Experiments in Real-Time Collision Avoidance for Dexterous 7-DOE' Arms

Homayoun Seraji, Bruce Bon, Robert Steele
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

Abstract

The paper presents experimental results that demonstrate a new approach to real-time collision avoidance for 7-DOF arms. The collision avoidance problem is formulated and solved as a force control problem. Virtual forces opposing intrusion oj the arm into the obstacle safety zone are computed in real time. These forces arc then nullified by employing an outer feedback loop which perturbs the arm Cartesian commands jor the inner position control system. The approach is implemented arid tested on a 7-DOF RRC arm and a set oj experiments are conducted in the laboratory. These experiments demonstrate perturbations of the end-effector position and orientation, as well as the arm configuration, in order to avoid impending collisions. The approach is simple, computationally fast, requires minimal modification to the arm control system, and applies to whole-arm collision avoidance.

1 Introduction

The need for human-equivalent manipulative capabilities has motivated the development of *dexterous* arms over the past decade. These robotic arms arc kinematically similar to the human arm and have 7 joints (instead of the conventional 6 joints), which makes them kinematically redundant. This redundancy is the basis for the arm dexterity, and implies that there are infinite distinct arm configurations which yield the *same* end-effecter position and orientation. Since 1985, the Robotics Research Corporation] (RRC) has been manufacturing a family of commercially available 7-DOF arms. Similarly, robots planned for space operations, including the Ranger free-flying telerobot and the Space Station dexterous robotic system, have 7-DOF arms.

M'hilt motion control of dexterous 7-DOF arms in an obstacle-free workspace has been the subject of considerable research in recent years, the more realistic problem of collision-free motion in an obstacle field has not been investigated extensively. Maciejewski and Klein[1] describe a method for collision avoidance using the Jacobian pseudoinverse approach for redundant arm control. Khatib[2] suggests a method for real-time collision avoidance in operational space using the gradients of artificial potential fields. Wikman and Newman[3] describe a reflex control approach for on-line collision avoidance. Boddy and Taylor[4] develop a whole-arm reactive collision avoidance scheme using the configuration control methodology. Glass et al[5] describe a real-time configuration controller for a 7-DOF arm utilizing collision avoidance as the additional task. Finally, Seraji, Steele, and Ivlev[6] and Bon and Seraji[7] develop methods for sensor-based and model-based collision avoidance for a position-controlled dexterous arm based on the concept of nullification of virtual proximity forces.

In this paper, we implement the methodology described in [7] and present a set of supportive experiments conducted at JPL on collision-free motion of a dexterous 7-DOF RRC arm. The experimental results, which are the focus of this paper, demonstrate the simplicity and yet the capability of the proposed approach to real-time collision avoidance. The range of experiments cover self-collision avoidance, arm-to-base collision avoidance, and arm-to-worksite collision avoidance. The paper is organized as follows. Section 2 describes an overview of the collision avoidance strategy. Six experimental case studies highlighting different types of collision avoidance are discussed in Section 3 and experimental results are presented. Section 4 presents conclusions drawn from this work.

2 Overview of Collision Avoidance Strategy

In this section, we give an overview of the collision avoidance strategy developed earlier for dexterous 7-DOF arms. The details can be found in references [6] and [7].

Robotic arms are basically positioning devices which can carry a payload from an initial position and orientation to a target destination along a prescribed Cartesian trajectory. This arm motion is accomplished by mapping the desired Cartesian path to joint angle trajectories which are then tracked using joint servo control loops. For a kinematically redundant 7-DOF arm, such as the RRC arm, we assume that the end-effector position and orientation and the arm angle[1] (which defines the arm configuration) are under the control of a configuration control scheme, as described in [8]. In this approach, we use the configuration-cent rolled arm as the baseline system and make the necessary enhancements to this system to provide the collision avoidance capability.

