SAND94-0500C
Conf-940550--11

Using Virtual Objects to Aid Underground Storage Tank Teleoperation

Robert J. Anderson
Brady Davies

Sandia National Laboratories,
Intelligent Machine Systems Division I: 2161,
P.O.Box 5800, Albuquerque, NM 87185[1]

Abstract

In this paper we describe an algorithm by which obstructions and surface features in an undergound storage tank can be modeled and used to generate virtual barrier functions for a real-time telerobotic system, which provides an aid to the operator for both real-time obstacle avoidance and for surface tracking. The algorithm requires that the slave's tool and every object in the waste storage tank be decomposed into convex polyhedral primitives, with the waste surface modeled by triangular prisms. Intrusion distance and extraction vectors are then derived at every time step by applying Gilbert's polyhedra distance algorithm, which has been adapted for the task. This information is then used to determine the compression and location of nonlinear virtual spring-dampers whose total force is summed and applied to the manipulator/teleoperator system. Experimental results using a PUMA 560 and a simulated waste surface validate the approach, showing that it is possible to compute the algorithm and generate smooth, realistic psuedo forces for the teleoperator system using standard VME bus hardware.

1. Introduction

Sandia National Laboratories has been developing telerobotic technology to support DOE's waste storage tank clean-up program [1]. The waste storage tanks in the DOE complex are characterized by large dimensions (up to 75 ft in diameter and 40 ft. deep), limited access (preferably through small 42 inch diameter or less risers in the top of the tanks), hostile conditions (radioactive, toxic, and potentially explosive), and contain relatively unstructured environments, i.e., the tank's contents are inadequately documented. Clean-up operations inside the tanks will be conducted by long-reach manipulators, with large mass, long flexible links, and hydraulic actuation. Their ability to respond quickly to contact with tank structures will be limited. It is therefore imperative to prevent collisions before they can cause damage to tank support structures, tank walls or to manipulator sensors and tools. This can be done by imposing virtual force fields around objects which can then deny collisions with the actual objects.

In addition, operations inside the tanks will be tedious. The waste tank surface must be scanned numerous times with sensor arrays to fully determine the material composition and geometry of the underlying waste. Next, the surface will be continually traversed to remove layers of material by scraping, pounding and sucking. Both scanning and material removal operations will require that close control of the distance to the surface of the material be maintained, even in the presence of substantial height variations.

To allow rapid prototyping and development of teleoperator systems Sandia has developed SMART

[1] This work was performed at Sandia National Laboratories and supported by the U.S. Department of Energy under contract DE-AC04-76DP00789.

MASTER

(Sequential Modular Architecture for Robotics and Teleoperation) which is currently being used at multiple sites inside and outside of Sandia [2]. SMART is a real-time telerobotic control architecture which allows the user to construct a telerobot system using independent modules describing input devices (e.g., space ball, force reflecting masters), manipulators (e.g., PUMA, Schilling Titan) and sensors (proximity sensors, force sensors). Each module represents a one-port or a two-port network element which can either perturb the force or the velocity of the teleoperator system. Recently a new OBSTACLE module for the SMART system was developed which generates virtual forces based on interactions between convex objects[3].

This paper describes some of the details of this module, and describes how it can be combined with other SMART modules to handle both obstacle avoidance and surface tracking behaviors in an underground storage tank environment. Experimental studies with a PUMA 560, a simple virtual surface, and two teleoperator input devices will then be described.

2. Approach

The first step in a virtual force feedback system is a method for generating virtual forces based on a world model of the manipulator and the environment. Using the network based passivity philosophy of SMART, boundary functions are generated around obstacles using non-linear springs and dampers, where the spring/damper combination provides zero force disturbance outside a given threshold region, and ramps up to a large force at the surface of the object. Figure 1 shows a diagram to illustrate this approach. As the gripper approaches the corner workpiece, computer generated spring-dampers push away the gripper from points of nearest contact.

