1 of 1

Conf-9309199--9

UCRL-JC-115169
PREPRINT

Telerobotics with Whole Arm Collision Avoidance

K. Wilhelmsen
S. Strenn

This paper was prepared for submittal to the
SPIE - The International Society for Optical Engineering Conference
September 7-9, 1993
Boston, MA

NOV 03 1993
OSTI

September 1993

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Telerobotics with Whole Arm Collision Avoidance

Karl Wilhelmsen, Member, IEEE
Stephen Strenn, Member, IEEE

Abstract

The complexity of telerobotic operations in a cluttered environment is exacerbated by the need to present collision information to the operator in an understandable fashion. In addition to preventing movements which will cause collisions, a system providing some form of virtual force reflection (VFR) is desirable. With this goal in mind, Lawrence Livermore National Laboratory (LLNL) has installed a kinematically similar master/slave system and developed a whole arm collision avoidance system which interacts directly with the telerobotic controller. LLNL has also provided a structure to allow for automated upgrades of workcell models and provide collision avoidance even in a dynamically changing workcell.

Introduction

Telerobotics has recently become an active area of research. These efforts have largely been driven by the need to perform complex tasks in hazardous environments (1,2,3,4). Such is the case in radioactive waste processing, space-based assembly, and high voltage work. Humans tend to be included in the control loop when novel or emergency situations arise. The objective of telerobotics, then, is to allow humans to stay in the control loop but out of the hazardous environment. A typical telerobotic system consists of a human, a *control arm*, and a *remote arm*. In such a case, the human manipulates a control arm and the remote arm follows kinematically similar motions.

At Lawrence Livermore National Laboratory (LLNL) there are several tasks requiring the performance of complex manipulation in a hazardous environment. Because of the hazardous nature of these tasks, these manipulations need to be performed remotely. The need for these remote presence manipulators has driven the development of telerobotics systems at LLNL.

A representative telerobotic task is the manipulation of mixed waste. Mixed waste is defined as waste containing both a radioactive hazard and a biological hazard such as a carcinogen or toxin. There are currently over 3 million cubic yards of mixed waste throughout the Department of Energy (DOE) (5). LLNL, other national labs and weapons sites have the responsibility for treatment of this waste. As part of LLNL responsibilities, LLNL is building a pilot scale system to treat low level mixed waste called the Mixed Waste Management Facility (MWMF).

LLNL will construct a facility to develop, demonstrate, test and evaluate the integration and operation of the best available mixed waste treatment technologies that provide alternatives to incineration for the treatment of low-level mixed wastes. MWMF will be designed to support the DOE's National Mixed Waste Strategic Plan and will provide major technical support to the solution of mixed waste treatment problems at other DOE sites. The conduct of such a development, and demonstration project at LLNL will greatly benefit the efforts of DOE and the State of California in finding solutions to the treatment of mixed wastes for disposal.

The decontamination of mixed waste in general involves the separation of materials, many in plastic bags, into treatable categories. Mixed waste will be sorted into categories including combustibles, metals, surface contaminated solids, and vitrifiable materials. All organic toxins will be oxidized, but waste will not be incinerated. After the toxins have been oxidized, the waste will be separated into reclaimable materials and low level radioactive waste to be stored. To accomplish this process, a large number of tasks must be accomplished with telerobotic systems.

The remote manipulation tasks inside the MWMF facility will include; venting barrels, dumping of barrels, size reduction of waste, singulation of solid objects, sensor manipulations, sample handling, transportation of waste and equipment, and maintenance of all equipment in containment enclosures. Although, many of these operations can be performed with hard automation, some operations will require telerobotic interaction. Collisions by the robot may precipitate the spread of contamination or cause damage to MWMF systems or the telerobot. Given the nature of the material to be handled, said collisions are unacceptable.

Since the waste to be manipulated is perpetually changing, the environment can be considered unknown. To navigate in this unknown environment, an accurate measurement of that environment is required. The more accurate the measurement of the environment, the better the operation of the collision avoidance system. This measurement of the workspace will be referred to as workcell calibration.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

for

Workcell Modeling

Initial work in the area of workcell modeling at LLNL has relied on CAD models. In high accuracy collision avoidance, the CAD model of the work space can be adjusted for errors in the robot positioning. By compensating for this positional error, the robot positional accuracy can approach the robot positional repeatability. This compensation will be referred to as robot calibration.

