THE PROCEDURAL SAFETY SYSTEM

Maureen E. O'Brien

Goddard Space Flight Center
Greenbelt, Maryland

ABSTRACT

Telerobotic operations, whether under autonomous or teleoperated control, require a much more sophisticated safety system than that needed for most industrial applications. Industrial robots generally perform very repetitive tasks in a controlled, static environment. The safety system in that case can be as simple as shutting down the robot if a human enters the work area, or even simply building a cage around the work space. Telerobotic operations, however, will take place in a dynamic, sometimes unpredictable environment, and will involve complicated and perhaps unrehearsed manipulations. This creates a much greater potential for damage to the robot or objects in its vicinity. The Procedural Safety System (PSS), developed at GSFC's Robotics Laboratory, collects data from external sensors and the robot, then processes it through an expert system shell to determine whether an unsafe condition or potential unsafe condition exists. Unsafe conditions could include exceeding velocity, acceleration, torque, or joint limits, imminent collision, exceeding temperature limits, and robot or sensor component failure. If a threat to safety exists, the operator is warned. If the threat is serious enough, the robot is halted. The PSS, therefore, uses expert system technology to enhance safety thus reducing operator work load, allowing him/her to focus on performing the task at hand without the distraction of worrying about violating safety criteria.

Introduction

As we move from industrial automated robot applications toward telerobotic operations, particularly for space applications, the need for a sophisticated safety system increases dramatically. Industrial automated robots, which traditionally involve repeating pre-programmed "pick and place" operations, utilize unsophisticated sensing capabilities and typically incorporate a very limited amount of safety since each point that the robot is supposed to move to is pre-programmed. Telerobotics which involve both autonomous and teleoperated control performing a wide variety of tasks utilizing many different sensing capabilities must incorporate a great deal of safety because the motions of the robot are, for the most part, variable. A robot in a manufacturing plant, for example, may be tasked to drill a 1/2 inch hole in a sheet of metal. Every point that the robot is supposed to go to in order to drill the hole has been predetermined. Safety checks that are sometimes used involve using a sensor to detect if a human has entered the work area of the robot or using a sensor to detect if a robot has stopped its motion.

The Flight Telerobotic Servicer (FTS), the robot which will be used to service the Space Station Freedom, will be tasked to do a wide variety of tasks such as refueling a satellite, repairing a satellite and assembling the trusses for the Space Station. These types of tasks, unlike traditional industrial automated robot tasks, incorporate both autonomous and teleoperated control utilizing a great deal of sensing capabilities, requiring sophisticated safety systems. There are several functions that a complete safety system for telerobotic operations must incorporate. First, the safety system must be able to detect unsafe robot commands being sent from the robot control computer to the robot. Second, the safety system must be able to detect unsafe robot health status to ensure the robot is not malfunctioning. Third, the safety system must monitor all other systems such as the workstation computer, sensors, and robot controllers to ensure that they are operating. Finally, the safety system must be able to monitor all sensor data to ensure the task is operating under safe conditions. All of these functions must be incorporated to ensure the safety of humans, the robot and the objects in the robot environment.

Overview of the Safety Problem

These functions can be divided into two safety systems: the Watchdog Safety System (WSS) and the Procedural Safety System (PSS). The WSS provides safety at the robot servo level. The WSS is a separate system which exists between the robot control computer and the robot. It monitors all commands sent from the controller to the robot to ensure that the following have not been exceeded:

- velocity limits
- acceleration or motor torque limits
- joint limits.

The Watchdog Safety System, unlike the Procedural Safety System, checks absolute limits. It, for example, checks to ensure that the robot never exceeds a velocity limit of 250 mm/sec. The WSS must also monitor all robot status data to ensure that the following are not present:

- temperature limits exceeded
- incorrect position reached.

All other systems such as the sensors, the robot and the workstation computers must also be monitored by the WSS to ensure that they are operating.

