

NIST
PUBLICATIONS

A11103 004045

NAT'L INST OF STANDARDS & TECH R.I.C.



A11103004045

Fiala, John C/Interfaces to teleoperatio
QC100 .U5753 NO.1254 1988 V198 C.1 NIST-

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(formerly National Bureau of Standards)

NIST Technical Note 1254

Interfaces to Teleoperation Devices

John C. Fiala

QC
100
.U5753
#1254
1988
C.2

**NATIONAL INSTITUTE OF STANDARDS &
TECHNOLOGY**
Research Information Center
Gaithersburg, MD 20899

NISTC
20100
.45753
no. 1254
1988
C.2

NIST Technical Note 1254

Interfaces to Teleoperation Devices

John C. Fiala

Intelligent Controls Group
Robot Systems Division
Center for Manufacturing Engineering
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899

Issued October 1988



NOTE: As of 23 August 1988, the National Bureau of Standards (NBS) became the National Institute of Standards and Technology (NIST) when President Reagan signed into law the Omnibus Trade and Competitiveness Act.

U.S. Department of Commerce
C. William Verity, Secretary

National Institute of Standards and Technology
(formerly National Bureau of Standards)
Ernest Ambler, Director

National Institute of Standards
and Technology
Technical Note 1254
14 pages (Oct. 1988)

U.S. Government Printing Office
Washington: 1988

For sale by the Superintendent
of Documents
U.S. Government Printing Office
Washington, DC 20402

Contents

1. Introduction	1
2. Joint-Space Teleoperation Interfaces	4
3. Cartesian Teleoperation Interfaces	7
4. Conclusion	9
5. References	9

Interfaces to Teleoperation Devices

1. Introduction

This document describes a basic logical architecture for teleoperation control devices and interfaces for integrating these devices with a telerobot control system architecture. The interfaces described are for manipulator control only. They do not consider articulated end-effector control, although an interface for this can be viewed as a special case of the interfaces described here. Most end-effectors currently available have only one motion to open or close the fingers. Thus they require only one data element in the interface to specify finger position. This could easily be added to the described interfaces.

In the general case, a teleoperation device may be actively controlled. This is usually done to provide what has come to be called "force feedback" to the human operator. Although force may not be the feedback variable, the idea is that the active control of the teleoperation device based on the state of the telerobot (so-called "kinesthetic coupling"), can provide the operator with more information on the progress of the telemanipulation task. Many devices, such as simple joysticks, are not actively controlled and provide no feedback to the operator. Devices like the Jet Propulsion Laboratory's Force-Reflecting Hand Controller [9] are specially designed to provide feedback. Such devices require active control. Thus, the general architecture for a teleoperation device is that of a control system as described in [1]. This type of architecture is depicted in figure 1. Here, the teleoperation de-

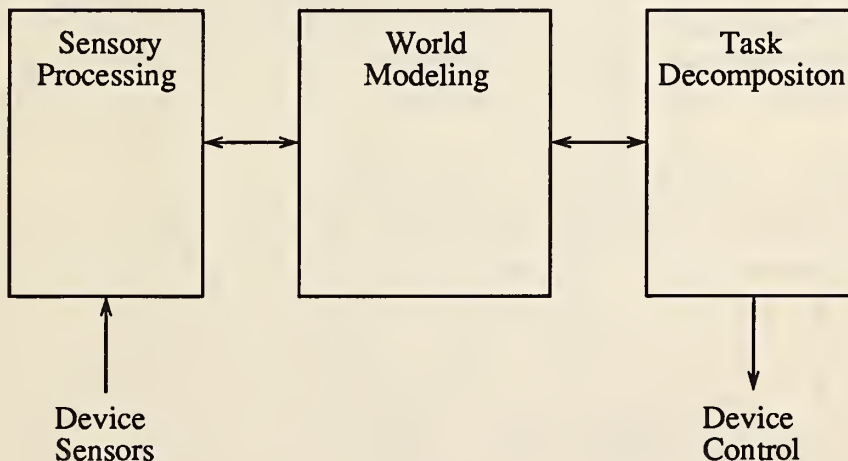


Figure 1. Teleoperation device control architecture.

vice control system is composed of three elements: Sensory Processing (SP), World Modeling (WM), and Task Decomposition (TD).