The workcell becomes dynamic and unknown when waste objects are introduced into the system. At this point, CAD modeling tends to be insufficient for workcell calibration. Therefore, a sensor system capable of providing geometric information for the workcell model will be needed. The sensor being incorporated is an optical range sensor. The optical range sensor maps the distance from the sensor to each point in a video image. Experimentation with two types of optical range sensor have been completed.

Range Sensors

The first employs binocular vision. A binocular vision range sensor uses two video cameras separated by a known distance (see Figure 1). All points common to both images are determined by measuring relative intensity information for each image pixel. After a point is determined to exist in both images, the distance from the point to the binocular range sensor is determined by triangulation.

$$\text{height} = \frac{\sigma}{\tan \theta_1 + \tan \theta_2} \quad \text{With knowns } \theta_1, \theta_2 \text{ and } \sigma \quad (1)$$

Fig. 1. Binocular vision determination of height through triangulation.

The binocular range sensor⁽⁶⁾ can make accurate models of surfaces but, because it is dependent on relative image intensity, it is subject to error from spectral reflection. Specular reflection is a particular problem with certain materials like plastic bags. The intensity information in each view of a bag can have a large variance depending on lighting sources. The variance in the intensity information can create errors in the determination of points common to both images producing large errors in triangulation. To minimize the errors from specular reflection, multiple algorithms for determining which points are common to both images are combined. Although these algorithms require extensive computation resources, binocular vision has additional advantages beyond just measuring the range data. Binocular vision can also provide binocular three-dimension vision for an operator during teleoperation and poses no eye safety issues. These advantages motivate the continued investigation of binocular vision sensors.

Another optical range sensor being investigated is a structured lighting system developed by Oak Ridge National Laboratory (ORNL). The structured lighting system is similar to the binocular vision system, but uses a single video input and a laser light fan (see Figure 2). The laser light fan draws lines on surfaces at set angles relative to the video input. The drawn line is captured in the video image and the effective angle, q_2 , is determined directly from this image (see Equation 1). Since the structured lighting system range picture is not dependent on the relative intensity of two video images, as is the binocular vision range sensor, structured lighting allows for simplified determination of height calculations using triangulation. Still another method of obtaining range images is laser radar, commercially available for scanning parts and workspaces.

Fig. 2. Structured light determination of height through triangulation.

In addition to the simplification of height calculations, a structured light range sensor is less subject to specular reflections than binocular vision range sensors. In the case of waste objects in plastic bags, the laser penetrates the plastic bags and allows for modeling of the actual waste instead of the bag. The negative aspects of structured lighting are the eye safety issues, and laser energy needed to insure sufficient light laser reflections for the camera to detect.

CAD Modeling

An individual range picture is insufficient for generating CAD models of the work space. Range images from multiple views must be collected and combined to complete workcell calibration. These multiple views will be obtained by having the robot manipulate the optical range sensor. The robotic manipulation of the range sensor has multiple advantages. First, the sensor can be used to model the interior of waste containers by using the robot to place the sensor through an entry hole. Second, as the robot moves, it can continually monitor for changes in the workspace and upgrade the workcell model transparent to the operator. In addition, by using the robot for sensor manipulation, the sensor can be used for robot calibration. Although both optical range sensors have been tested, a system capable of determining the complete CAD model has not yet been integrated. There are several commercial vendors emerging which can convert multiple range images into CAD data files, but a commercial system is not scheduled to be integrated into the LLNL collision avoidance (CA) system until 1994.

Collision Avoidance System

The LLNL CA system works in concert, with a graphical programming system and a force reflecting hand-controller. Before a collision occurs, a retarding force is applied to both the robot and the teleoperator input system. These retarding forces slow the robot down to prevent damaging collisions and inform the teleoperator of the collision in a non obtrusive manner. Clearly, since such a CA must operate in real time, this is a computationally daunting task, especially in a changing workcell. To accomplish this task, a potential fields based approach, known as the segmentation algorithm (SA) was developed.

The SA algorithm was designed with the following statement in mind: a teleoperator should be able to manipulate a control arm to easily move a remote arm through a complex, dynamic workspace in a collision-free manner. Inherent in this statement is the need for fast, on-line CA processing. However, the potential fields approach has heavy computational costs associated with it. The SA addresses this problem by employing two levels of segmentation to reduce these costs. The first level uses a linear workspace grid to separate the robot and objects by spatial location. Objects which are outside of the threshold distance, d , are not processed. Objects within d units of the robot apply virtual forces which are reflected to the teleoperator. The second level segments the robot arm by link (or sublink). In this way, only segments of the arm which have objects close to them are processed.

