

# Status Driven Teleoperation System

## A new paradigm and an application to the Microworld

Stephen Palm    Hideki Murayama\*    Taketoshi Mori    Tomomasa Sato

Email: {palm, murayama, tmori, tomo } @lssl.rcast.u-tokyo.ac.jp

RCAST, The University of Tokyo  
4-6-1, Komaba, Meguro-ku, Tokyo 153, JAPAN

\* Engineering Systems Laboratory, Aoyama Gakuin University  
6-16-4 Chitose-dai, Setagaya-ku, Tokyo 157, JAPAN

### Abstract

*This paper proposes a third generation teleoperation system based on a status driven slave symbolically instructed by the master control panel. Furthering the first generation joint-angle control techniques and second generation coordinate transformation techniques, this system recognizes the target status specified by the operator by extracting the task status from visual sensors. The task environment is initially described via sensing points in the work environment and on the manipulated object. The relationships between the sensing points and the change of those relationships describes the task at hand. Operation can proceed in automatic mode where the system completely directs the slave or in shared mode where the system assists the operator by constraining the slave motion while the operator directs via a visual communication interface. Although Status Driven techniques offer several improvements in teleoperated control, this work explores the robustness and reliability provided in dynamic workspace environments. One successful application of this technique is in micrometer scale part handling, the so-called "microworld assembly" where an operator's past experience in the macro world is not applicable to the physics experienced in the microworld.*

### 1. Introduction

Teleoperated robots are indispensable in environments where humans cannot perform direct manipulation. In dangerous or distant locations or in the microworld, humans must manipulate the environment through a remotely controlled mechanism. Although teleoperation techniques have been extensively researched and developed, operators still experience problems in accomplishing tasks when working through machines. Improving the teleoperating worker's situation is our underlying theme.

From the master slave point of view, we survey the traditional research in teleoperation. First generation "joint-angle correspondence" techniques and second generation "general coordinate" techniques will be briefly introduced, followed by an introduction of the proposed "status driven" technique.

The first generation of master slave teleoperated control can be categorically grouped under "joint-angle correspon-

dence". The master arm joint angle(s) and the corresponding slave arm joint angle(s) are united under this control method. [1] The principle merit of this system is that very good response times can be achieved in well designed systems. The primary demerits of this method is that the master and slave arms must be of identical mechanical configurations and autonomous control by the robot is not possible.

The second generation of master slave teleoperated control can be categorically grouped as "generalized coordinate master slave control method". The method can be summarized as: the slave hand coordinate system and the master hand coordinate system are made to coincide.

Merits of this method include: 1) the master and slave can have different mechanical configurations; 2) the master arm can be optimized to interface with the human operator and the slave arm can be optimized for the work environment; and 3) various automatic control methods can be used. [2] [3]

There are several demerits to this methodology. First, long delays in the control signals can result from calculating the computationally expensive forward and inverse coordinate transformations. This is still an issue even with recent advancements in digital signal processors. [4] Second, the system cannot adapt to dynamic environments since it does not have a concept of the task at hand nor the requirements for executing that task. In a dynamic environment, the operator himself must manipulate the object to compensate for the changes in the environment. [5] For example, when moving a liquid filled open container in a shifting environment, the operator must explicitly move the manipulator to track the changing direction of gravity so that the liquid would not spill out. Third, if external environmental sensing information is to be introduced into the system, it must first be translated into coordinate system information. The operator cannot easily use the external information to help him exercise control.

Status Driven techniques for teleoperation systems are proposed to cope with the problems of existing master slave control methods. This third generation control method can be summarized: the status which expresses the slave situation is united with the status expressed by the master situation. The status based slave manipulations are

derived from the sensor signals from the slave environment.

There are two overall merits to the system. First, the system can adaptively respond to a dynamic environment because the changes in the target circumstance or workspace situation can be detected with visual sensors. Second, not only can the system use external sensor information, the sensor input can be directly integrated into the slave situation. Thus, the external information does not need to be translated to the coordinate system in order to be utilized as in the generalized coordinate method.