The Sensory Processing component of the control system is responsible for reading system sensors, and then filtering and integrating this information over space and time. The Task Decomposition component computes the required control outputs. The World Modeling component is between Sensory Processing and Task Decomposition. This module functions to maintain the system's model of both itself and the environment, obtaining new information from Sensory Processing and providing the latest estimates to Task Decomposition. One of the primary activities of World Modeling is to "maintain the global memory knowledge base, keeping it current" [1 p. 9]. The "global memory knowledge base" refers to a data system in which is stored "the system's best estimate of the state of the world, including both the external environment and the internal state of the [Sensory Processing, World Modeling, and Task Decomposition] modules. Data in global memory is available to all modules at all levels of the control system" [1 p. 14]. Thus, this "global data system" provides a communica-

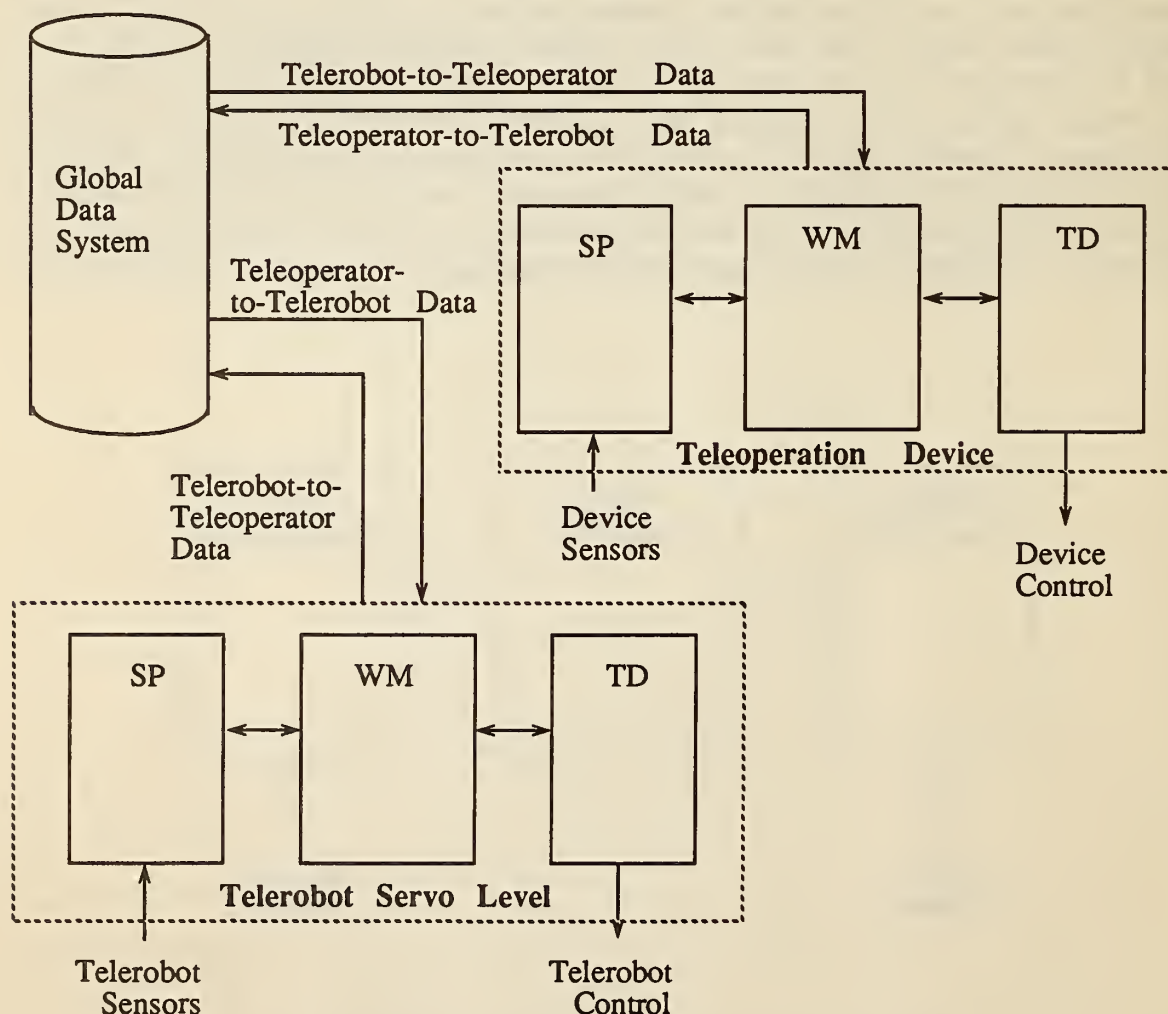


Figure 2. Logical architecture with teleoperation device.