This paper focuses on the Procedural Safety System (PSS) developed at the Goddard Space Flight

Center's Robotics Laboratory which is an expert system which provides safety for operating the FTS. The PSS exists at a higher level than the Watchdog Safety System. It, unlike the WSS which checks absolute limits, detects unsafe conditions based on the operational limits of the step of the task. Sensor data and commands are sent to PSS which checks this data against the operational limits of the task as shown in Figure 1. The PSS, which exists between the operator interface and the robot controller, obtains all sensor data and commands from the sensors and the operator. It compares the data and commands with the operational limits of the present step of the task. If the data or commands lie outside of the operational limits, the PSS sends messages to the operator interface to warn the operator or sends commands to the robot controller to stop the robot motion.

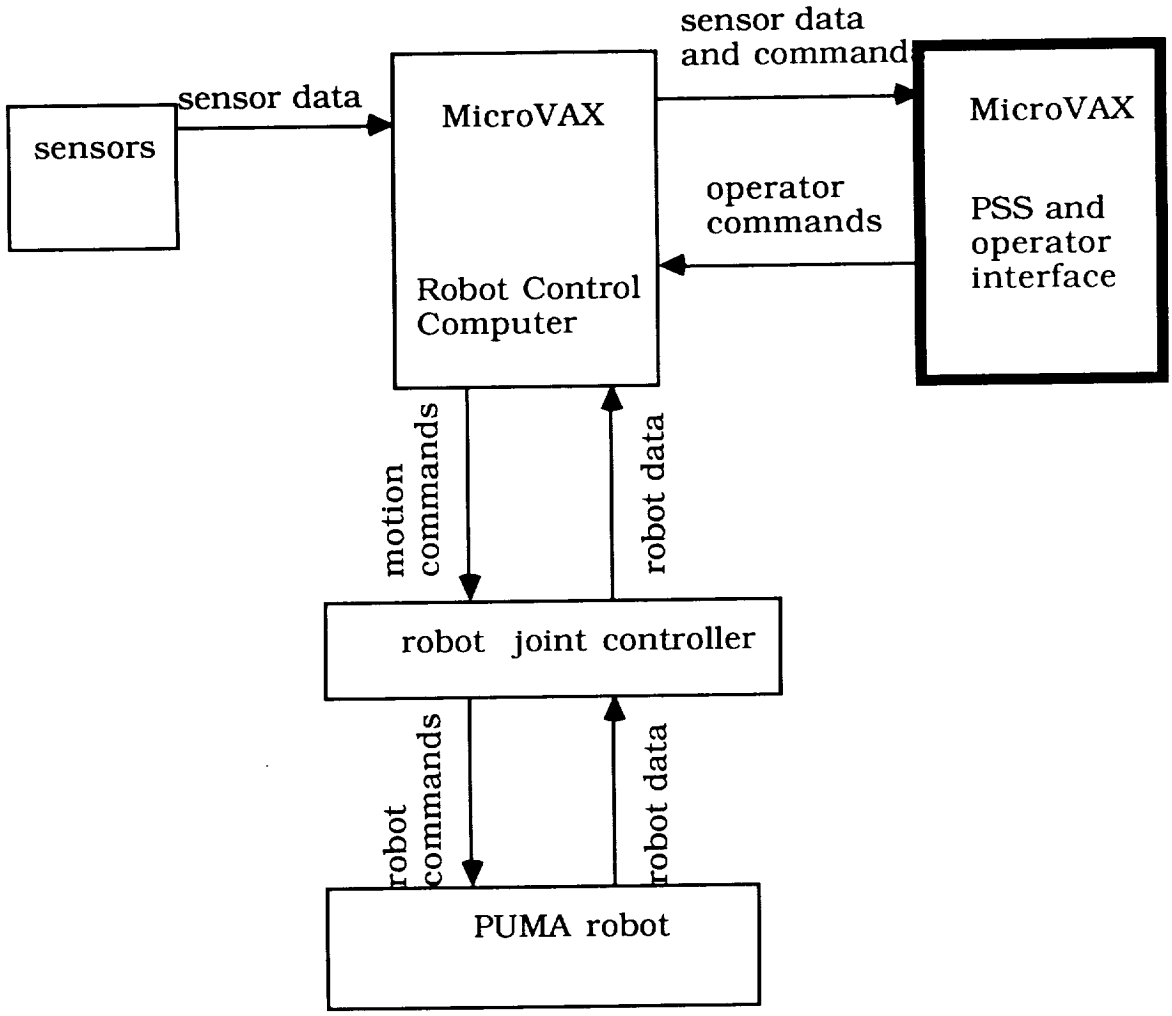

Figure 1. GSFC Robotics Laboratory Procedural Safety System Layout

The Application

The Procedural Safety System was added to the Orbital Replacement Unit (ORU) demonstration that exists in the robotics laboratory. The ORU demonstration involves using a PUMA robot to move an ORU, a generic housing for a flight experiment, from one position on a platform to a second position. The robot then opens the door of the ORU, allowing the robot to replace submodules or repair the experiment. Prior to the implementation of the PSS, the ORU demonstration incorporated very few safety checks. The burden of ensuring that the task was operating safely was placed on the operator. The PSS relieves this burden by monitoring sensor data and commands and reporting anomalies to the operator.

The ORU task was broken down into steps as shown in figure 2. Each step must be completed and conditions must exist in order to continue to the next step. For example, the gripper must be latched to the ORU handle before the robot can move it to the second position on the platform. Associated with each step of the task are the operational limits for the various sensor data. Figure 3 shows the operational limits for steps one and two. When step one of the task is being executed no forces or torques should exist, the gripper should be unlatched, and the switches on the platform indicting which position the ORU is in should indicate that the ORU is in position one (switches 1,2 and 3 connected and switches 4,5 and 6 disconnected). Operational limits for step two are also shown in figure 4. These operational limits were determined by using a computer program to obtain the minimum and maximum values of the sensor data as the ORU demonstration was performed by several different telerobot operators, multiple times. A certain percentage was then added to these values to account of noise in the readings.

