Submitted to the 1994 IEEE International Conference on Robotics and Automation,San Diego,California, May 8-13, 1994.

WHOLE ARM OBSTACLE AVOIDANCE FOR TELEOPERATED ROBOTS

J. T. Feddema and J. L. Novak
Sandia National Laboratories*
Albuquerque, NM 87185

Abstract

This paper describes a collision avoidance system using Whole Arm Proximity (WHAP) sensors on a PUMA 560 robot arm. The capacitance-based sensors generate electric fields which can completely encompass the robot arm and detect obstacles as they approach from any direction. The directional obstacle information gathered by the WHAP sensors together with the sensor geometry and robot configuration is used to scale the commanded joint velocities of the robot. A linearized relationship between the WHAP sensor reading and the distance from the obstacle allows direct transformation of perturbations in WHAP readings to perturbations in joint velocities. The WHAP reading is used to directly reduce the component of the command input velocity along the normal axis of the sensor, allowing graceful reductions in speed as the arm approaches the obstacle. By scaling only the component of the velocity vector in the direction of the nearest obstacles, the control system restricts motion in the direction of obstacles while permitting unconstrained motion in other directions.

1 Introduction

Much of the current robotics effort at the U.S. Department of Energy is directed toward remote handling of hazardous waste. Some of this waste threatens the environment, requiring that active steps be taken to remotely stabilize, detoxify, or repackage the material. Because of the hazards involved, telerobotic systems are being developed to remotely inspect, characterize, and process waste and the containers. Sophisticated systems are being developed to permit sensor mapping of the environment, building of world models, and graphical programming of the robot system [1].

An important tenet of this waste handling scheme is that the act of processing the waste must create no additional hazards and do no damage to the containment structure. Collision-free paths for robot manipulators are calculated using world models generated from sensor information. To provide the required degree of safety, however, it must be assumed that the information contained in these world models is inaccurate due to errors such as sensor noise or incorrect assumptions. This mandates the use of an independent collision avoidance system incorporating both sensors for detecting the approach of obstacles, and a control system that gracefully overrides commands which would result in collisions.

Considerable effort has been put into path planning in cluttered environments, but most of the work assumes a perfect world model and the collision avoidance is performed based on the world model [2, 3, 4, 5, 6]. There are relatively few path planners that are implemented on a real robot manipulator and are capable of handling modeling uncertainties via online sensing [7, 8, 9].

An excellent real-time technique for preventing collisions, especially along the length of a robot link, involves the use of proximity sensors spaced over the robot surface. Cheung and Lumelsky [7, 8] use a system of infrared emitters and detectors on flexible printed circuit boards. These sensors are directional and can provide information useful in mapping obstacles. However, large numbers of these devices would be required to completely protect the surface of a robot arm. A capacitive sensor, termed the Capaciflector, was developed at NASA for collision avoidance [10]. This sensor design uses active electrical guarding techniques to allow sensors to measure the capacitance between the sensor and electrical ground. Merritt Systems, Inc. has developed a multi-mode sensor architecture that combines infrared and ultrasonic sensors for collision avoidance [9]. In all these systems, information from the sensors is used in a variety of control and planning schemes to permit collision-free motion in the presence of obstacles. Additionally, since it is desirable to continue purposeful motion in the presence of obstacles, these sensor systems can deliver spatially-resolved proximity data that reflects the distance to the obstacle, as well as the location on the robot surface and corresponding robot surface normal. This vector information can then be used to modify trajectories to permit path replanning and (if possible) subsequent progress toward the final destination.

This paper expands upon previous work [11, 12, 13] with the WHAP sensor and collision avoidance system. First, the principle of operation and advantages of the WHAP sensors are reviewed. Second, the hardware implementation of the WHAP sensors on the PUMA is described. Third, we describe a new control algorithm that is tailored for collision avoidance of multiple obstacles while in a teleoperated mode of control. Finally, experimental results are presented using the PUMA 560 robot.

*Work supported by the U.S. Department of Energy at Sandia National Laboratories under Contract DE-AC0476DP00789.