tion mechanism for widely separated components of the control system. (See [1 pp. 25-26].) In particular, this mechanism serves to connect the Servo Level of the telerobot with the teleoperation device control level shown in figure 1.

Figure 2 depicts the interaction between the telerobot Servo Level and the teleoperation device control system. As shown in the figure, the teleoperation device control system updates the global data system as to the current state of the teleoperation device. This may include the teleoperation device's position, velocity, and sensed forces. When in "teleoperation mode" it is desired that the telerobot track the teleoperation device state. Thus, the telerobot World Modeling module obtains the state of the teleoperation device from the global data system and provides this information to Task Decomposition, where it is used in computing the control outputs. To simplify the discussion, this data will be referred to as the "teleoperator-to-telerobot" interface. Since the state of the telerobot may need to be reflected back to the operator via force feedback, there is another data path through the global data system for feedback. The data in this path shall be referred to as the "telerobot-to-teleoperator" interface.

In general, the teleoperation device controller is much simpler than the full telerobot control system, which includes four control levels. For this reason, it will tend to be easier to put all of the global data system in with the hardware of the telerobot control modules (but this is not required). If the teleoperation device controller and the telerobot controller use separate hardware systems, then the most reasonable place to put a hardware communication link (e.g., RS-422) is between the teleoperation device control system and the global data system. Thus, the telerobot system may need to have a simple process to receive the data and put it in the data system.

Note that, the data should be transferred via the data system and not directly between control modules. There are two principal reasons for this requirement. One reason is that a control module should not be forced to read data at all times, but the data should be available when the module needs it [10]. This is easily achieved when the data is written to a storage area, since a module can choose to read it or not depending on the algorithm currently in use. Secondly, the control information must be available to many system components, such as safety systems. The easiest way to give many components access to the information is to post it in a central location [10].

This discussion will consider teleoperation devices as divided into two classes. The first class consists of those devices which are kinematically similar to the telerobot or otherwise provide state information in the joint-space of the telerobot. These are the *joint-space teleoperation devices*. They include identical master-slave arm configurations and the so-called mini-master devices. Joysticks can operate in joint coordinates as well, albeit less conveniently. The second class of devices consists of all devices providing data in coordinate systems linked to some Cartesian frame of reference. These devices are the *Cartesian teleoperation devices*, which include the Force-Reflecting Hand Controller and Cartesian joysticks such as DFVLR's sensor ball device [4]. Obviously, it is possible to have Cartesian robots, in which case Cartesian teleoperation can be achieved through joint-space teleoperation devices.

The interfaces between the teleoperation device controller to the telerobot system Servo Level, as depicted in figure 2, are described in detail in the next two sections. Section 2 de-

Table 1. Joint-space teleoperator-to-telerobot data

<u>Data</u>	<u>Nature of Data</u>	<u># of Elements</u>	<u>Bits/Element</u>
θ_m	Joint positions given by device	# of joints	32
$\dot{\theta}_m$	Joint velocities given by device	# of joints	32
τ_m	Joint torques given by device	# of joints	32
Algorithm	Control mode from operator	1	32
K_τ, K_p, K_v, K_i	Control gains for telerobot	4 x # of joints	32

Table 2. Joint-space telerobot-to-teleoperator data

<u>Data</u>	<u>Nature of Data</u>	<u># of Elements</u>	<u>Bits/Element</u>
θ_s	Joint positions of telerobot	# of joints	32
$\dot{\theta}_s$	Joint velocities of telerobot	# of joints	32
τ_s	Joint torques of telerobot	# of joints	32
Status	Status of telerobot control	1	32

scribes the interfaces for joint-space teleoperation devices. The interface requirements for Cartesian teleoperation devices are detailed in section 3.