Although the Status Driven teleoperation technique can be viewed primarily as a control methodology, it also contains some elements of task planning. Whereas methods such as Assembly Plan from Observation (APO)[6] and teaching-by-showing focus on the path and manipulator trajectory, the Status Driven technique ensures task completion and recognition of task completion in the slave work environment. Further, the master in Status Driven teleoperation is able to instruct the slave at the time of desired operation instead of a prior teaching session. Also, Status Driven techniques differ from teaching-by-showing methods since the operator does not manipulate the actual object nor a representation of the object through the control series procedure.

The remainder of the paper is organized as follows. Section 2 describes the Status Driven teleoperating method using two typical manipulation examples. The experimental system, Status Driven Microhanding system, is described in section 3. Results of the microworld teleoperation experiments are presented in section 4. Section 5 summarizes the work and offers possibilities for future work.

## 2. Status Driven control method

The main components and information flow of the Status Driven master slave control method are illustrated in Figure 1(b). The operator interfaces with a computer screen instead of a traditional master manipulator. The slave manipulator itself can be any conventional manipulation arm. The implementation of a Status Driven control method requires and produces different information than conventional teleoperation control methods (e.g. Figure 1(a)). In particular, data from sensors in the slave environment is essential to the system control and the task specification by the operator. Section 2.1 introduces the Status Driven control components and Section 2.3 describes the system information flow characteristics

To aid the explanation of Status Driven techniques, two familiar teleoperation control examples are interjected in Section 2.2. Each example illustrates a different Status Driven control style: 1) system controlled movement and 2) operator and system shared control movement. In the “placement” example, the designated target object is auto-

matically placed on a designated surface in the work environment. The “movement” example is an operator and system shared control movement; the operator directs the object movement and the system constrains the movement so that the object does not enter a specified area in the work environment. In both scenarios, the object is assumed to be already gripped by the slave manipulator, however, Status Driven techniques are also applicable to grasping of objects.

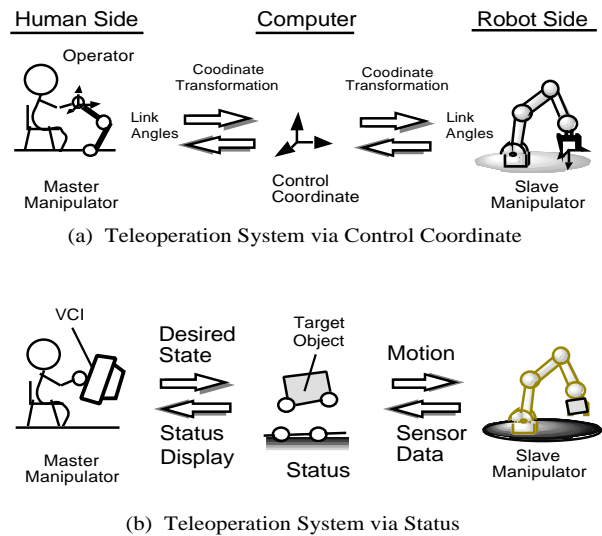


Figure 1. Teleoperation System Comparison

### 2.1 Paradigm Components

The Status Driven control method relies on sensor information from the slave environment. The integration of sensor information into the control algorithm is accomplished through *sensing points*. Sensing points are used to describe pertinent features in the workspace. In general, there are three types of sensing points: target, environmental, and milestone. Target sensing points are associated with the target object that is to be manipulated. Once designated, the target sensing points remain fixed to the object as the object is moved throughout the task. Likewise, environmental sensing points are associated with pertinent features in the manipulation environment. Environmental sensing points are similar to target sensing points in that they also remained fixed to the environmental location they were assigned. If the environment moves with respect to the viewing frame of reference, the sensing points would be made to track that movement. Environmental points differ from target sensing points in that the environmental sensing points are also the goal states to which the target sensing points are being made to coincide during automatic mode. That is the system controls the manipulator in such a way as to cause the target

sensing point(s) and environment sensing point(s) to coincide.