For the development of the collision avoidance strategy, it is convenient to segment the 7-DOF arm into three links or arm segments as shown in Figure 1. These segments are the tool-link TW, the *forearm link* WE, and the upper- *arm link* ES, where T, W, E, and S refer to the toed tip, wrist, elbow, and shoulder, respectively. Three classes of obstacles are now defined as illustrated in Figure 1: *tool-tip obstacle, wrist obstacle*, and *elbow obstacle.* A tool-tip obstacle is one whose nearest point on the arm is on the tool-link closer to the tool-tip than a user-specified distance D. A wrist obstacle is one whose nearest point on the arm is on the tool-link further away from the tool-tip than D. An elbow obstacle is the one whose nearest point on the arm is located either on the upper-arm or on the forearm. Notice that an extended obstacle can be represented by a combination of a tool-tip obstacle, a wrist obstacle, and an elbow obstacle. In our control strategy, collision with a tool-tip obstacle is avoided by perturbing the three exld-effecter position coordinates. Collision avoidance with a wrist obstacle is achieved with perturbations of the three end-effector orientation coordinates. Finally, an elbow obstacle is avoided by perturbing the arm angle, i.e. rotating the elbow E about the shoulder-wrist axis SW without disturbing the tool frame (arm self-motion). This separation of influence of the obstacles is adopted to avoid unnecessary trajectory perturbations to both the end-effecter position and orientation, as well as to the arm angle.

For every reachable object in the workspace, the user defines a *safety zone,* which is displaced from the object surface by a user-specified *stand-off distance* d_r. The proximity of the arm to each object, d_m, is computed continuously by obstacle detection software[9] or measured by arm-mounted proximity sensors[6]. When any point on the arm enters the safety zone of an object as determined by the detection system (d_m < d,), a *virtual intrusion force* is generated in the control software and is exerted on the arm at the intrusion point. The magnitude of this force is related directly to the extent and rate of the intrusion into the safety zone, and the direction is opposing the intrusion. The intrusion force is computed from

$$F = k_i e + k_p \frac{de}{dt}$$

where $e = d_r - d_m$ denotes the extent of intrusion in the Cartesian x direction (for instance), arid k_i and k_p are user-defined constants. The physical interpretation of this expression is a force due to a fictitious spring-pltrs-damper with stiffness k_i and damping k_p located in the safety zone of each object. Figure 2 shows the block diagram of the collision avoidance system. The goal of the collision avoidance system is to perturb the nominal motion trajectory x_r in order to nullify the intrusion force F, i.e. to drive this force to zero. This is accomplished by employing an external force control loop around the internal position control system as shown in Figure 2. The virtual force F representing the intrusion is compared with the force setpoint F_r = O. When e > 0 indicating intrusion, the virtual force F *is* driven to zero by an integral controller which produces the appropriate trajectory perturbation x_f that modifies the nominal trajectory x_r as shown in Figure 2. The integral gain k is chosen such that the overall closed-loop system is stable and has a desirable performance. Observe that when the arm does not intrude any safety zone (e $\leq$ O), no corrective action is necessary (F = O, x_f = O), and the nominal trajectory x_r is executed. Finally, notice that to avoid abrupt changes as the arm enters or exits the safety zone, a nonlinear gain is introduced in Figure 2 to smooth out this transition.

3 Experimental Results

In this section, we present a set of experiments conducted at JPL on real-time collision avoidance of an RRC arm.

The experimental setup[10] consists of two RRC model K1207 7-DOF arms and control units, a VME

[1] The arm angle is defined to be the angle between the arm plane passing through the upper-arm and forearm and a reference vertical plane passing through the shoulder-wrist axis

chassis, two hand controllers, a SUN Ultra 1 workstation, and a one-third scale partial mockup of the truss structure of the Space Station. 'The VME chassis serves as the real-time controller for the al-in under study, and uses three Motorola MC68040 processors along with shared memory card and various data acquisition and communication cards. The VME-based real-time controller interfaces directly with the Multibus-based arm control unit supplied by RRC. This interface is through a two-card VME-to-Multibus adapt or set from the BIT3 Corporation. This allows a high speed bi-directional shared memory interface between the real-time controller and the arm control unit. The Servo-Level Interface software supplied by RRC enables low-level communication between the VME controller and the arm servo control loops at the rate of 400117,. The VME controller is also linked via socket communication to the SUN workstation, u'here the user interface software resides.