2. Joint-Space Teleoperation Interfaces

This section presents the interfaces for connecting joint-space teleoperation devices to the telerobot control hierarchy. Information from this type of device enters the control hierarchy at the Servo Level, where it is directly used in the servo control of the telerobot.

The teleoperator-to-telerobot interface for joint-space teleoperation devices is given in table 1. This data consists of information generated by the teleoperation device and sent to the telerobot control system. Note that the joint positions, velocities, and torques are obtained from sensors located on the teleoperation device. For example, joint positions might be output from encoders on individual joints of a master arm. Joint velocities could be from tachometers on the joints (or, possibly derived from position information by teleoperation device world modeling). The control gains (K 's) and the algorithm are input in some manner by the operator. The input could be through sensors located on the teleoperation device, e.g., switches or potentiometers, or from an operator terminal.

The data that passes from the telerobot controller to the teleoperation device is given in table 2. This information comes mainly from sensors located in the joints of the telerobot. Note that numeric data is 32 bits in both data paths. An argument could be made for 16-bit data elements since this is about the limit for current A/D converters. Also, integer formats may be more readily compatible with various type of hardware. However, 32-bit floating

point seems to be a convenient format for computation with modern floating point coprocessors.

In terms of units for these measured values, there appears to be no reason not to use metric SI units. This means that angular measurements are in radians and linear measurements are in meters. Although the output from sensors may be in various nonstandard units such as ticks or counts, the world modeling modules have models of the devices and can relate these values to standard SI units understandable by all systems.

The joint-space interfaces allow any of the four teleoperation control modes described by Vertut [2] and Thring [5]. The first mode is position-position control [2], which can be expressed in the following control equations.

$$u_{\text{telrobot}} = K_p(\theta_m - \theta_s) + K_v(\dot{\theta}_m - \dot{\theta}_s)$$

$$u_{\text{device}} = K_{pm}(\theta_s - \theta_m) + K_{vm}(\dot{\theta}_s - \dot{\theta}_m)$$

Here, the u 's represent control inputs to the motor drivers and the K 's are diagonal gain matrices multiplying the difference vectors. Note that the elements of the equations are available from the defined interfaces. K_{pm} and K_{vm} are not in the interfaces because they are set on the teleoperation device side of the system and do not need to be passed across.

To establish the second control mode, position-force control, the torque vectors must be used [2]. The equations are

$$u_{\text{telrobot}} = K_p(\theta_m - \theta_s) + K_v(\dot{\theta}_m - \dot{\theta}_s)$$

$$u_{\text{device}} = K_{tm}(\tau_s - \tau_m)$$

For the third control mode described in [2,3], called improved position-position control, the following control equations specify the behavior.

$$u_{\text{telrobot}} = K_\tau \{ [K_p(\theta_m - \theta_s) + K_v(\dot{\theta}_m - \dot{\theta}_s)] - \tau_s \}$$

$$u_{\text{device}} = K_{tm} \{ [K_{pm}(\theta_s - \theta_m) + K_{vm}(\dot{\theta}_s - \dot{\theta}_m)] - \tau_m \}$$

These control equations are intended to represent a position control loop between manipulators executing outside of local torque loops on each joint.

The final control mode is force-force control [5].

$$u_{\text{telrobot}} = K_\tau(\tau_m - \tau_s)$$

$$u_{\text{device}} = K_{tm}(\tau_s - \tau_m)$$

The feasibility of digital implementation of this control is somewhat in question [2,3,5]. The entire loop must be closed in about a millisecond to achieve stable behavior [3,8]. The position control modes may be performed at much slower rates, 5–20 ms, as described in [2,3,9].

Naturally, all of these bilateral control modes are only valid when the teleoperation device is actively controlled. When the teleoperation device is only capable of acting as a

sensor, then only the control equation on the telerobot is valid for a given control mode. In these cases, there is no need to pass information back to the teleoperation device, except maybe a status word. Thus, the minimal data required for joint-space teleoperation consists of Algorithm, θ_m , and Status. Although these might be sufficient to produce nominal teleoperation, the other parameters are needed for good performance and force-feedback.