The third type of sensing point, milestone, is useful in system-operator shared control scenarios. In shared scenarios, the target object is still specified with target sensing points and there may also be environment sensing points, but a milestone sensing point designates an intermediate goal state to which the target sensing point will try to coincide. Depending on constraints imposed by the environment sensing points, the target object may not be able to coincide with the milestone sensing point. The system will attempt to make the target sensing point be as close to the milestone point as possible, but if the operator has inaccurately selected a milestone sensing point that would cause the target sensing points to cross designated environment constraints, the system will prevent the full completion of movement to the milestone point. The milestone sensing point could be considered as a specialized case of an environment sensing point in which that “environment” sensing point would be operator controlled during the task execution. For clarity, we give this special type of sensing point a separate designation since it is under operator control during the task execution phase.

In general, the occurrence of sensing points in a Status driven task can be divided into two phases: task teaching and task execution. Control of a target object is initiated by associating sensing points to it. This process of association of points is referred to as the teaching phase and is generally only performed once in a control situation at the beginning of the session. The operator interfaces with a video image of the work environment where the sensing points will be superimposed as they are being selected. The visual image, with possible status and query text, and the sensing point input method and superimposed symbolic display constitute the so-called Visual Communication Interface (VCI). An example VCI (pronounced VIK’ee) is described in Section 3.

During the teaching phase, appropriate locations on the target and the environment (i.e., work area) are selected and identified symbolically by the operator using the VCI. These sensing points form the basis of the control mechanism because visual detection techniques are used to extract the condition with respect to those points. Depending on the task at hand, the points may be grouped to indicate boundaries or lines of reference.

Once the task has been described to the system via the specification of the sensing point during the teaching phase, task execution begins. In automatic mode, the system attempts to make the corresponding target and environment sensing points coincide. In operator and system shared mode, the system causes the target sensing point to follow the operator’s specification of intermediate milestone points while movement may be constrained by environment sensing points.

## 2.2 Status Driven Teleoperation examples

The previously mentioned two teleoperation examples will now be described in terms of how they would be performed by the status driven control method.

Consider the following example in two dimensional space where the system completely controls the motion after the teaching phase. The teaching phase requires specifying the type of task and the pertinent attributes of the target object and slave environment by specifying sensing points. Two sensing points would be designated with respect to the target in the image to indicate the object’s edge that will be placed on a surface. Two additional sensing points would be designated with respect to the work environment to indicate a surface on to which objects may be placed or an area where objects must not be allowed to enter.

After the teaching phase, the system automatically controls the manipulator. As the target begins to move, the visual sensors track the movement so that the target’s sensing points remains fixed with respect to the target object. Movement continues until the target sensing points coincide with the environment sensing points. Once they coincide, the movement stops since the task has been completed.

The second scenario uses shared mode control where the operator and the system share control over the object motion. In the teaching phase, a sensing point placed on an object can also have a “milestone” sensing point which is initially placed coincident with the first sensing point. After the teaching phase, this milestone sensing point is relocatable by the operator and will be used to “attract” the target object sensing point to it. Thus when the operator moves the milestone sensing point using an input device such as a touchscreen, mouse, or joy-stick, the system will try to move the slave so that the sensing point associated with the slave (target) will again coincide with the milestone sensing point. The operator is not controlling the slave manipulator directly, instead he is setting (and possibly continually moving) the intermediate desired location (milestone) of the slave. Thus the target could be made to follow an arbitrary path in the workspace by a succession of goal locations indicated by the successive positions of the milestone sensing point.

Additionally, if environment description sensing points were included in the movement scenario, the target sensing point could be prevented from penetrating the segment joining the environmental sensing points even though the milestone sensing point were to penetrate the environment sensing segment. In that situation, the system would still try to move the target sensing point as close to the milestone sensing point as possible while not crossing the environment sensing points boundary.

## 2.3 Characteristics

The following components form the basis of the interface in a status driven teleoperation system.

**Input.** The primary input to the system is the symbolic description of the elements in the slave world via sensing points. The elements include the target object itself as well as relevant characteristics of the work environment. There are two aspects to the sensing points. First, the sensing points define the physical status of the target object and the work environment. This can be thought of as defining the shape or relevant surfaces of the object and the environment. Second, the relationship between the points, in particular, the relationship between the target sensing points and the environment sensing points defines the status of the task to be completed. When the corresponding points are made to coincide, the system has accomplished the task. Thus the definition of the relationship between the points plans the task.