Systems Overview

In terms of overall teleoperation system control, CA is accomplished by adding torque's to the control arm which counteract the human operator's dangerous movements. The basic flow graph is shown in Figure 3. In the figure, t_h is the torque due to the human operator's manipulation, and t_{ca} is the torque generated by the CA system. The run-time input to the CA system is the current remote arm configuration, q . This information, coupled with the known locations of the workspace objects, is used to calculate appropriate output torques to apply to the control arm to avoid any obstacles. As stated above, these torque's provide virtual force reflection to the teleoperator.

Fig. 3. Basic control scheme for a telerobotic system with CA.

Note the arrow between the "Collision Avoidance System" and "Remote Arm Controller" blocks in Figure 3. It represents a wide variety of information that could be sent directly to the remote arm controller, bypassing the control arm loop. For example, since most control arms can be over-powered by a human operator, there may be times when the human, perhaps unwittingly, overcomes the CA torque's, t_{ca} . In this case it is necessary to send commands directly to the remote arm controller to avoid collision.

The controller arm used in the initial implementation was a Schilling force reflecting master arm. The Schilling master arm is a six axis robotic arm capable of supplying 3 lbs force along each axis with a maximum bandwidth of approximately 3 Hz. The Schilling master arm is manipulated by a teleoperator and the arm can control the forces exerted on the teleoperator. To maximize the useful operating range of the Schilling master, the forces needed to maintain the Schilling master arm position were minimized. The forces on each joint of the Schilling master arm were measured and all forces were amplified with a positive force feedback loop. This positive force feedback had the effect of pulling the master arm away from any operator induced forces, minimizing the forces an operator needed to impart on the master arm movement. In addition, a complete gravity compensation system was included to eliminate the effective weight of the arm in the controller.

The robot controlled was a Puma 562. An interface to the VAL II controller from a real time system is accomplished using a software package developed by Sandia National Labs called Sequential Modular Architecture for Robots and Teleoperation (SMART). The real time operating system communicates to the CA and the Schilling controller using a dedicated Ethernet line and Berkeley Construct message handler (see Figure 4).

Fig. 4. An overview of the hardware used. The dotted line refers to future developments.

Collision Avoidance Algorithm Development

The concepts on which the SA is founded are developed in this section. It opens with a discussion of the workcell model which consists of robot and workspace primitives. Next, the hierarchical segmentation techniques mentioned above are explained and their effects investigated. This is followed by a detailed description of the SA's run-time operation. In addition, the control arm joint torques and velocities are derived. Finally, procedures which account for dynamic workcell conditions are developed.

Workcell Model Description

The workcell is composed of the robot and the objects within its reach. These, in turn, are made up of primitives. Recall that a primitive is one of a set of geometric objects that in sum represent a real object. The SA assumes that there are two basic types—robot primitives (*rprims*) and workspace primitives (*wprims*). In order to emphasize the general nature of the SA, no particular geometries have been chosen for the *rprims* and *wprims*.

Rprims, denoted by r , model the surface of the robot. In this paper, it is assumed that there is one *rprim* for each of the J links of the robot. In addition to its intrinsic geometry, each *rprim* has associated with it a bounding box (*bbox*) which completely encloses the *rprim* and is rectilinear in the world coordinate frame (O_0). The *bbox* makes it easy to identify the general region in which the *rprim* lies. Each *rprim* is considered to be rigidly attached to a coordinate frame (O_i), the location of which is determined by the robot's Denavit-Hartenburg (D-H) parameters.

Wprims, denoted by w , model the surface of workspace objects. *Wprims* differ from *rprims* in two ways: they do not have *bbox* but do have a field strength attribute. Actually, the field strength is an array of strengths, with one element for each *rprim*. Each element in the array is an indication of the strength with which this *wprim* will repel the corresponding *rprim*. As will be seen in later, this attribute allows for an interaction between the robot and its environment known as obstacle accommodation. Every *wprim* is considered to be rigidly attached to a coordinate frame (O_w).

Segmentation Scheme

It was noted previously that the operator should be able to feel nearby objects in the remote arm's environment. The term "nearby" indicates that objects outside some threshold distance, δ , will not generate torques on the control arm. Indeed, objects outside the threshold distance represent no danger of collision and applying the corresponding torques would only serve to confuse the operator. Therefore, these objects can be excluded from collision avoidance processing. This fact underlies the speed gains that can be achieved through workspace segmentation.