The method of using nonlinear springs is similar in concept to the potential field approach [4], but substantially different in implementation. Here, only object geometry is embedded and no attraction functions exist. The springs provide a barrier function over a very limited region, and serve only to repel. The influence function is zero over the vast majority of the workspace.

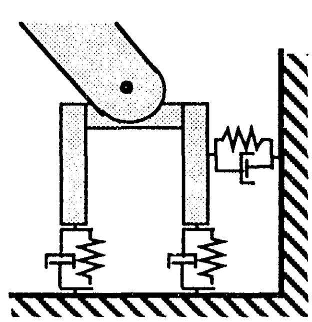

Figure 1: Use of spring-dampers for simulating contact.

To calculate the forces from a number of nonlinear springs an algorithm was developed which would compute the distances and the points of nearest contact in real-time. This algorithm requires that the manipulator and the environment are first decomposed into convex primitives.

Consider the gripper shown in Figure 2a. It can be decomposed into convex polyhedra in many ways. A decomposition consisting of the minimum number of sections using the maximum amount of overlap was chosen. The minimum number reduces the computation required, and utilizing maximal overlap reduces the likelihood of driving the system to a zero-potential at the interface of two polyhedra. Such a decomposition is shown in Fig 2b. The overlap areas are shown in grey.

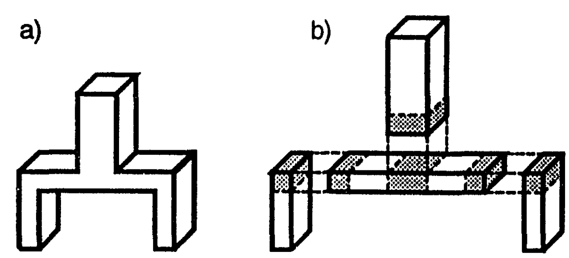

Figure 2: Convex object decomposed into primitives. a) original object; b) exploded view of decomposed object.

A waste tank surface can be decomposed into convex obstacles by using triangular prisms oriented such that a prism end triangle is aligned with three points on the surface of the waste tank. This is shown in Figure 3. Figure 3a shows a number of triangular prisms, Figure 3b shows how they are combined to represent a section of the waste tank surface.

Once every object of interest in the workspace has been decomposed into convex polyhedra and the manipulator tool has been similarly decomposed the pairwise proximity information can be determined. Namely, for each object from the set of gripper polyhedra and each object from the environments polyhedra found to be in

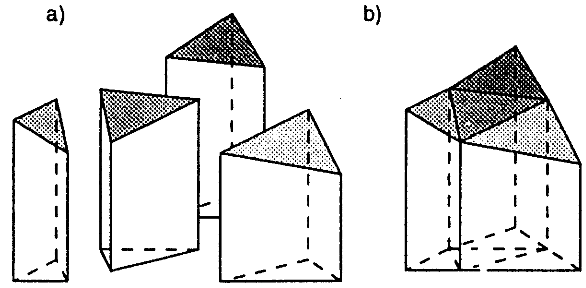

Figure 3: Modeling waste surface with polyhedra a) Individual polyhedra; b) Combined surface section.

close proximity the distance is computed. This needs to be done at a high sampling rate in order to achieve realistic stiffness and a minimal region of perturbation. After studying various possible approaches [5, 6], an algorithm based on Gilbert's algorithm [7] was implemented. The basic algorithm computes the unique distance and the non-unique vector between the convex hulls of two sets of points in Cartesian space. The convex hull of a set of points generates a polyhedron. The algorithm has two main drawbacks. It won't return anything if the two obstacles are in collision, and if two objects have parallel edges or sides the algorithm will only return a single vector with no information about the contact state [8].