TASK : replace an orbital replacement unit (ORU)

Step 1 : goto point above oru door handle

Step 2: seat on door handle

Step 3: latch gripper to handle

Step 4: move ORU to position 2

Step 5: seat ORU on platform

Step 6: unlatch gripper from handle

Step 7: goto park position

Figure 2. ORU Task Steps

Step 1: goto point above oru door handle

force in x < 0 lbs
force in y < 0 lbs
force in z < 0 lbs
torque about x < 0 in-lbs
torque about y < 0 in-lbs
torque about z < 0 in-lbs
gripper unlatched
platform switch 1,2,3 connected
platform switch 4,5,6 disconnected

Step 2: seat gripper on door handle

force in -5 < x < 5 lbs
force in -5 < y < 5 lbs
force in - 15 < z < 15 lbs
torque about -1 < x < 1 in-lbs
torque about -1 < y < 1 in-lbs
torque about -10 < z < 10 in-lbs
gripper latched
platform switch 1,2,3 connected
platform switch 4,5,6 disconnected

Figure 3. Operational Limits for ORU Replacement Task

Nexpert's Representation of the operational limits

```
Rule:  Rule 33
If
        step.number is precisely equal to 1.00
        And there is evidence of assign_limit_values
Then limit_values_assigned
        is confirmed.
        And task_gripper.status is set to unlatched
        And -5 is assigned to task_lower_force_limits.x
        And 5 is assigned to task_lower_force_limits.y
        And 10 is assigned to task_lower_force_limits.z
        And -70 is assigned to task_lower_torque_limits.x
        And -10 is assigned to task_lower_torque_limits.y
        And -10 is assigned to task_lower_torque_limits.z
        And 20 is assigned to task_upper_force_limits.x
        And 30 is assigned to task_upper_force_limits.y
        And 40 is assigned to task_upper_force_limits.z
        And -10 is assigned to task_upper_torque_limits.x
        And 50 is assigned to task_upper_torque_limits.y
        And 50 is assigned to task_upper_torque_limits.z
        And task_oru_position.position1_status is set to connected
        And task_oru_position.position2_status is set to disconnected
        And assign_limit_values is set to FALSE
        And 17 is assigned to message_num.number
        And Execute dectalk_male(@ATOMID=message_num.number;)
```

Nexpert's Representation of decision process

```
Rule : Rule 16
If
        there is no evidence of force_in_z_approaching_upper_limit
        And task_upper_force_limits.z_forces.z is less than 0.0
Then indicate_unsafe
        is confirmed.
        And force_in_z_approaching_upper_limits is set to TRUE
        And 5 is assigned to message_num.number
```