Another aspect of the status driven teleoperation input system is that instructions of movement to the slave are operations on the sensing points. The manipulator itself is not being controlled by the master, the slave component of the system moves the manipulator in response to the desired status of the sensing points as described by the master. Movement of the manipulator is caused by the relative locations of the sensing points, and in operator-system shared controlled situation the operator guides the manipulator by moving a desired state (i.e., milestone) sensing point.

**Output.** The primary output of a status driven teleoperation system is movement of the slave which satisfies the desired status. Simply put, the output is the achievement of the desired task. In particular, the movement of the slave or the particular path of the target movement is not meaningful to the operator, the prime objective is having the task accomplished.

Consider the differences in the outputs of the three generations of master slave control systems. In joint angle methods, the output of the system is when the desired set of joint angles has been achieved in the slave. Similarly for generalized coordinate methods, the output is when the desired location in the coordinate workspace of slave's manipulator has been achieved. The output of a status driven system is when the desired work status has been achieved in the slave world.

**Control.** The essence of status driven techniques is sensing. Without sensing, control cannot be accomplished by a status driven system. The task completion status is the essential interest of the operator in a manipulation situation, thus task completion status should be the primary interest of a master slave control system. Sensing is the means that allows the system to recognize that the primary objective has been satisfied. Sensing points are

the communication medium between the master and the slave.

If we view master slave systems from the modeling viewpoint, second generation systems explicitly model the task merely as a path. Task failure can still occur in a dynamic environment even though the model path is followed precisely. On the other hand, a third generation status driven system models the completed task. If the operator wishes to also model the environment by describing a path, the operator-system shared form of the status driven method allows path modeling while still retaining task completion modeling.

It is important to note here that the automatic form of status driven control is not autonomous control by the system. The master must teach the task to be accomplished by specifying the relationship between the sensing points.

## 2.4 Merit

During the task execution phase, sensor based feedback is possible and allows task completion in dynamic environments. Automatic correction, assistance action, or work action can be produced corresponding to the sensed slave movement, change of work environment, or the change of work situation. Robustness is achieved since the slave is manipulating directly on sensed information in the slave environment and is not dependent on model information generated in the master. Thus the control actions conform to the actual work situation.

The fundamental approach of the system, direct usage of sensor output, also becomes one of the fundamental merits. The type of information most necessary for the system to use is also the easiest for human operators to process. The visual display and feedback combined with the symbolic display and manipulation of the sensing points is a very intuitive way to specifying the algorithm of setting and using the sensor information in the system. In other words, the visual control interface is a better method of "programming" than entry of typed text to control the slave.

Finally, specification of the task via sensing point relationships alleviates the need for the system to automatically analyze the task and environment to determine the final destination and the path to achieve it. Since human operators tend to be rather good at almost unconsciously planning tasks and recognizing environmental constraints, the Visual Communication Interface allows rapid and precise specification of the task to the system. The system can then execute the necessary movements with the manipulator in the slave world because it understands the goal state and is directly monitoring and reacting to the dynamic environment through sensor information.

### 3. Status Driven Micro Handling System Description

The first implementation of a status driven teleoperation system is the Status Driven Micro Handling System (SD-MHS). This system provides the operator several methods for manipulating objects on a micrometer scale. The hardware, software, and interface will be described in this section.