2 WHAP Principle of Operation

The WHAP sensor described in this paper measures a mutual capacitance between two conductors patterned on the sensor substrate itself [13]. The two electrodes can be designed with precisely defined geometries to generate a spatially-resolved fringing electric field. Conductive or dielectric obstacles disturb the electric field through a shielding effect and alter the measured mutual capacitance. Since the electric field between the two plates is well-defined by the conductor arrangement, it is possible to reconstruct the obstacle surface and range more accurately. Because this configuration does not measure displacement currents to electrical ground, stray capacitances to ground (such as between the sensor and a metal robot surface) do not affect the measurement. No active shielding is required, and this type of capacitive sensor is insensitive to the electrical potential of the obstacle.

A schematic 2D model of the WHAP sensor is given in Figure 1. One electrode is driven by an oscillator, while the other is connected to an amplifier for sensing capacitor charge. For a fixed oscillator drive voltage, this charge output signal, V_c is proportional to the sensor capacitance, C_{12}. The sensor capacitance is altered by the presence of obstacles within the electric field. More detail of the WHAP sensor operation is given in [13].

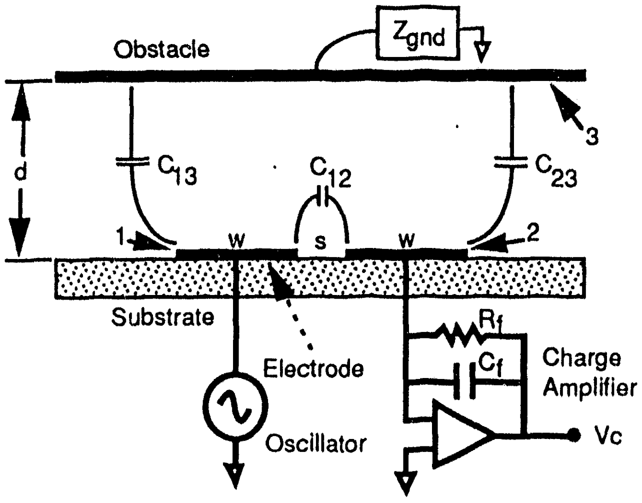

Figure 1. Schematic 2-D Model of the WHAP Sensor.

3 Experimental Methods

The WHAP sensor "skin" for the PUMA was fabricated using a three layer printed circuit board. Each sensor consists of a 20mm diameter disc surrounded by a 30 mm i.d. ring that is 4 mm wide. A total of 49 of these sensors has been mounted on the robot. Twenty-two sensors are mounted on the third link, and 27 are mounted on the large planar region of the second link (Figure 2). The sensors were positioned to provide enough overlap of the sensing fields to provide redundant information. Center-to-center distances ranged from 50 to 120 mm, depending on the location on the robot arm.

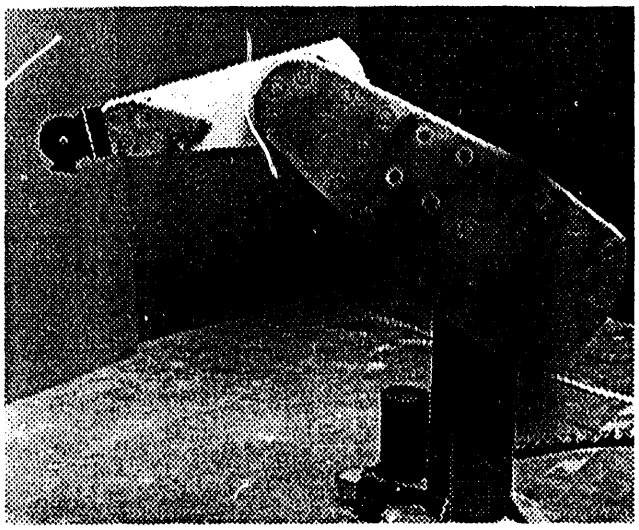

Figure 2. WHAP Sensor Skin on PUMA Robot. The WHAP sensors are visible as disks surrounded by light-colored rings.

The bottom layer of the sensor circuit board (nearest the robot) was used for power, drive signals, and the charge amplifier components, while the middle layer was grounded for isolation. The top layer was patterned to provide the disk and ring sensor configuration discussed above. The sensor receiving plates (right electrode in Figure 1) from two adjacent sensors were electrically connected to a single charge amplifier input. By driving the sensors at two different frequencies, the single charge amplifier output lead can be used for both sensors. This configuration can be extended to further reduce the number of signal lines by increasing the number of unique operating frequencies. Recent changes to the sensor design using this technique allow WHAP sensors to be fabricated using two-layer boards, significantly reducing complexity and cost.