The data rates required to support teleoperation using these interfaces are quite high. A 32-bit parallel data channel would seem appropriate. Even in the minimal 7 degree-of-freedom case, nine 32-bit words passed every 10 ms, a serial rate of 28.8 Kbaud is required, exceeding RS-232 capabilities [13]. For the general teleoperation case, 72 32-bit words passed every 5 ms, a serial data rate of approximately 0.5 Mbaud is required. Note that all values need not be passed every cycle. For instance, the K's and Algorithm parameters will only change occasionally, and thus can be passed less frequently than the other data. This can relieve some of the burden on the communication channel.

Table 3. Cartesian teleoperator-to-telerobot data

<u>Data</u>	<u>Nature of Data</u>	<u># of Elements</u>	<u>Bits/Element</u>
z_m	End-effector positions from device	7	32
\dot{z}_m	End-effector velocities from device	6	32
f_m	End-effector forces from device	6	32
Algorithm	Control mode from operator	1	32
C_z	Coordinate system specifier	25	32
$K_p, K_v, K_i,$ K_{pf}, K_{vf}, K_{if}	Control gains for telerobot	36	32
S, S'	Control selection matrices	36	32

Table 4. Cartesian telerobot-to-teleoperator data

<u>Data</u>	<u>Nature of Data</u>	<u># of Elements</u>	<u>Bits/Element</u>
z_s	Telerobot end-effector positions	7	32
\dot{z}_s	Telerobot end-effector velocities	6	32
f_s	Telerobot end-effector forces	6	32
Status	Status of telerobot control	1	32
θ_s	Telerobot joint positions	# of joints	32

3. Cartesian Teleoperation Interfaces

This section presents the interfaces for Cartesian teleoperation devices. These interfaces provide information to the telerobot Servo Level in the manner depicted in figure 2, just as the joint-space devices did. However, more information needs to be communicated for the complex algorithms available with Cartesian servo control of the telerobot.

Table 3 shows the information in the teleoperator-to-telerobot data path for Cartesian teleoperation. The vectors z_m and \dot{z}_m are the Cartesian positions and velocities specified by the teleoperation device. The Cartesian force vector f_m is a six-dimensional vector expressed in the same coordinate system as z_m and \dot{z}_m . The inputs z_m , \dot{z}_m and f_m are all derived from the movement of the joystick or hand-controller part of the teleoperation input. The remaining parameters of the interface are obtained from the operator either through switches or a terminal. The nature of these parameters is described in detail in [10].

The operator inputs the Algorithm parameter, which determines the control mode of the telerobot Servo Level. The parameter $C_z = \{ \text{coord. sys.}, T_w, T_e \}$ specifies the coordinate system in which the control is to be executed. This determines how the vectors z_m , \dot{z}_m , and f_m will be interpreted in terms of absolute end-effector motion. The K's are gain matrices which multiply the error terms in the control equation of the telerobot. The parameters S and S' select which degrees of freedom receive a specific control mode. This is used primarily for hybrid schemes such as hybrid position/force control and combined position/rate control.

The data returned to the teleoperation device controller from the telerobot Servo Level is given in table 4. The telerobot controller returns a Status word and the state vectors z_s , \dot{z}_s , and f_s . These Cartesian vectors give the current state of the telerobot in coordinates C_z . The vector f_s represents world modeling's "best guess" of the forces in the servo coordinates. This information may represent a fusion from a number of sensors including wrist force/torque sensors, joint torque sensors, and tactile force sensors.

The six-dimensional Cartesian velocity vectors in the interfaces have the form

$$[v_x, v_y, v_z, \omega_x, \omega_y, \omega_z],$$

where v_x , v_y , and v_z are the linear velocity components of the end-effector motion with respect to the x, y, and z axes of the control coordinates, and ω_x , ω_y , and ω_z are the angular velocity components about the same axes. The six-dimensional Cartesian force vectors are defined analogously. The definition of the position vectors is not as straightforward as the others. To avoid the ambiguities of using only three orientation parameters, the orientation part of the position vector is represented by an equivalent angle-axis form [11]. Thus, the form of the position vector is

$$[x, y, z, \theta, n_x, n_y, n_z],$$

where x, y, and z give the position with respect to the origin of the control coordinates, and

the orientation is given by a rotation θ about the unit vector \mathbf{n} in the same coordinate system.