The arm is controlled by a configuration controller which runs on a CPU with the computational time of about 1 msec. This controller ensures that the six end-effector position and orientation coordinates, as well as the arm angle, track user-specified trajectories. The user commands seven Cartesian trajectories for the arm task coordinates using either the trajectory generation software or the hand controllers. Tbe configuration controller causes the arm to execute the commanded motion for the e~ld-effecter and the elbow in the absence of workspace obstacles. During arm motion, the obstacle detection software running on a CPU in the VME chassis continuously computes the distances between the arm segments and the workspace obstacles. The collision avoidance strategy outlined in Section 2 and shown in Figure 2 is implemented in the VME controller so that the arm coordinates deviate from their commanded trajectories as soon as an impending collision is detected.

Six experimental case studies on real-time collision avoidance are now described. In all experiments, k_i = 0.2, k_p = 0.2, k = *1.0*, and the units of length and angle are meters and radians, respectively.

3.1 Experiment 1: Position perturbation

The goal of this experiment is to demonstrate perturbation of the end-effecter position in order to avoid an impending collision. A rectangular "window" representing an opening in the Space Station truss structure is placed in the arm workspace parallel to the world frame y-z plane, and the four sides of the opening are defined as obstacles in the arm database. The e[ld-effecter of the RRC arm is commanded to move to the center of the opening initially and then execute a diamond-shaped path inside the opening. Figure 3 (dashed line) shows the Cartesian path traversed by the end-effecter when the obstacle avoidance capability is disabled. The path is designed so that the elld-effecter is moved close to the four sides of the opening with a clearance of 1 - 2 cm and then returns to the center. The experiment is then repeated with the collision avoidance enabled. The end-effector is commanded to traverse the *same* path as before. However, when the end-effector is now closer to a side than the user-s~)ecified threshold d_r = *10* cm, the collision avoidance software inhibits motion toward the side. The path traversed by the end-effecter in this case is shown by the solid line in Figure 3. It is seen that collision avoidance is accomplished by truncating the peaks of the diamond automatically to maintain the specified minimum distance to the sides, thus turning the quadrilateral diamond into an octagon.

3.2 Experiment 2: Orientation perturbation

In this experiment, we demonstrate perturbation of the end-effecter orientation to avoid collision. A large box is attached to the truss structure mockup in the laboratory. The end-effecter is initially positioned somewhat above the box, with the forearm pointing diagonally upward. The end-effecter is then commanded to traverse a trapezoidal path that will move it diagonally toward the top of the box, then parallel to the edge, and finally diagonally away from the box. Figure 4 shows the experimental results obtained in this case. With collision avoidance disabled, the wrist comes very close to the edge of the box. When the collision avoidance is enabled, the wrist will automatically rotate away from the edge of the box in order to maintain the specified clearance. This behavior is shown in Figure 4, indicating a rotation about the world frame x-axis, where the commanded rotation about the x-axis is 0.6 radians.

3.3 Experiment 3: Arm angle perturbation

The purpose of this experiment is to demonstrate perturbation of the arm configuration, i.e. the arm angle, in order to avoid an impending collision. The second RRC arm is positioned in the upright configuration to pose as a vertical obstacle during the first arm motion, and the data representing the second arm configuration is inputted to the obstacle detection software. The first arm is then commanded to execute a trajectory that brings its elbow very close to the second arm, and the

arm motion is then reversed. The dashed line in Figure 5 shows the constant commanded value of 0.493 radians for the arm angle. With collision avoidance disabled, the constant commanded arm angle is maintained and the elbow almost touches the second arm. The collision avoidance is then enabled and the *same* arm motion is commanded. The experimental results are depicted by the solid line in Figure 6. The results clearly demonstrate that the elbow now rotates away from the second arm in order to maintain a clearance of d_r = 20 cm specified by the user.

3.4 Experiment 4: Inter-arm collision avoidance

The goal of this experiment is to demonstrate inter-arm collision avoidance in our laboratory setup. The second RRC arm is configured to a specific set of joint angles and base position, and this information is inputted to the collision detection database for the first arm. The end-effecter of the first arm is commanded to move close to the second arm base pedestal. Figure 6 (dashed line) depicts the top-down view (x-y plane projection) of the end-effecter motion with the collision avoidance disabled. It is observed that the end-effecter comes to within about 2 cm of the base pedestal. With the collision avoidance enabled, the end-effecter is commanded to execute the *same* motion, and the results are shown by a solid line in Figure 6. Observe that the end-effector now follows a smooth trajectory to avoid the second arm base pedestal and maintain the stand-off distance of d_r = 10 crn to the base.