In [3] a solution to these drawbacks was presented. First, the object overlap problem was solved using a reduced primitives technique, where each convex object contains a linked list of embedded polytopes ending in a point. For instance a rectangular prism might contain a plane, a line and a point as reduced primitives as shown in Figure 4a. A waste surface element can be reduced asymmetrically, using shorter and shorter prisms, since any collisions would be with the top of the surface element. This is shown in Figure 4b. If two polytopes were found to be in collision using the Gilbert algorithm then the same algorithm was applied to the reduced polytopes until no collision condition existed. This approach was shown to give a quick and simple method to find a continuous distance function and the extraction vector necessary for computing the potential force field.

The potential problem with unspecified surface interaction type was resolved by experiment. The instantaneous determination of contact type and the exact location of nearest contact vectors was found to be inconsequential. What mattered was how the system reacts to instantaneous rotations. If small shifts in orientation are opposed in two directions there is surface contact, if opposed in one there is edge contact, if opposed in no

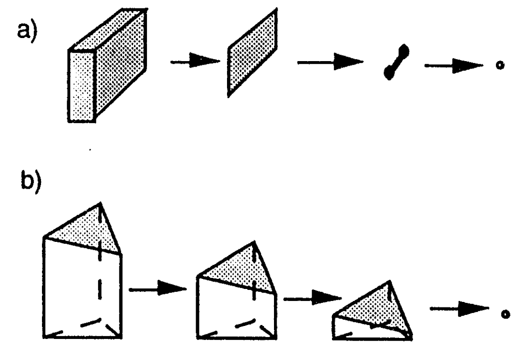

Figure 4: Convex object reductions: a) box, b) surface element..

directions there is point contact. In essence, the type of contact did not need to be determined instantaneously, but could be determined over time by gauging the net effect of reaction forces.

Consider the parallel plate interaction shown in Figure 5. The distance algorithm might select any of the four points (A, B, C or D) as being the nearest point. The virtual force algorithm generates a virtual force at the selected point, which due to the impedance algorithm implemented for the manipulator will result in that vertex rotating away from the surface. This will cause another vertex to become the "nearest" vertex, and the algorithm will next rotate that corner away. This will continue indefinitely. The extrusion vector will race all across the surface of the tool opposing any infinitesimal infraction, and the plates should remain in parallel

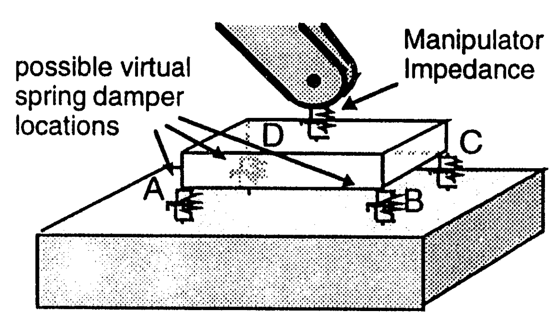

Figure 5: Parallel plate interaction.

Experimental work verified this conjecture. Using a simple world model consisting of 8 convex objects, and a gripper model consisting of 4 convex objects stable virtual contact was maintained without chattering. The manipulator tool could be aligned with the surfaces of virtual objects in the environment. At all times the motion resulting from the system was smooth and intuitive. In effect, the determination of contact type was unneccesary for developing obstacle avoidance algorithms.

To extend this capability to waste storage tanks the tank structural features must first be modelled. Next the tank surface should be scanned and decomposed into a coarse grid of triangles. For each triangle, a convex polyhedra is created. Gilbert's algorithm can then be implemented as previously described.

3. Algorithm Summary

The algorithm used for generating virtual forces is summarized below:

4. Experimental Set-up

The virtual force algorithm was previously implemented using the algorithm given above and incorporated into the OBSTACLE module. For this work, code for the OBSTACLE module was modified to allow it to accept input from an IGRIP part file describing the surface contours. As with all SMART installations to date, all code was written in C, and set to run in a multi-processor VME bus environment running under VxWorks. The OBSTACLE module itself was run on a Mercury MC860 Intel I860 based attached processor. SMART module connections were made to connect a PUMA robot , a 6 DOF force ball and a 1 DOF force reflecting input device, to the model of the virtual surface. The IGRIP model for the system is shown in Figure 6.