An important feature of this configuration is that only the charge amplifier integrated circuits and associated passive components and connectors need to be on the sensor board (Figure 1). As mentioned earlier, no driven electrical guard electrode is necessary in this design since the charge amplifier configuration is insensitive to parasitic capacitances to ground. Synchronous detection circuitry was used to measure the amplitude of the corresponding frequency component in the charge amplifier output. These circuits provided an extremely low noise signal output by phase- and frequency-locking onto the input drive signal. Because of this, no shielding was required on the signal leads. Only the cables leading to the drive electrodes must be shielded to minimize stray coupling to the charge amplifier inputs. This electrical configuration significantly reduces the number of components that must be in a hazardous environment, facilitating the environmental and radiation hardening of the sensors. The use of unshielded cabling also significantly reduces the weight of the system.

Sensor noise levels of 2-5 mVrms were measured during motion of the robot. It is important to note that the analog signals are carried approximately 8 meters on unshielded ribbon cable immediately adjacent to the robot motors which generate large amounts of electromagnetic interference. The synchronous detection circuitry rejected most of the interference and allowed low-noise measurements to be made. Other techniques, including rectification and oscillator frequency variation, were found to inadequately reject the interference. The range of the sensors was approximately 330 mm (13"), using a flat metal plate as the obstacle. In general, the sensor output as a function of the distance from the obstacle is a non-linear curve. Figure 3 shows the WHAP sensor output as a function of distance and offset from a 60 mm diameter steel pipe. The sensors have been placed on the arm such that they have overlapping fields. When the pipe is between two sensors that are 50mm apart, both sensors will have a response similar to the "25mm off center" curve.

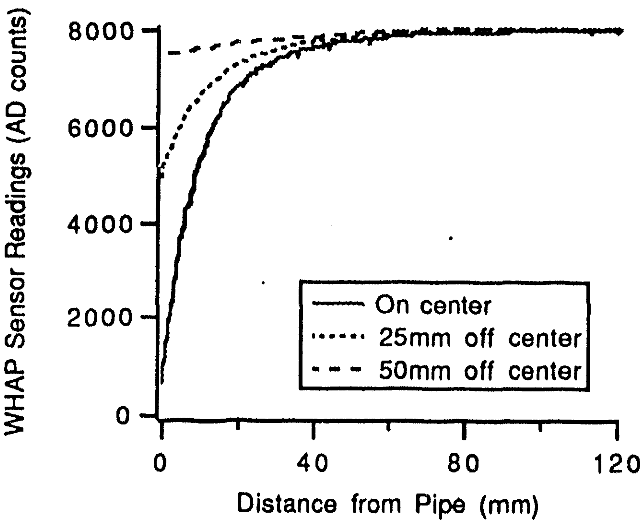

Figure 3. WHAP Output Versus Distance.

4 Control Algorithm

A general robot control algorithm was developed which collects readings from a large number of WHAP sensors distributed over a general 'n' degree of freedom robot arm and keeps the arm from colliding with obstacles in the robot workspace. Past work on robot control algorithms for whole arm obstacle avoidance can be divided into pre-planned motions [7, 8] and teleoperated motions [9, 6, 14]. Algorithms for pre-planned motions assume the final destination of the robot end-effector is known, and the robot path is dynamically altered using whole arm sensor data. In teleoperated systems, the input command to the robot controller is a velocity of the end-effector in either world or tool frame. The operator directs the end-effector towards a final destination rather than giving it a final destination. The work presented here concentrates on the teleoperated case.

The end-effector velocity $\dot{\mathbf{x}}_e$ is related to the joint velocities $\dot{\mathbf{q}}$ by the differential equation

$$\dot{\mathbf{x}}_e = \mathbf{J}_e \dot{\mathbf{q}} \tag{1}$$

where $\dot{\mathbf{x}}_e$ is a 6×1 column vector, $\dot{\mathbf{q}}$ is an $n \times 1$ column vector, and $\mathbf{J}_e$ is the $6 \times n$ manipulator Jacobian matrix either with respect to world or end-effector coordinate frames. As described in [9, 6], a similar differential relationship may be defined that relates a "specified obstacle avoidance point velocity", $\dot{\mathbf{x}}_o$, to the joint velocities