When δ is a fraction of the length of the arm, a *wprim* that is within δ units of some part of the arm will tend to be more than δ units away from some other part of the arm. This statement implies that segmenting the robot arm minimizes the number of *rprims* involved in collision avoidance processing. For example, in Figure 5 the *wprims* closest to link 2 are more than δ units away from link 1. Therefore, no distance calculations need to be made between these *wprims* and link 1. The most obvious segmentation scheme breaks the arm down by link, but sublink segmentation is justifiable in certain situations.

Fig. 5. Robotic arm and workspace in a linear segmentation grid

The theoretical benefits of workspace and arm segmentation can be clarified by investigating their effects on the Brute Force Algorithm's (BFA) timing in equation (2). The objective of segmenting the arm and workspace is to reduce N_r and N_w in equation (2) to N_{rd} and N_{wd} , where N_{rd} is the number of rprims with wprims within d units, and N_{wd} is the number of wprims within d units of any rprim. Hence,

$$T_{BFA} = N_r[T_u + N_w(T_d + JT_r + T_b)] \quad (2)$$

becomes,

$$T_u = N_{rd}[T_k + N_{wd}(T_d + JT_r + T_o)] \quad (3)$$

where

J	Degrees of freedom of the robot
N_r	Number of robot primitives
N_w	Number of workspace primitives
T_d	Time to calculate the distance
T_k	Time to forward kinematic transform a robot primitive
T_o	Time spent in miscellaneous overhead operations
T_r	Time to calculate the torque due to a distance vector
T_u	Time spend using the SA Algorithm

While the equation remains much the same, the performance improvement can be dramatic, as will be seen later in this paper.

Basic SA Operation

The SA is quite complex in its complete form, but an attempt is made here to describe its run-time operation. Every T_u seconds, the new robot joint position, q , is received from the remote arm controller (see Figure 3). First, the search region for rprim i is determined by transforming its bbox, r_{bb} from O_i to O_0

$$r_{bb0} = T_{0i}r_{bb} \quad (4)$$

where T_{0i} is the forward kinematic transform from O_0 to O_i . The point defining the bbox, O_0 , are then transformed into O_g according to equation (5). A point p_0 in O_0 can be transformed into O_g as shown below

$$\mathbf{P}_g = \begin{bmatrix} \text{floor} \left(GX \frac{p_{0x} - XMIN}{XMAX - XMIN} \right) \\ \text{floor} \left(GY \frac{p_{0y} - YMIN}{YMAX - YMIN} \right) \\ \text{floor} \left(GZ \frac{p_{0z} - ZMIN}{ZMAX - ZMIN} \right) \end{bmatrix} \quad (5)$$

where the $\text{floor}(\cdot)$ function removes the fractional portion of its argument, and $(XMIN, YMIN, ZMIN)$, $(XMAX, YMAX, ZMAX)$ are the minimum and maximum workspace coordinates in O_0 .

A one grid segment buffer is added to the grid space bbox. The enlarged grid space bbox is the search region for rprim i . If the search region has no wprims in it, then no further processing of this rprim takes place. Otherwise, a forward kinematic transform is applied to the rprim's geometry and the distance vectors (and corresponding control arm torques) between the rprim and all wprims in the search region are determined. This procedure is continued for $i \dots j$. A pseudo-code version of the procedure is given below.

Receive q from remote arm controller

For each rprim(i) in robot

 Forward kinematic xform rprim's bbox

 Xform bbox from O_0 to O_g

 Add one segment buffer to get search region

 If search region is not void of wprims

 Forward kinematic xform rprim

 For each wprim in search region

 Calc rprim to wprim distance vector

 Determine corresponding control arm torq

Send torques to control arm controller

Note the savings achieved when the search region is empty. No forward kinematic transforms or distance calculations have to be made. This is very significant when one considers that while carrying out most tasks, only the end-effector is close to any workspace objects. Likewise, only a small percentage of all wprims in the workspace will be near the robot during any given update period (and even these will tend to be near the end-effector). Therefore, N_{rd} and N_{wd} will indeed be small with respect to (w.r.t.) N_w and N_r respectively.