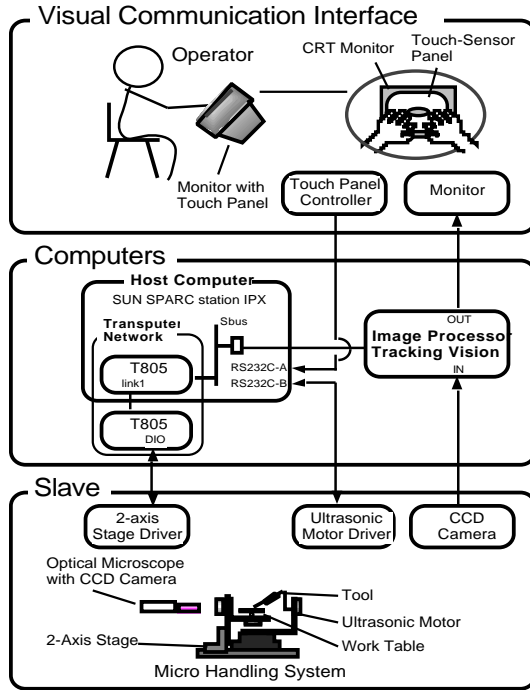


Figure 2. Status Driven MHS Block Diagram

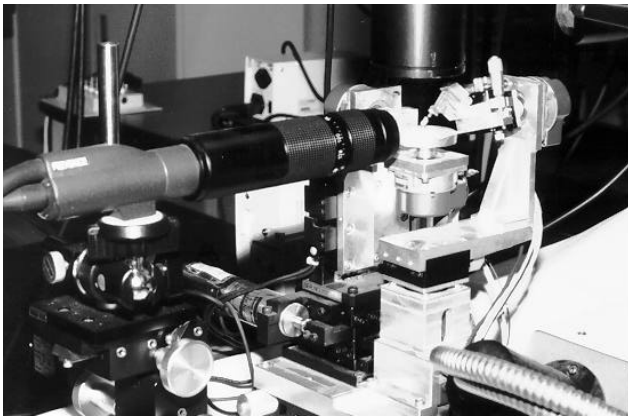


Figure 3. Micro Handling System I (MHS-1)

#### 3.1 System and Interface

A schematic diagram of the overall system is shown in Figure 2. The slave portion utilizes the Micro Handling System I (MHS-I) described in [7] and shown in Figure 3. For this system implementation, MHS-I views and con-

trols the slave environment in two dimensions with three degrees of freedom through a two-axis work table with 17.4nm resolution and a tool that can be rotated with 0.1° resolution. The operator communicates to the master through the hardware components of the Visual Communication Interface (VCI) which consist of a video monitor with an integrated touch panel. The video monitor continuously displays a view of the slave work area obtained from an optical microscope and CCD camera mounted close to the work area. System command and control software is implemented in C and LISP running on a Sun workstation, two Transputers, and a Fujitsu image tracking hardware.

When operating the system, the operator teaches the sensing points and can direct the slave motion by appropriate finger movements on the touch screen. In addition to the image of the slave work environment, computer generated symbols, text, and graphics are overlaid on the monitor. They are used to prompt the operator during the teaching phase and symbolically display the location and movement of the sensing points as shown in Figure 4.

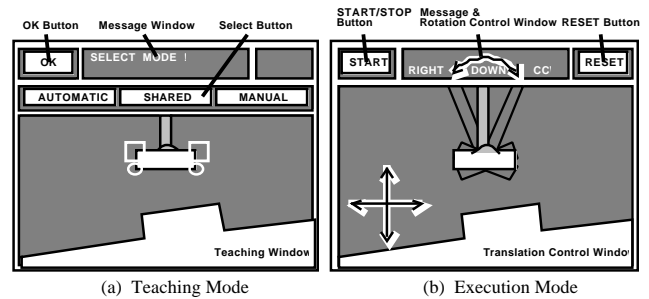


Figure 4. Visual Communication Interface of SD-MHS

#### 3.2 Sensing points and tracking

The location of the sensing points should be selected by the operator based on two criteria. First, each sensing point should be placed on a relevant physical attribute of the target or environment objects. For example, sensing points could be placed on edges or vertices. Second, the sensing points in the target object and the environment will eventually be made to coincide so the sensing points should be located appropriately indicating the final desired state of the target with respect to their environment.

After an object or environment sensing point is specified by the operator, the system must maintain the sensing point in the same relative position on the physical element even though it is moving with respect to the video camera frame of reference. In this system implementation, that is accomplished through using vision tracking hardware. The tracking hardware uses template matching techniques to continuously update the location of the center in each of the several tracking windows in the image. The

operator selects the initial center of each tracking window. Geometric methods are then used to keep the sensing points in the appropriate location with respect to the tracking window. Examples of typical placement in a microworld manipulation are shown in Figure 5.