Figure 4. Nexpert Representations

There are three safety issues that need to be addressed pertaining to telerobotic operations. How should an unsafe condition be detected? After it is detected, what action should be taken to respond to this unsafe condition? What should be done to recover safely from this unsafe condition? In the PSS implementation of the ORU demonstration, we chose to use the expert system shell, NEXPERT, to detect an unsafe condition based on the operational limits. NEXPERT is an object oriented, rule based expert system shell which allows one to represent knowledge in a rule format and reason about this knowledge to solve a problem. We chose to use NEXPERT for three reasons. First, after evaluating several other expert shells such as Clips and KEE, NEXPERT was

the best expert system shell for the money. Second, the safety problem lent itself to the rule format, for example, if force in x is greater that 10 pounds then notify operator of unsafe condition. Finally, NEXPERT is object oriented which means that it performs operations on objects depending upon the state of the world. Each control cycle, NEXPERT evaluates only those rules which contain objects which pertain to the state of the world. For example, if the object force_x reaches its operational limits then the rules which contain that object will be evaluated. This differs from procedural languages such as C and Pascal because procedural languages perform operations sequentially as they exist in a procedural language program. NEXPERT receives sensor data and commands and compares these data and commands to its knowledge of the operational limits of the step of the task to determine if an unsafe condition is present.
NEXPERT'S user interface was used the to load the rules into NEXPERT's knowledge base. Figure 4 is an example of the output from NEXPERT's knowledge base after the rules were entered. Rule 33 provides an example of a rule which assigns the operational limits of a step of the task. If the step which involves grabbing the ORU handle is being executed then the hypothesis assign_limit_values becomes true and the operational limits for that step one assigned. Rule 16 provides an example of how NEXPERT determines if sensor data or operator interface commands lie outside of the operational limits of the step of the task. If the force is z is greater than the operational limit for that step of the task then a message number is assigned which will be reported to the operator.

The next issue that needs to be addressed in the area of procedural safety for telerobotic operations is once an unsafe condition is detected, what action should be taken. Currently, in the ORU demonstration, if the PSS detects an unsafe condition it notifies the operator both visually and audibly. Messages are printed to the terminal in the workstation to indicate to the operator which unsafe condition is present. Messages are also sent to the voice synthesizer, Dectalk, which conveys the unsafe message audibly to the operator. Thus, we provide two way of communicating to the operator that an unsafe condition is present. This is necessary since the operator may, for example, be watching the camera monitors instead of the workstation terminal. If and unsafe condition arises and the only form of communication to the user is messages to the workstation terminal, the operator is not going to receive the warning. If the unsafe condition is serious enough that the task should not proceed then the robot motion should stop and the operator should be notified.

Once the unsafe condition is detected and an action is taken, how should the Procedural Safety System recover from the unsafe condition? The PSS that was implemented in the ORU demonstration recovers from an unsafe condition by returning control to the operator to correct the problem. The operator then continues the task at the current step while the PSS continues to monitor the sensor data and commands. Figure 5 summarizes the functions of the PSS.

The Results

The Procedural Safety System that has been implemented in the robotics laboratory has enabled us to begin to look at safety for telerobotic operations. The PSS has made the ORU demonstration easier and much safer to operate. Prior to the implementation of the PSS, the burden of monitoring the sensor data and operator commands was placed on the operator. The PSS relieves much of this burden by monitoring the sensor data and commands from the operator to ensure that the task is operating safely, enabling the operator to concentrate on performing the task itself.

Future Work

There is a great deal of safety related work that needs to be researched and implemented in the robotics laboratory. The effectiveness of the PSS that exists in our laboratory depends upon the amount of sensing capability. At the present time, the robot system is limited by the amount of