3.5 Experiment 5: Self-arm collision avoidance

In this experiment, we demonstrate self-collision avoidance for the RRC arm. The arm is positioned initially "curled in" on itself, with the end-effecter and wrist pointed in opposite directions and the end-effector pointed toward the upper-arm link away from the truss structure, The end-effecter is commanded to move toward the truss. The top-down view (x-y projection) of the path traversed by the end-effecter is shown by a dashed line in Figure 7. With collision avoidance disabled, the end-effecter comes very close to the upper-arm. The collision avoidance is now enabled and the *same* end-effecter motion is commanded. The experimental results are shown by a solid line in Figure 7. The e~ld-effecter deviates from the nominal path and makes a 'detour" to stay clear of the upper-arm and maintain the stand-off distance of d_r = 10 cm.

3.6 Experiment 6: Teleoperation

The goal of this experiment is to demonstrate the collision avoidance capability when the arm is under teleoperation. In this experiment, commands for the end-effector position and orientation and the arm angle are issued by the operator acting on two hand controllers. This is in contrast to previous experiments where the commanded motions are produced by the trajectory generation software. Experiment 1 is repeated with the user teleoperating the end-effector within the fake truss opening using the hand controller. M'bile the end-effector is away from the sides of the opening, the teleoperated commands are executed. As soon as the end-effector is commanded to move close to a side, a counter-command is generated by the collision avoidance software that nullifies the teleoperated command and inhibits motion toward the side. Thus erroneous teleoperated commands that would otherwise cause collision are corrected on-line and in real time. This is an important augmentation to teleoperation, particularly for operation in partially or completely occluded regions of the workspace.

4 Conclusions

A new approach for real-time collision avoidance of dexterous 7-DOF arms is demonstrated in this paper. This approach is based on representing the proximity of the arm to workspace obstacles by virtual forces, and servoing these forces to zero by employing an outer feedback loop around the inner arm position cent rol system.

The approach presented in this paper is equally applicable to both model-based and sensor-based collision avoidance. For the sensor-based case, the on-line distance computations are replaced by the readings provided by the proximity sensors mounted on the arm[6]. Furthermore, the proposed approach applies to both stationary and moving obstacles, since the distance computations (or measurements) are updated continuously in real time. Finally, the approach is pragmatic because: it is simple and computationally very fast, it requires minimal modification to the arm positioning system, and it applies to *whole-arm,* not just the end-effector, collision avoidance.

5 Acknowledgment

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration (Code SX Telerobotics Program).

6 References

1. A. Maciejewski and C. Klein: "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", Intern. Journ. of Robotics Research, vol. 4, no. 3, pp. 109-117, 1985.

2. 0. Khatib: "Real-time obstacle avoidance for manipulators and mobile robots", Intern. Journ. of Robotics Research, pp. 90-98, 1986.

3. T.S. Wikman and W.S. Newman: "A fast, on-line collision avoidance method for a kinematically redundant manipulator based on reflex control", Case Western Reserve University, CAISR Technical Report 91-160, 1991.

4. C.L. Boddy and J.D. Taylor: "Whole-arm reactive collision avoidance control of kinematically redundant manipulators", Proc. IEEE Intern. Conf. on Robotics and Automation, vol. 3, pp. 382-3S7, Atlanta, 1993.

5. K. Glass, R. Colbaugh, D. Lim, and H. Seraji: "Real-time collision avoidance for redundant manipulators", IEEE Trans. on Robotics and Automation, vol. 11, no. 3, pp. 448-457, 1995.

6. H. Seraji, R. Steele, and R. Ivlev: "Sc!nsox-base'd collision avoidance: Theory and experiments", Journal of Robotic Systems, vol. 13, no. 9, pp. 571-586, 1996.

7. B. Bon and H. Seraji: "On-line collision avoidance for the Ranger telerobotic flight experiment", Proc. IEEE Intern. Conf. on Robotics and Automatic, vol. 3, pp. 2041-2048, Minneapolis, 1996.