**Figure 5. Example of object and environment sensing points (indicated by circles) and the corresponding tracking windows (window centers are represented by squares).**

### 3.3 Task Functions

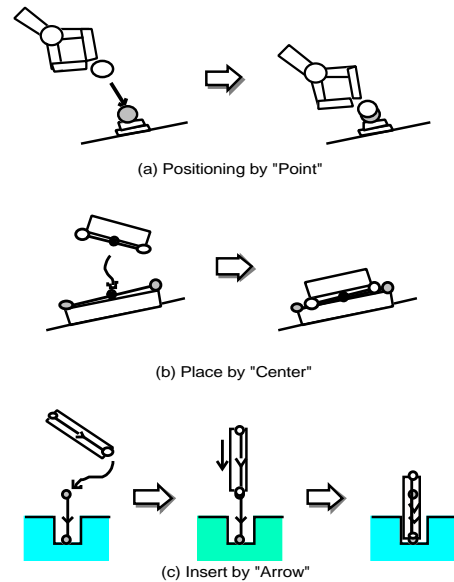
Although the use of status driven techniques are applicable to a wide range of assembly tasks, in this paper we concentrate on a few exemplary tasks. The motions are divided into two basic control modes: automatic and shared. In the automatic mode, the system completes all of the movement from the initial to final state. In the shared control mode, the human operator directs the target object motion but the system assists the human by constraining the motion in a manner beneficial to the specified task.

In both the automatic and shared modes, several task movement functions have been implemented. These correspond to the relationship between the initial and final locations of the sensing points. When the operator is teaching the sensing points to the system, one of the task functions is also selected so that system knows the desired final state of the sensing points. Different task functions require differing numbers of sensing points.

**Automatic Task Mode.** In the fully automatic mode, three task functions have been implemented: “point”, “center”, and “arrow” which progressively have more constraints on the movement from initial to final state. These three functions superimposed on tasks in the slave world are illustrated in Figure 6. For more complicated assembly tasks, these functions can be combined or further extended functions could also be implemented.

The “point” function operates on two points, moving one point so that it coincides with another point. In general, the initial position of the moving point would represent a sensing point on a target object in it’s initial state.

This function can be executed with purely translational movement. (See Figure 6(a)).



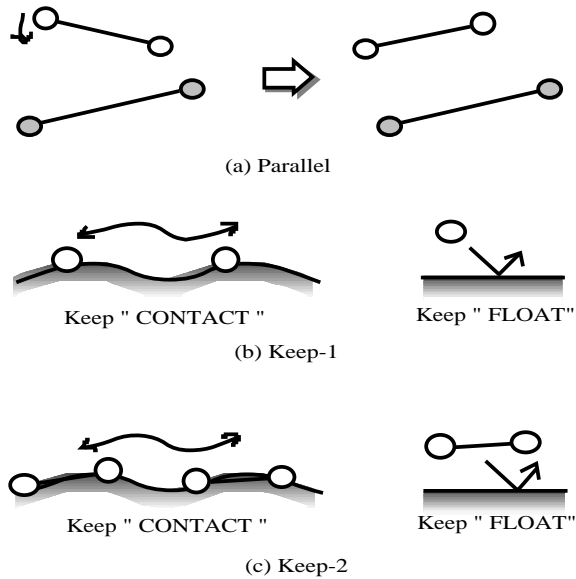
**Figure 6. Automatic Mode Task Functions**

The “center” function is similar to the “point” function but operates on two pairs of sensing points. One pair of points is associated with the target and the other pair is associated with the environment. Each of the two points on an object forms a line segment that can be thought of as a surface in the slave world. In order for the target and environment points to coincide, the two line segments must be also be coincident. Manipulating one of the lines to be parallel with the other may involve rotational movement in conjunction with the translational movement to make the points coincident. This function actually uses 6 points (see Figure 6(b)), four explicit and two implicit points. The explicit points (white or gray circles) are the end points of the line segments. The implicit points (black circles) are the midway locations on the segments which are actually made to be centered. The function is applicable for placement tasks.

The “arrow” function also operates on two line segments and utilizes both rotational and translation movement. In this function, the two segments are also made to be coincident, but one set of endpoints (the “heads” of the “arrows”) is required to be coincident instead of the midpoints. (See Figure 6(c).) An additional constraint in the movement control is that the head of the moving arrow must be coincident with the tail of the stationary arrow and the arrow coincident before the final translational movement along both arrow’s axis to place the heads coincidentally. This function is useful in tasks involving insertion.