Virtual Force Algorithm

I. Decompose tool and the environment into overlapping convex object primitives.

A. Divide environment and tool into convex objects, using maximal overlap.

B. Approximate convex objects by polytopes.

C. For each convex object primitive determine a list of embedded sub-polytopes ending in a point.

II. Determine object interactions.

A. Use bounding boxes to eliminate object pairs which are definitely outside a delta region.

B. Apply Gilbert algorithm (using the last computed distance vectors as a starting point) to all remaining pairs of objects.

C. For any objects pairs found to be in collision use the reduced polyhedra until an intrusion distance, an interaction point, and an extrusion vector can be determined.

III. Determine force on tool.

A. For each pair within a delta distance, compute the force (f_i) as a function of intrusion distance based on non-linear spring/damper (spring constant changes from 0 at object interaction boundary to maximum value at contact point).

$$f_{TOOL} = \sum_{\substack{\text{Pairs in}\\ \text{proximity}}} f_i \qquad\qquad N_{TOOL} = \sum_{\substack{\text{Pairs in}\\ \text{proximity}}} d_i \times f_i$$

B. Apply repulsive force element along the extrusion vector at the interaction point.

C. Determine net force (f_{TOOL}) and moment (N_{TOOL}) on tool by summing up repulsive forces (f_i), and computing sum of cross products of the distances to the point of nearest contact (d_i) .

IV. Compute motion of manipulator based on impedance law.

A. Add force from world model interaction to forces from other inputs and sensors.

B. Compute delta change of position based on local dynamics and impedance parameters.

C. Go back to II.

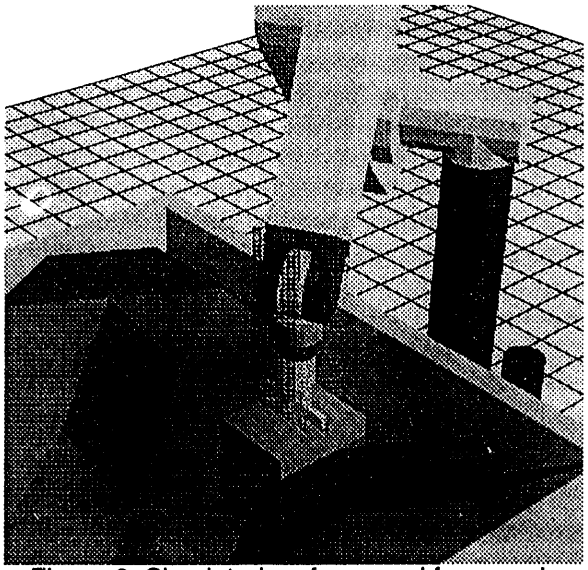

Figure 6: Simulated surface used for experiment with PUMA 560.

The SMART module connection diagram for this experiment is shown below (Fig. 7), and uses the following modules: A torque arm for 1 DOF force reflection, a MULTIPLEX module for coupling a 1 DOF device to 6 DOF with force and velocity scaling, a SPACEBALL module for enabling 6 DOF unilateral motion from a Dimension 6 Force Ball, a newly developed HYBRID_FORCE module to apply bias forces and to enable compliance along arbitrary DOF, a KBB2 module to provide filtering and reduce wave reflections, the OBSTACLE module to generate virtual object forces, a PUMA_KIN module for mapping the PUMA's kinematics from world space to joint space, a LIMITS module for imposing joint limit restrictions, a VISUAL module for displaying the system in a 3-D modeling environment (Deneb's IGRIP), and a PUMA_JOINTS module to connect to the PUMA 560 hardware.