8. H. Seraji, M.K. Long, and T.S. Lee: "Motion control of 7-DOF arms: The configuration control approacb", IEEE Trans. on Robotics and Automaion, vol. 9, no. 2, pp. 125-139, 1993.

9. B. Bon and H. Seraji: "Obstacle detection for the Ranger telerobotic flight experiment", Submitted to IEEE Intern. Conf. on Robotics and Automation, 1997.

I(). D. Lim. T.S. Lee, and H. Seraji: "A real-time control system for a mobile dexterous 7-DOF arm", Proc. IEEE Intern. Conf. on Robotics and Automation, vol. 2, pp. 118 S-1 195, San Diego, 1994.

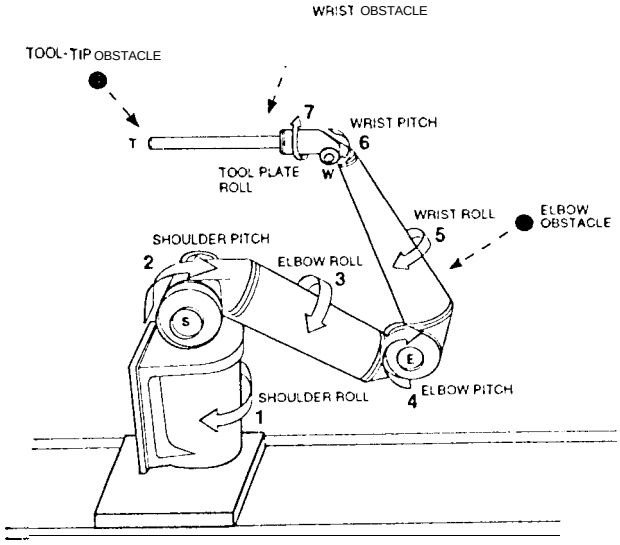

Figure 1: Dexterous 7-DOF RRC arm