Joint Torque Derivation

In this section, a derivation of the torques to be applied to the control arm is presented. Recall that the torques arise when some part of the robot comes near a workspace object. Each wprim projects a potential (force) field into the workspace whose strength is inversely related to the distance from the wprim. In SA parlance, the distance and direction between the two primitives is called a distance vector. The distance vector in turn becomes a virtual force applied to the remote arm at the point nearest the wprim. The torque on each control arm joint is a summation of the torque's generated by the virtual forces of all wprims within d units of all rprims.

As noted above, the particular geometries of rprims and wprims have intentionally been left unspecified. This makes the task of deriving the distance vector between two primitives somewhat unstructured. Fortunately, a large body of published material on the subject of minimum distance calculation exists (7, 8, 9, 10, 11, 12) et al. Therefore, it is assumed here that the distance vector between wprim i and rprim j can be calculated as

$$\mathbf{d}_{ij} = \|\mathbf{r}_j - \mathbf{w}_i\| \quad i = 1 \dots N_{w\delta} \quad j = 1 \dots N_{r\delta} \quad (6)$$

Since joints $k=1 \dots j$ can be controlled to help avoid a collision at joint j , the lever arms and distance vectors given w.r.t. O_k need to be determined. Figure 6 depicts these relationships.

Fig. 6. The distance vector generates a torque on all lower joints

Therefore, let $\mathbf{d}_{ijk}$ be the distance vector given in Equation (6) expressed in $\mathbf{O}_k$.

$$\mathbf{d}_{ijk} = \mathbf{R}_{kj} \mathbf{d}_{ij} \quad k = 1 \dots j \quad (7)$$

where $\mathbf{R}_{kj}$ is the rotation matrix from $\mathbf{O}_k$ to $\mathbf{O}_j$. The virtual force vector is inversely related to the distance vector and is given by

$$\mathbf{f}_{ijk} = \frac{\zeta_{ij}}{\|\mathbf{d}_{ijk}\|} \begin{bmatrix} \frac{d_{ijkx}}{\|\mathbf{d}_{ijk}\|} & \frac{d_{ijk y}}{\|\mathbf{d}_{ijk}\|} & \frac{d_{ijkz}}{\|\mathbf{d}_{ijk}\|} \end{bmatrix} \quad (8)$$

where ζ_{ij} is $\mathbf{w}_i$'s field strength w.r.t. $\mathbf{r}_j$. z_{ij} controls the degree of interaction between $\mathbf{w}_i$ and $\mathbf{r}_j$. When ζ_{ij} is at a maximum, the repelling force $\mathbf{f}_{ijk}$ is large and no contact is allowed. When ζ_{ij} is zero, however, $\mathbf{f}_{ijk}$ is zero and the $\mathbf{r}_{prim}$ and $\mathbf{w}_{prim}$ can touch. This flexibility is essential for allowing the end-effector to approach and grasp workspace objects while still maintaining CA for the rest of the arm.

As seen in Figure 6, $\mathbf{v}_j$ is the lever arm from the point on the $\mathbf{r}_{prim}$ nearest the $\mathbf{w}_{prim}$ to the joint j origin. This lever arm, expressed as $\mathbf{v}_{jk}$ in $\mathbf{O}_k$, is

$$\mathbf{v}_{jk} = \mathbf{T}_{jk} \mathbf{v}_j \quad (9)$$

In general, a torque is calculated in vector form as $\mathbf{t} = \mathbf{v} \times \mathbf{f}$, but since revolute joints only rotate about the z-axis,

$$\tau_{ijk} = \alpha_k (v_{jkx} f_{ijk y} - v_{jky} f_{ijk x}) \quad k = 1 \dots j \quad (10)$$

where α_k is a positive gain constant appropriate for control of joint k . When multiple $\mathbf{w}_{prim}$ s act on the robot, the effects on the joint torques are cumulative. Thus the torque on joint k due to the N_{wd} $\mathbf{w}_{prim}$ s acting on the robot is

$$\tau_k = \sum_{i=1}^{N_{wd}} \tau_{ijk} \quad k = 1 \dots J \quad (11)$$

Alternatively, the joint torques can be found by use of a modified Jacobian, $\mathbf{J}^{(13)}$. In this case,

$$\tau_{ijk} = \mathbf{J}_k^T \mathbf{f}_{ijk} \quad (12)$$

where $\mathbf{J}_k$ is the robot's Jacobian from $\mathbf{O}_0$ to $\mathbf{O}_j$ with the link lengths replaced by the corresponding components of $\mathbf{v}_{jk}$ defined above.