The system was run with and without force reflection. Update rates of 100 Hz were achieved for a tool consisting of 2 convex objects, and an environment consisting of 56 convex objects. To allow surface tracking the operator used a 1 DOF input device to generate

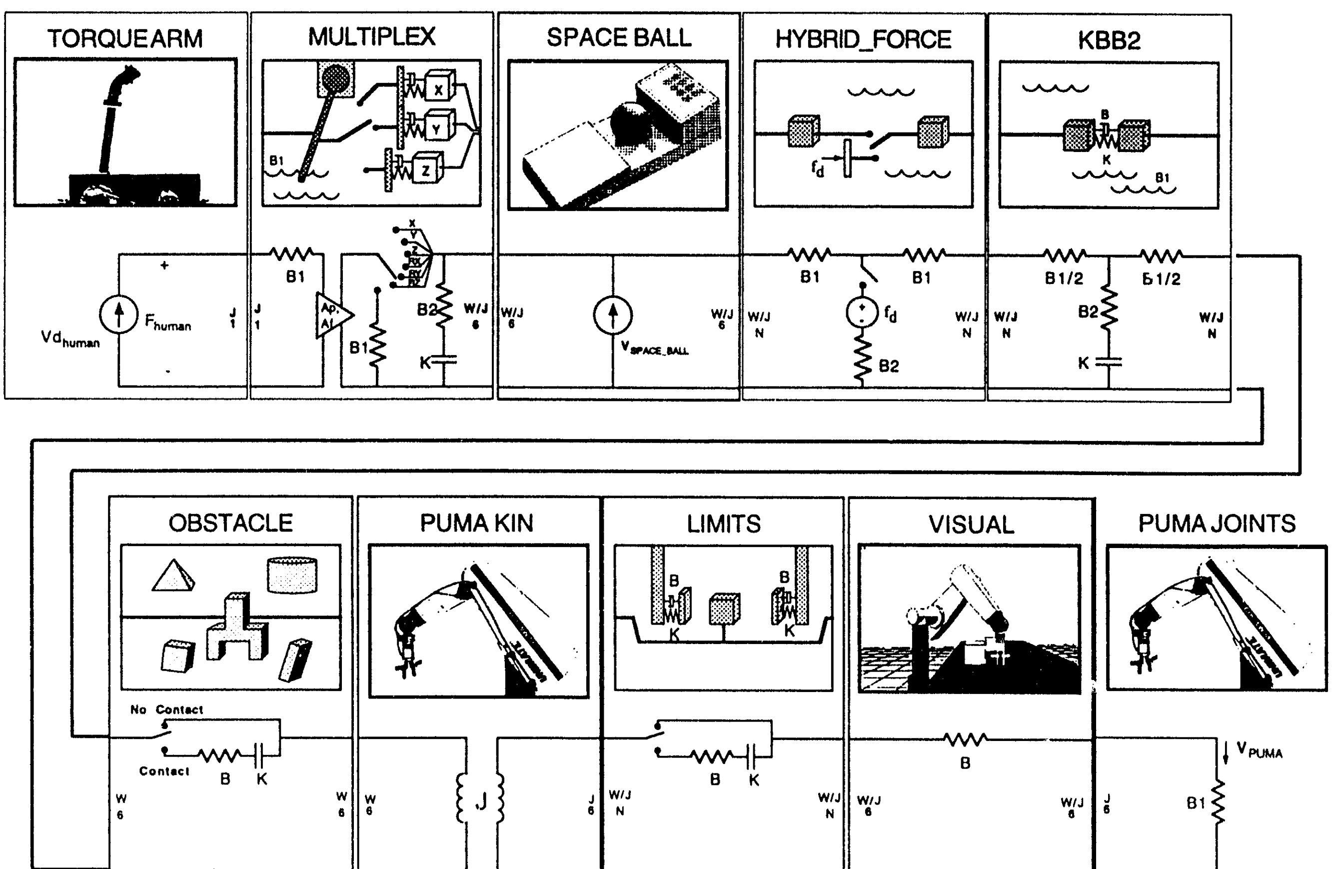

Fig. 7: SMART Modules used for experimental validation.