**Shared Task Mode.** Three task functions for the operator-system shared mode of control have also been

implemented. These task functions constrain the sensing points (and hence the objects in the slave world) as the operator sets new goal sensing points or the environmental sensing points are moved relative to the viewing frame of reference. The functions are shown in Figure 7.



**Figure 7. Shared Mode Task Functions**

The “parallel” function rotates the slave sensing points so that the respective segments joined by the sensing points in the target and the environment are made to be parallel. This correction feature occurs whether the target or the environment moves with respect to the camera frame of reference. This correction feature also occurs as the operator is directing the target motion by setting new milestone sensing points.

The “keep-1” function is used to keep the target a specified distance from an uneven surface as it travels along the surface. The specified distance could be zero indicating the target should maintain contact with the environment surface. If a nonzero distance is desired, a so-called float situation, the sensing points can be offset from the surface by an appropriate distance from the tracking window centers. In all cases, the environmental reference points are determined through edge detection by the binarization method.

The “keep-2” function is an extension of the “keep-1” function where two target sensing points maintain a fixed distance from the environment surface. The two target sensing points are also constrained to keep a fixed distance between themselves. This function is useful for keeping a target tangential to the local surface as it travels along the surface.

**Motion Method.** Rotational and translation motions each have two speeds which we term “Fast” and “Slow”. Selection of the translation speed is dependent on

the distance between the sensing points. Selection of the rotation speed is based on the angle between the pairs of sensing points. The movement thresholds are shown in Table 1 and Table 2. The movements are executed five to ten times per second depending on processing time. The movements are assumed to be slow enough to ignore dynamic effects.

**Table 1. Translational Movement Thresholds**

	step output (pixels)	difference (pixels)
Stop	0	0 $x < 2$
Slow	1	2 $x < 20$
Fast	3	20 $x$

**Table 2. Rotational Movement Thresholds**

	step output (degrees)	difference (degrees)
Stop	0	0 $< 3$
Slow	2	3 $< 10$
Fast	5	10

## 4. Experiments

Two sets of experiments were performed to illustrate the merits of status driven teleoperation methods. The first set of experiments are an operator-system shared control example that demonstrates the ease of control provided by the visual communication interface. The second set of experiments demonstrates the robustness of the automatic mode.

### 4.1 Surface following

Using conventional control methods in the micro world, it is rather difficult to move an object along a surface and to maintain manipulator direction orientation at the same time. The “surface following” task is where the operator wishes to maintain a fixed distance and orientation from the surface as the object is being moved by the operator along the surface. Using a status driven shared control mode, the operator can control the speed of travel of the object while the system maintains the target’s distance and orientation with respect to the environment surface. The sequence in Figure 8 shows the operator wishing to move the target across a “hilly” surface.

In Figure 8(a), the operator completes the teaching phase by setting the mode, sensing points, and tracking reference windows. In Figure 8(b), the execution phase begins with the system determining the distances from the environment to the target references. The operator now begins motion along the horizontal axis by setting a new milestone sensing point on the touch panel. The system responds by moving the target to try to make the target sensing point coincide with the milestone sensing point. As movement proceeds, one of the sensing points detects

that its trajectory would increase the object-environment separation so corrective movement is taken. Figure 8(c) shows the right-hand sensing point changing from solid to outline to indicate that change in status. The right corner of the Message window displays “CW” in anticipation of a clockwise correction rotation. Figure 8(d) shows that the target orientation has been maintained while traversing the “hilly” terrain. Thus status driven teleoperation is successful in the surface following task.