Note that Equation (11) doesn't represent the entire torque actually applied to each joint. It is the total torque added to help the teleoperator avoid collisions—denoted as t_{ca} in Figure 3. In the next section, the question of *what are the joint velocity maximums?* is addressed.

Dynamic Workcells

The SA exhibits its flexibility by easily accounting for rotations, translations, and grasping of workspace objects. This is a distinct advantage over neural network and configuration space methods. In C-space approaches, for example, when the robot picks up a sizable object, the entire memory map may have to be regenerated. This is not a reasonable solution for complex three-dimensional workspaces. In the SA, however, these events are accounted for by kinematic transformations.

Assuming that wprims are rigidly attached to their coordinate frames, rotation and translation can be accomplished by updating the transform relating O_0 to O_w , T_{O_0w} . Arbitrary rotations of coordinate frames can be accomplished in many ways. One common method is called the Euler Angles Representation⁽¹⁴⁾. In this representation, the following sequence of rotations are carried out: f about the z axis, g about the rotated x axis, and y about the (twice) rotated z axis. Translation of the coordinate frame by some vector distance, v , is accomplished by extending the Eulerian rotation matrix into a homogeneous transformation matrix

$$T_e = \begin{bmatrix} R_e & v \\ 0 \dots 0 & 1 \end{bmatrix} \quad (13)$$

where R_e is the Eulerian rotation matrix. Thus, when a wprim is rotated and translated according to T_e , T_{O_0w} is adjusted according to

$$T'_{O_0w} = T_e T_{O_0w} \quad (14)$$

When the robot picks up an object, its wprim(s) become rigidly attached to the coordinate frame O_J (the end-effectors coordinate frame). As far as the SA is concerned, it is simply an extension of the end-effector and as such is protected from collision. Note that this changes the kinematics of the robot—an event accounted for by transforming the new rprim's geometry from O_w to O_J according to

$$r = T_{O_J}^{-1} T_{O_0b} w \quad i = 1 \dots M \quad (15)$$

Equation (16) holds for any geometry that can be expressed as a parametric vector of the form

$$w = [f(u) \ f(v) \ f(w)]^T \quad (16)$$

Obviously, the wprim(s) is deleted from the workspace grid at this time. When the object is released, the problem becomes unstructured because the final orientation of the object is unknown. Even if the approach vector for release is the same as when the object was picked up, it could tip, topple, or roll when released. In this case it is safest to use sensors to remodel and add the object to the list of workspace primitives.

Performance Analysis

Performance Metrics and Simulation Environment

In a telerobotic system, the primary performance metric is the update frequency, f_u . At the human operator level, $f_u > 30$ Hz translates into useful force feedback information⁽¹⁵⁾. That is, when the remote arm approaches a workspace object, the operator will feel crisp, clean forces on the control arm. By contrast, as $f_u \approx 4 \sim 8$ Hz, said forces may seem to be sluggish and somewhat out of sync with any visual feedback. Below 4 Hz, the operator will begin to notice a distinct time lag between control arm manipulations, visual feedback, and torques coming in from the CA system (see Figure 3).

The simulations used a Puma 560 robot with $N_r = 6$ located in a 3000 mm^3 workspace. Points and boxes were chosen as primitive geometries for rprims and wprims respectively. All objects in the workspace were modeled with a maximum error of 25 mm^3 .

Typical simulations were conducted in the following manner. Let's assume that N_w is the parameter of interest: all other SSM parameters are held constant. Now, during any *single* update period, the SA's performance is a function of the arm configuration, the number of wprims in the workspace, and where those wprims are located. Hence,

$$f_u(\mathbf{q}, N_w, \omega), \quad (17)$$

where w is a sample taken from wprim distribution W (i.e. where the wprims are located). Averaging the measured values of $f_u(\mathbf{q}, N_w, w)$ over large sets of $\mathbf{q}$ and w removes the corresponding dependencies. This yields the desired relationship of $f_u(N_w)$.

Fig. 7 f_u and N_w for various δ ; $\delta = 3000 \text{ mm}$ is the BFA

Figure 7 shows the relationship between f_u and N_w for various buffer sizes, δ . As either δ or N_w increases, performance degrades. The figure shows that with a 12cm buffer around the robot arm and the workspace modeled by 250 wprims, an f_u of greater than 30 Hz is feasible.