motion across the surface, (in either X or Y), while the HYBRID-FORCE module was used to apply a steady force to the virtual surface. To achieve tool alignment parallel to the virtual surface, the HYBRID_FORCE module was set to zero force for rotations around both X and Y. Thus because of the flat nature of the "sensor head" the head would rotate to reduce any moments generated from virtual surface contact, and thus align naturally with the surface. At all times the motion was smooth and intuitive, although the effective responsiveness as felt by the force-reflecting master was noticeably less than when only 8 world objects were modeled. As in [3], no chattering was ever observed for any type of contact.

4. Conclusions.

The experiments described in this work demonstrated that it is possible to extend the virtual force algorithm to problems arising in remediating underground storage tanks. The OBSTACLE module and the HYBRID_FORCE module could be used in conjunction to keep the manipulator a fixed distance from the object surface, while avoiding joint limits and other obstacles. A trajectory generator or the operator could then be used to traverse the X-Y surface, while the Z-distance was maintained. Experiments were conducted with a PUMA 560 robot and a virtual surface model consisting of over fifty different surface elements. Unfortunately, using the existing computing power, we could not model many more than fifty surface elements using this technique without substantially degrading the performance. If the task demanded close tracking of actual surface features rather than just a close approximation, then surface feature densities of over a 10000 elements would probably be required. This would require the development of a new SMART module dedicated to surface tracking.

Nevertheless, the existing capability is useful for maintaining sensor array distance, and for avoiding collisions with the tank surface while conducting other operations. Furthermore, we have increased by an order of magnitude the number of objects simulated and tested in a virtual force system and demonstrated that it is computationally achievable. Although requiring substantial computation every sample instant, the computation can be achieved using relatively inexpensive off the shelf computer hardware. In addition, using the SMART architecture, this capability can be applied to any robot or manipulator having a real-time trajectory modification capability.

Currently, the virtual force algorithms are being extended to the entire arm, rather than just the tool. This will allow the SMART architecture to prevent arm collisions while working in a cluttered environment, allowing full utilization of any redundancy, and allowing such advanced behaviors as virtual arm bracing. This requires modeling the manipulator's links as additional rigid body polyhedra moving with respect to the same set of fixed polyhedra in the environment. The net virtual force signal will then be mapped back into joint space using the appropriate Jacobians to achieve the desired behavior.

5. References.

[1] B. Davies, "Remediating Hazardous Waste Robotically Using a High-Level Control System and Real-Time Sensors", *Proceedings of the '93 SPIE Int. Sym. on Optical Tools for Manufacturing and Advanced Automation*, Sept. 1993, Boston, MA, (to be published).

[2] R. J. Anderson, "SMART: A Modular Architecture for Robotics and Teleoperation", *Proc. of the Fourth ISRAM*, 1992, Santa Fe, NM, pp 467-474.

[3] R. J. Anderson, "Teleoperation with Virtual Force Feedback",*Proceedings of the '93 SPIE Int. Symp. on Optical Tools for Manufacturing and Advanced Automation*, Sept. 1993, Boston, MA, (to be published).

[4] O., Khatib, , "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, Vol 5, No 1, Spring 1986.

[5] J. E. Bobrow, "A Direct Minimization Approach for Obtaining the Distance Convex Polyhedra", *The International Journal of Robotics Research*, Vol. 8, No. 3, 1989, 65-76.

[6] M. C. Lin, and Canny J. F., "A Fast Algorithm for Incremental Distance Calculation", *Proc. IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 1008-1014.

[7] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space," *IEEE Journal of R&A*, vol. 4, no. 2, pp193-203, April 1988.

[8] S. Hirai and K. Iwata, "Recognition of Contact State Based on Geometric Model", *1992 IEEE Int. Conf. on R&A*, pp 1507-1518, Nice.

DATE
FILMED
4/18/94

END