The interfaces presented here support a number of different algorithms for manipulator teleoperation. For example, a hybrid position/force scheme can be invoked using end-effector coordinates, i.e., a Cartesian frame fixed in the telerobot end effector, for use with a joystick [4]. The joystick inputs are interpreted as force commands along the degrees of freedom in which the telerobot is constrained by the environment, and as rate commands otherwise.

$$\mathbf{u}_{\text{telerobot}} = \mathbf{K}_v {}^e\mathbf{J}^{-1}(\theta) \mathbf{S}'(\dot{\mathbf{z}}_m - \dot{\mathbf{z}}_s) + {}^e\mathbf{J}^t(\theta) \mathbf{S} \mathbf{f}_m + \mathbf{K}_{pf} {}^e\mathbf{J}^t(\theta) \mathbf{S}(\mathbf{f}_m - \mathbf{f}_s)$$

In this control expression, ${}^e\mathbf{J}^{-1}(\theta)$ represents the inverse of the Jacobian relating joint rates to end-effector coordinates and ${}^e\mathbf{J}^t(\theta)$, the transpose of this Jacobian.

Active control of the teleoperation device is achievable in numerous ways. The technique given in [9] for the Force-Reflecting Hand Controller is straightforward.

$$\mathbf{u}_{\text{device}} = \mathbf{K}_{fm} \mathbf{J}^t(\theta_m) \{ \mathbf{K}_{pm} \text{Dbnd}(\mathbf{z}_s - \mathbf{z}_m) - \mathbf{K}_{vm} \dot{\mathbf{z}}_m + \mathbf{f}_s \}$$

The function $\text{Dbnd}(\)$ is a deadband function which eliminates corrections for position errors of small magnitude. The above equation differs slightly from the algorithm of [9] in that the T6 matrix is not used to represent Cartesian positions. However, the error term $(\mathbf{z}_s - \mathbf{z}_m)$ can be computed from the seven-dimensional position form to give a six-dimensional error vector [6,12]. The choice of seven-dimensional positional vectors instead of the 12-element matrix representation is compatible with the Force-Reflecting Hand Controller functions. The use of T6 for this device is simply a matter of computational convenience, as expressed in [9].

Any scaling or indexing of the teleoperation commands performed for the convenience of the operator should be handled in the World Modeling module of the teleoperation device. Thus, the teleoperation commands are ready to be used in the control when they enter the global data system of the telerobot.

The data format and data rate requirements for the Cartesian teleoperation interfaces can be determined in manner similar to that used for the joint-space interfaces of section 2. It seems practical to suggest that the floating-point formats and communication protocols chosen for joint-space interfaces be used for Cartesian interfaces as well. Note, however, that all parameters in the interfaces need not be communicated every time. It is expected that the parameters Algorithm, \mathbf{C}_z , \mathbf{S} , \mathbf{S}' , and the \mathbf{K} 's will not change as rapidly as the other parameters, and therefore do not need to be communicated as often. For this reason, these parameters can appear as optional parameters at the end of the communication buffer. They would only be transferred when the values change.

One important consideration regarding Cartesian teleoperation is the handling of singularities. The input to the Servo Level of the telerobot controller must be "small in a dynamic sense" [10]. This means that each input should define a relatively small change in motion for the manipulator. When the servo algorithm uses Cartesian inputs this can be difficult to achieve, since a small Cartesian motion along a nearly singular direction produces a large change in the joint-space control input. During autonomous operation it is the Primitive Lev-

el's job to ensure that input commands do not exercise a singularity. This same criterion applies to the inputs from a Cartesian teleoperation device – the inputs to Servo must not ask for motions along singular directions.

Note that this does not mean that the operator cannot move the telerobot into a singular region. However, in a singular region, control should probably be handled in two ways. First of all, the algorithm executing in the Servo Level should monitor all the singular expressions of the determinant of the telerobot Jacobian [6]. When the algorithm detects that the mechanism has entered the region of a singularity, the control should switch to treat the mechanism as redundant with respect to the motion of the end-effector in the subspace orthogonal to the singular direction [6]. A function based on the *manipulability measure* [7], operating in the associated null space, can be used to control the movement along the singular direction [6]. Secondly, the operator's command input should be restricted to movement which does not exercise the singularity. This means that the Cartesian input is scaled according to the manipulability measure, and, at the point of singularity, reduced to the remaining degrees of freedom. For a Cartesian teleoperation device with force feedback this means that the manipulability ellipsoid [7] should be reflected back to the operator such that the operator senses the mechanism's singular regions. In order for the teleoperation device to specify the correct inputs it must have a local model of the telerobot. The correct inputs can be determined by using the feedback position of the telerobot θ_s and the local telerobot model.