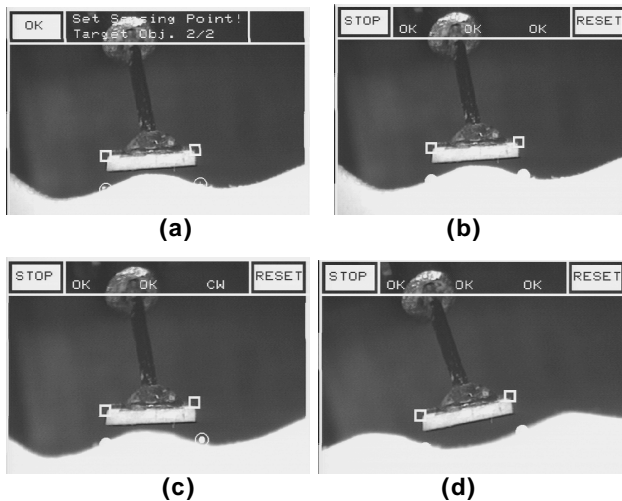


Figure 8. Surface following sequence (see text)

#### 4.2 Manual vs. status task execution time

Experiments were conducted to compare “Status Driven” operation versus “Manual” operation. For each task, two phases were required. In the first phase, termed “teaching”, the operator must configure the system for the desired class of movement. In either the Status Driven or Manual case, the type of movement such as “place” or “insert” is input to the system through a text box selection on the touch panel. In the second phase, the actual movement is performed. In the case of Status Driven control, the system controls and performs the movement. In manual control, the human manipulates the object by touching and dragging his finger in the direction of the desired motion.

The results of the first experiment with ten subjects were averaged and shown in Table 3. The results clearly show that the moving sub-task is performed more quickly when under Status Driven control. Unfortunately, the current Status Driven setup user interface requires considerable time to specify all of the necessary sensing points.

It is also important to note in the last line of Table 3 that the automatic mode of SD-MHS ensured the completion of the task. One of the operators during the insertion task trial was unable to complete the task before ramming the “peg” into the work environment.

Table 3. Comparison of Status Driven vs. Manual Operation for Placement and Insertion

	Placement		Insertion	
	SD	Manual	SD	Manual
Teaching Time (s)	43.1	6.0	40.1	5.2
Execution Time (s)	21.4	32.0	27.3	38.7
Total Time (s)	64.5	38.0	67.4	43.9
Completion (%)	100	100	100	90

## 5. Conclusions

The status driven teleoperation control method has been introduced and realized through the Status Driven Micro Handling System (SD-MHS). Direct utilization of sensor information via sensing points allows robust manipulation even in dynamic environments. Experiments have shown the effectiveness of the approach in both automatic and shared control modes in microworld manipulation tasks. Obviously, reducing the time of the teaching phase would further improve the current implementation. Work is currently underway to streamline the input of the sensing points and task.

In the initial system implementation described in this paper, the sensing points are input manually by the operator, but a suitable system could be devised where the critical sensing points of the work environment and the initial target state are extracted automatically. Likewise, the tracking reference windows could also be automatically assigned after the sensing points had been selected. We envision future work to automatically assign sensing points through analysis of operator behavior to extract task intention.

## References

- [1] K. S. Fu, R.C. Gonzalez, and C.S.G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill 1987.
- [2] *Robotics Handbook*, Rob. Soc. of Japan, 1990, pp594-9.
- [3] G. Andre and R. Fournier, Generalized End Effector Control in a Computer Aided Teleoperation System with Application to Motion Coordination of a Manipulator Arm on an Oscillating Carrier, ICAR 85.
- [4] S. Lee and H.S. Lee, Modelling, Design, and Evaluation of Advanced Teleoperator Control Systems with Short Time Delay, IEEE Trans. on Robotics & Automation, Vol 9, No 5, Oct. 93, pp607-23.
- [5] T. Sheridan, Space Teleoperation Through Time Delay: Review and Prognosis, IEEE Trans. on Robotics & Automation, Vol 9, No 5, Oct. 93, pp592-607.
- [6] K. Ikeuchi and T. Suehiro, Towards an Assembly Plan from Observation, part I: Assembly Task Recognition with Polyhedral Objects. IEE Transactions on Robotics and Automation, vol 10, no. 3, 1994, pages 368-385.
- [7] T. Sato, K. Koyano, M. Nakao, and Y. Hatamura. Novel manipulator for micro object handling interface between micro and human worlds. *Proc.s IEEE/RSJ Intl Conf. on Intelligent Rob.s and Sys*, Vol 3, pp 1674-1681, July 1993.