Perhaps the best way of comparing the SA to the BFA is to look at the frequency with which important routines are called. For example, matrix multiplications are used extensively in forward kinematic transformations and torque vector calculations. In addition, point to box distance calculations are carried out between the hundreds of points making up the robot arm and the boxes representing workspace objects. In Table 1, the number of calls to matrix multiplication and distance vector calculation routines are shown for simulations run on equivalent workcells. The striking difference in number of calls to each routine stands as a testament to the fact that in the SA, most of the rprims and wprims are eliminated from processing during any given update period. Even more important than the speed up factor of the SA system, it shows the Puma 560 can be controlled with collision avoidance while maintaining $f_u > 30 \text{ Hz}$ with a maximum error of 25 mm. Also, as the computation power is increased the resolution can be improved.

Table 1. Simulated collision avoidance with 50 objects for 100 interactions

	SA	BFA
Matrix Multiplies	13058	64572
Distance Calculations	2443	2835000

Conclusions

The experimental system described demonstrates that usable collision avoidance can be obtained using off the shelf technology. Combining the segmentation algorithm mapping system with the workcell calibration system under development, a completely transparent collision avoidance system is possible with current technology. This form of collision avoidance provides virtual force reflection in an inherent manner for an operator to understand the remaining usable joints. Each object in a workcell generate a retarding potential field to each joint of the robot. To improve the update frequency of the collision avoidance, the segmentation algorithm system was developed, which increased the frequency response of the collision avoidance by a factor of 10 beyond the brute force algorithm. As an added bonus beyond collision avoidance, the proposed structure of the system will provide a continuously updated workcell model for maintaining system modeling and off-line programming. With the technology described, a resolution of 25 mm squared was obtained, but with faster computation resources, a commensurate improvement in resolution can be obtained.

This work was performed under the auspices of the U.S. DOE by Lawrence Livermore National Laboratory, Livermore California under Contract No. W-7405-Eng-48.

References

1. V. Lumelsky, "On Human Performance in Telerobotics," *IEEE Trans. on Sys., Man, and Cybern.*, vol 21, no 5, Sept 1991
 2. R.F. Foyle, "The Use of Teleoperators in Hostile Environmental Applications," *Proc. IEEE Int. Conf. Robotics and Automation*, 1992
 3. P.M. Bobgan, H. Kazerooni, "Achievable Dynamic Performance in Telerobotic Systems," *Proc. IEEE Int. Conf. Robotics and Automation*, 1991
 4. A.K. Bejczy, Z.F. Szakaly, "A Harmonic Motion Generator for Telerobotic Applications," *Proc. IEEE Int. Conf. Robotics and Automation*, 1991
 5. Congress of the United States, Office of Technology Assessment, *Complex Cleanup, The Environmental Legacy of Nuclear Weapons Production.*, 1991.
 6. S-Y Lu, R. K. Johnson, J. E. Hernandez, Lawrence Livermore National Laboratory, *Vision-based Grasping for Autonomous Sorting of Unknown Objects.*
 7. S. Cameron, "A Study of the clash Detection Problem in Robotics," *Proc. IEEE Int. Conf. Robotics and Automation*, March 1985
 8. S. A. Cameron, R. K. Culley, "Determining the Minimum Translational Distance Between Two Convex Polyhedra," *Proc. IEEE Int. Conf. Robotics and Automation*, April 1986
 9. W. Meyer, "Distances Between Boxes: Applications to Collision Detection and Clipping," *Proc. IEEE Int. Conf. Robotics and Automation*, April 1986
 10. S. Strenn, "Model-Based Collision Avoidance for Telerobotic Applications," Masters Thesis, University of California at Davis, June 1993
 11. M. E. Mortenson, *Geometric Modeling*, John Wiley and Sons, 1985
 12. G. Hurteau, N.F. Stewart, "Distance Calculation for Imminent Collision Indication in a Robot System Simulation," *Proc. 13. R. Volpe, P. Khosla, "Artificial Potentials with Elliptical Isopotential Contours for Obstacle Avoidance," Proc. 26th Conf. on Decision and Control*, Dec 1987
 14. K. S. Fu, R. C. Gonzalez, C. S. G. Lee, *Robotics—Control, Sensing, Vision, and Intelligence*, McGraw-Hill, 1987
 15. McCormick, E., *Human Factors Engineering and Design*, McGraw Hill, 1976
- IEEE Int. Conf. Robotics and Automation*, 1991

**DATE
FILMED**

12 / 10 / 93

END