4. Conclusion

This document has presented basic teleoperation interfaces for the two main classes of teleoperation devices, joint-space devices and Cartesian devices. These interfaces support numerous algorithms for teleoperation.

These teleoperation interfaces are compatible with the Servo telerobot system described in [10]. In particular, the interfaces of tables 1 and 3 represent special cases of the general operator control interface at the Servo Level [10]. The information on the state of the telerobot fed back to the teleoperator, i.e., the data of tables 2 and 4, is available from the global data system of the telerobot control architecture. The world modeling modules of the telerobot update this information continuously so that the most recent data will be available to all systems which need it.

5. References

- [1] Albus, J. S., McCain, H. G., Lumia, R., NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM), NBS Technical Note 1235, July 1987.
- [2] Vertut, J., Coiffet, P., Teleoperations and Robotics: Evolution and Development, Robot Technology Series, Vol. 3A, Prentice-Hall, Englewood Cliffs, NJ, 1986, pp. 195-210.
- [3] Vertut, J., "Control of Master Slave Manipulators and Force Feedback," Joint Automatic Control Conf., San Francisco, June 1977.
- [4] Hirzinger, G., "The Space and Telerobotic Concepts of DFVLR ROTEX," IEEE Conf. Robotics & Automation, Raleigh, NC, March 1987.

- [5] Thring, M. W., Robots and Telechairs, Ellis Horwood Limited, Chichester, England, 1983.
- [6] Khatib, O., "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," IEEE Jour. Robotics & Automation, Vol. RA-3, No. 1, February 1987.
- [7] Yoshikawa, T., "Manipulability and Redundancy Control of Robotic Mechanisms," IEEE Conf. Robotics & Automation, St. Louis, March 1985.
- [8] Paul, R. P., Robot Manipulators: Mathematics, Programming, and Control, MIT Press, Cambridge, MA, 1981.
- [9] Handlykken, M., Turner, T., "Control System Analysis and Synthesis for a Six Degree-of-Freedom Universal Force-Reflecting Hand Controller," Nineteenth IEEE Conf. Decision & Control, Albuquerque, NM, December 1980.
- [10] Fiala, J. C., "Manipulator Servo Level Task Decomposition," NIST Technical Note, NIST, Gaithersburg, MD, October 1988.
- [11] Craig, J. J., Introduction to Robotics; Mechanics and Control, Addison-Wesley, Reading, MA, 1986, p. 46.
- [12] Luh, J. Y. S., Walker, M. W., Paul, R. P. C., "Resolved-Acceleration Control of Mechanical Manipulators," IEEE Trans. Auto. Control, Vol. AC-25, No. 3, June 1980.
- [13] Townsend, M. J., The Data Communications Source Book: Vol. 1, The Merritt Company, 1987.

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET <i>(See instructions)</i>	1. PUBLICATION OR REPORT NO. NIST/TN-1254	2. Performing Organ. Report No.	3. Publication Date October 1988
4. TITLE AND SUBTITLE <p style="text-align: center;">Interfaces to Teleoperation Devices</p>			
5. AUTHOR(S) <p style="text-align: center;">John Fiala</p>			
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (formerly NATIONAL BUREAU OF STANDARDS) U.S. DEPARTMENT OF COMMERCE GAITHERSBURG, MD 20899		7. Contract/Grant No.	8. Type of Report & Period Covered <p style="text-align: center;">Final</p>
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP) <p style="text-align: center;">S/A</p>			
10. SUPPLEMENTARY NOTES <p><input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.</p>			
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here) <p style="text-align: center;">This document describes a basic architecture for teleoperation device control systems and describes interfaces to robot control systems. The ideas incorporated in this document are derived from the NASREM Architecture specification.</p>			
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) <p style="text-align: center;">master-slave teleoperation; robot control; servo control; teleoperation; teleoperation interfaces; telerobot control system</p>			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		14. NO. OF PRINTED PAGES <p style="text-align: center;">14</p> 15. Price	

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899

Official Business
Penalty for Private Use \$300



Stimulating America's Progress
1913-1988