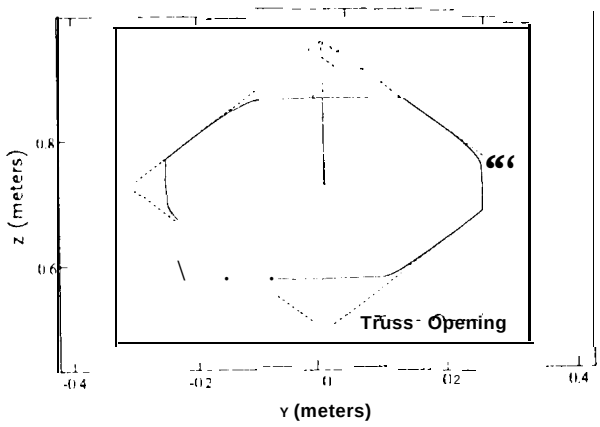

Figure 3: Position perturbation for collision avoidance

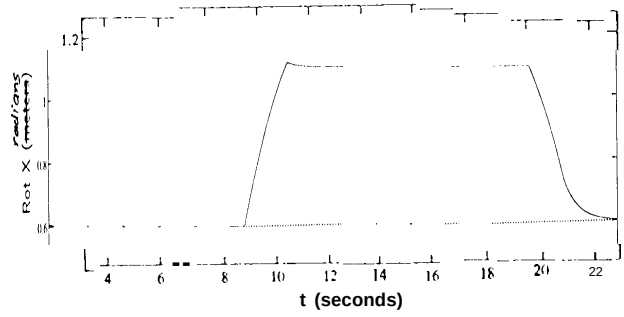

Figure 4: Orientation perturbation for collision avoidance

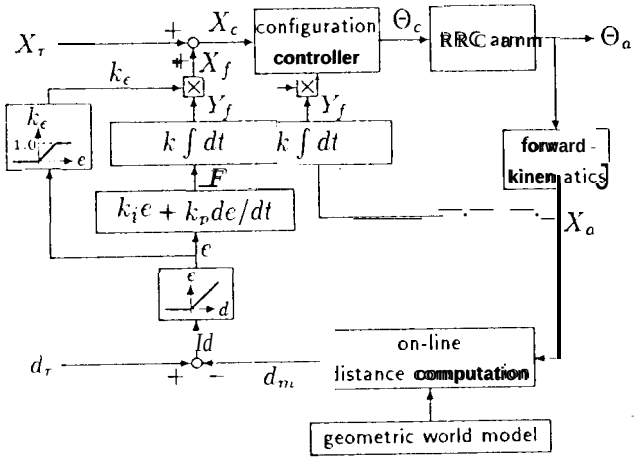

Figure 2: Control diagram for perturbing arm motion for collision avoidance

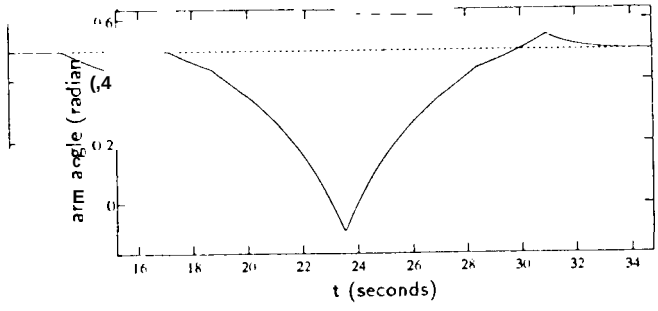

Figure 5: Arm angle perturbation for collision avoidance

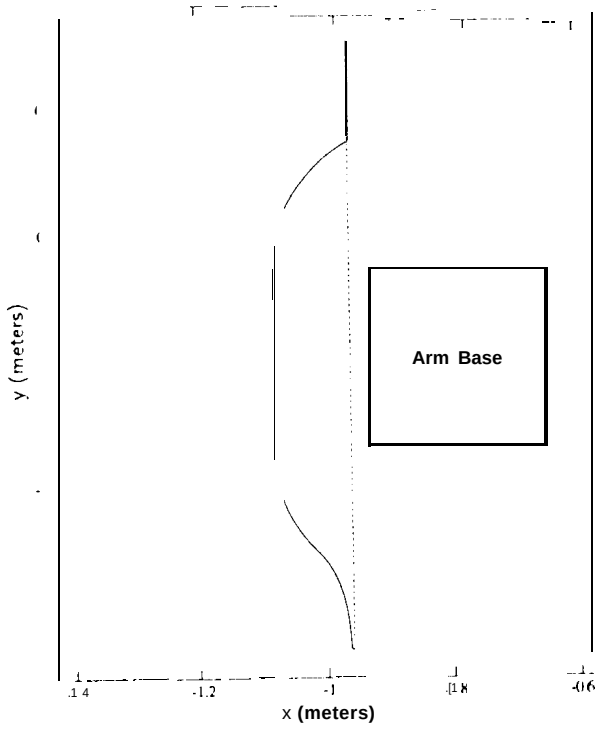

Figure 6: Inter-arm collision avoidance

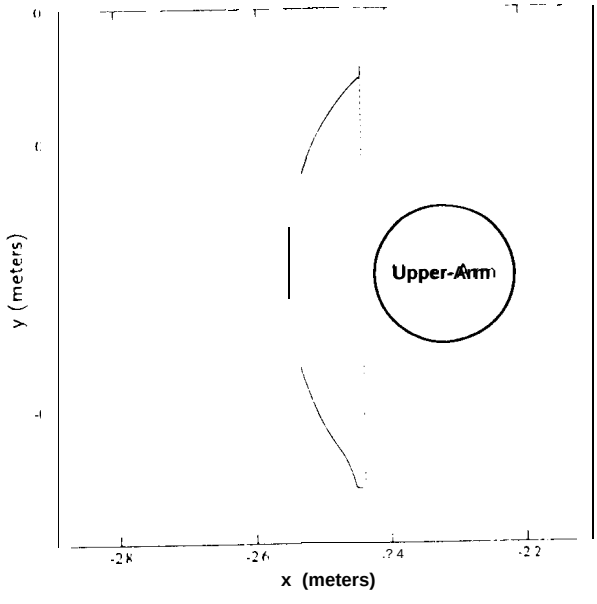

Figure i: Self-arm collision avoidance