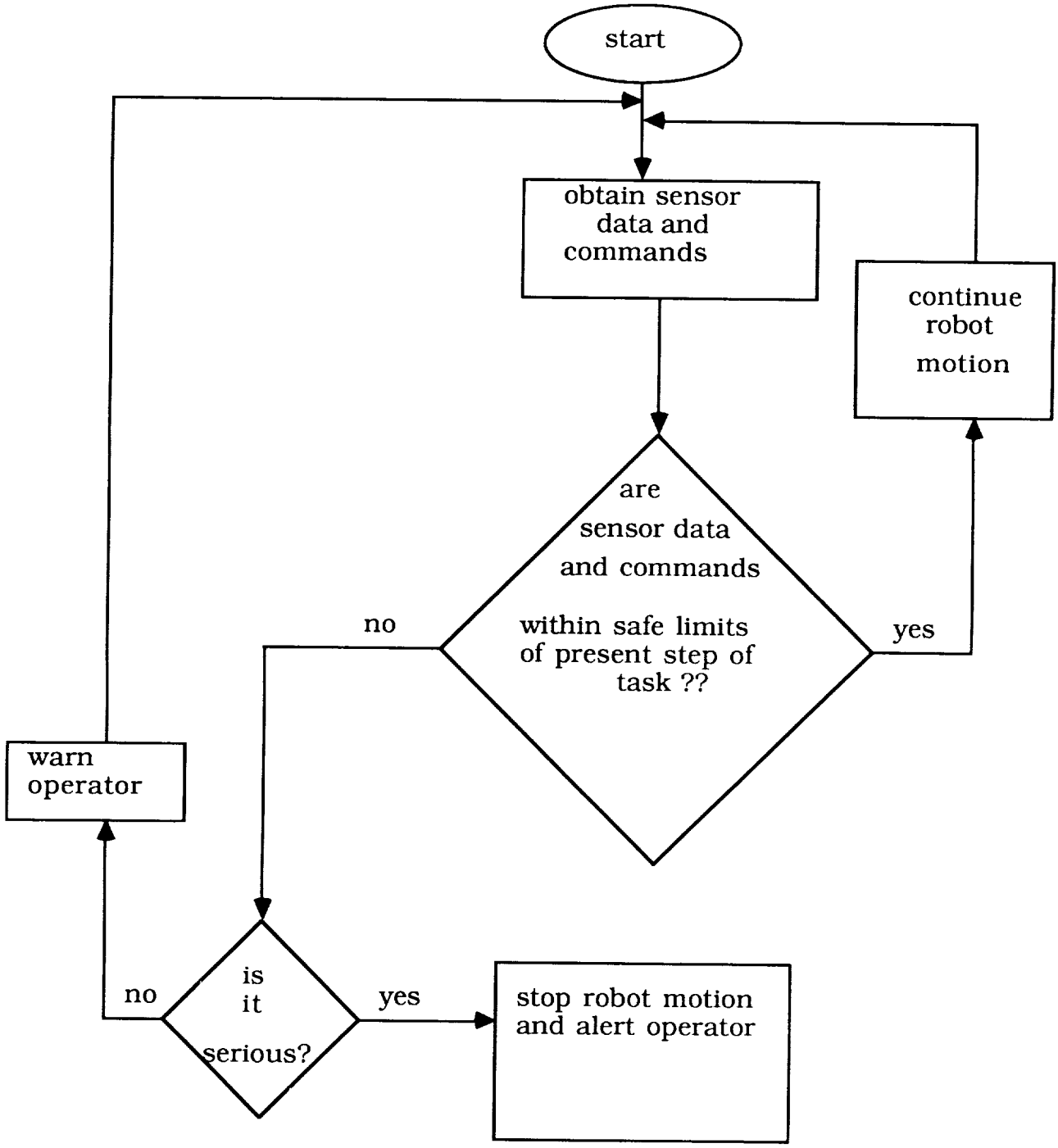

Figure 5. Procedural Safety System's Flowchart

sensing capabilities since its only sources of sensor data come from a force torque sensor, the gripper, microswitches and encoder values of the robot. Several other sensing capabilities need to be incorporated. Laser ranging provides 3-D information about an object without having to touch the object. This sensing technique is necessary for object recognition. Tactile sensing provides information about an object between the gripper fingers which cannot be obtained using a force torque sensor because a force torque sensor only provides sensor information at the wrist of the robot. Proximity sensing is another sensing capability that needs to be added to the robot system. It provides quick sensing data indicating the presence of an object which can used for object avoidance algorithms. Besides incorporating addition sensing capabilities into the lab, world modeling techniques need to be researched and implemented which provide a way to organize and represent sensor data so that the PSS can quickly and efficiently acquire the data.

Acknowledgements

The author would like to acknowledge Ms. Susan Cofer who conceived the Procedural Safety System and was instrumental in its development.

Robotics/Intelligent Control