation compared with the stowing operation. This is because the needed force was very small for the stowing operation and it was difficult for the predictive force to precisely predict the sensor force. On the other hand the sensor force was stably predicted for deploying operation.

4.4 Comparison of Operation Time

Table.1 shows the experiment time and average forces needed to finish the operation. The program control was done using the designed trajectory or the predictive force method. For the joystick operation, the operator tried to keep the force as low as possible by strictly following the visual cues.

<table>
<thead>
<tr>
<th></th>
<th>Time (min)</th>
<th>av. Force (N)</th>
<th>Maximum Force (N)</th>
<th>av. (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Control</td>
<td>Stow1 3.98</td>
<td>3.7</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stow2 6.82</td>
<td>5.2</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stow3 6.73</td>
<td>5.3</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stow4 5.07</td>
<td>5.8</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deploy1 6.1</td>
<td>13.0</td>
<td>10.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deploy2 12.35</td>
<td>11.6</td>
<td>10.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deploy3 9.23</td>
<td>8.7</td>
<td>9.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deploy4 6.1</td>
<td>9.1</td>
<td>9.1</td>
<td></td>
</tr>
<tr>
<td>Joystick Operation with ADM</td>
<td>Stow 6.0</td>
<td>5.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deploy 6.7</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joystick Operation without ADM</td>
<td>Stow 8.0</td>
<td>6.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table.1. Comparison of Force Telemetry
For the stowing operation, program control always operated with forces less than 6N in about six minutes on the average. With the additional delay method, the stowing operations were achieved with the same level of force and time as program control. But, without the additional delay method, the operation took longer time to achieve the same level of low force. With the additional delay method, the operator did not need to try to follow the visual cues with extreme carefulness, because the method told her when she generated commands that would lead to excessive force and eliminated them. Without the additional delay method, any command that were generated was executed and it was very difficult to follow exact trajectory needed to keep the force low.

The additional delay method also successfully supported the deploying operation, which was more difficult and required higher force due to the emergency spring, to keep the executed forces as low as in the program control.

In addition, the additional delay method reduced the mental stress of the operator. With this method, the operator can try again and do not need to worry about generating higher force. The point of this method is that it not only stops the operation but also automatically goes back to the past command. This makes the operation a lot easier. The operator, for example, does not need to move the joystick in reverse direction to go back to the safe position. They only need to slightly change the joystick tilt keeping commanding forward. For the real space operation where mistakes are not allowed, we think this kind of operator aid is necessary to ease the mental pressure of the operator.

5. Conclusion

Newly developed additional time delay method was implemented to support the direct teleoperation by joystick to deploy and stow the deployable truss of the truss structure experiment unit onboard the Engineering Test Satellite 7. This method reserves the verified commands for a while before transmission and, if an inappropriate command is detected, re-edits it together with the stored command to generate a series of proper new commands online. In the truss deployment experiment, the additional time delay method was used to keep the force low by re-editing the commands every time the predictive force exceeded a defined value. This method successfully supported the operator to conduct the deployment with low force. In addition, it made the operation easier for the operator, because, if wrong commands were generated, they were automatically eliminated and the operator was able to try again. The operator was even able to do trial-and-error operation.

Since the effectiveness of the additional time delay method was confirmed, we need to continue to test it to get more results with several subjects and applying it to other applications.

References