$$\dot{\mathbf{x}}_o = \mathbf{J}_o \dot{\mathbf{q}} \tag{2}$$

where $\mathbf{J}_o$ is the "obstacle avoidance point Jacobian." For a redundant robot arm, a least squares solution which first satisfies the end-effector velocity and then uses the redundant degrees of freedom to satisfy the obstacle avoidance velocity is given by [6]

$$\dot{\mathbf{q}} = \mathbf{J}_e^+ \dot{\mathbf{x}}_e + [\mathbf{J}_o(\mathbf{I} - \mathbf{J}_e^+ \mathbf{J}_e)]^+ (\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^+ \dot{\mathbf{x}}_e) \tag{3}$$

where the pseudo-inverse is $\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$. This equation can be extented to include multiple obstacles; however, the computational burden of each additional pseudo-inverse makes it difficult to compute in real-time. Also for most waste remediation applications, top priority should be given to the obstacle avoidance rather than following the desired end-effector velocity. The least squares solution can force the arm into the obstacle. This is especially true for non-redundant arms.

The approach we have taken is to use distance measurements from the WHAP sensors to filter the magnitude of the reference velocity from the operator (see Figure 4). We assume that the WHAP sensors only give us a distance measure of object's perpendicular to the sensor. When the arm is at a distance d_i greater than d_{max} from the obstacle, the motion of the arm should be unconstrained. The reference perpendicular velocity at each sensor, $\dot{z}_{ir}$, is given by

$$\dot{z}_{ir} = \mathbf{J}_i \; \dot{\mathbf{q}}_r = \mathbf{J}_i \; \mathbf{J}_e^+ \; \dot{\mathbf{x}}_e \tag{4}$$

where the reference joint velocity $\dot{\mathbf{q}}_r$ is a determined from the end-effector velocity $\dot{\mathbf{x}}_e$ as specified by the operator. For a six degree of freedom robot, the pseudo-inverse is simply the inverse $\mathbf{J}_e^{-1}$. The sensor Jacobian $\mathbf{J}_i$ is a $1 \times n$ matrix which depends on the sensor's geometric position on the arm and the current joint configuration. Considering m sensors distributed over the entire arm, this can be written as

$$\dot{\mathbf{z}}_r = \mathbf{J} \; \dot{\mathbf{q}}_r = \mathbf{J} \; \mathbf{J}_e^+ \; \dot{\mathbf{x}}_e \tag{5}$$

where $\mathbf{J}$ is $m \times n$ and $m \gg n$.

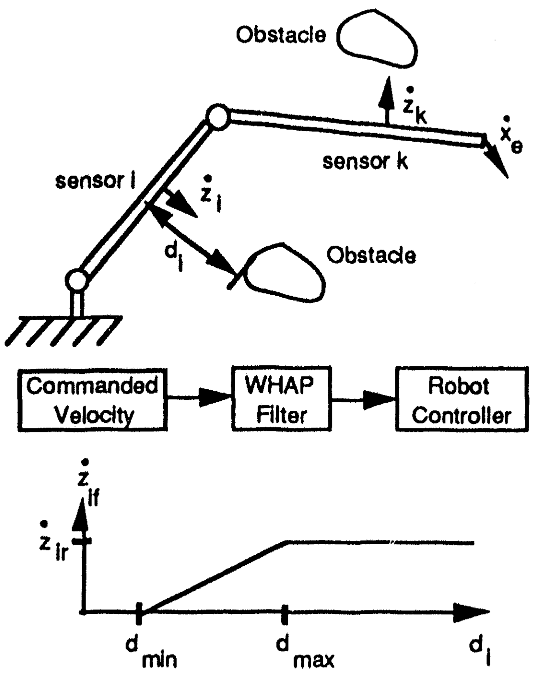

Figure 4. Scaling Of Filtered Perpendicular Velocities $\dot{z}_{if}$ As Arm Approaches Obstacle .

When a sensor i detects an obstacle between d_{max} and d_{min}, the perpendicular velocity $\dot{z}_i$ should be scaled linearly to zero, thus disabling motion in the direction of the obstacle when the distance equals d_{min}. Therefore, the filtered velocity of the arm perpendicular to the obstacle is ·

$$\dot{z}_{if} = \frac{\dot{z}_{ir}}{d_{max} - d_{min}}(d_i - d_{min}) \qquad (6)$$

for $d_{min} \leq d_i \leq d_{max}$. Relating this back to the robot joint angles,

$$\dot{\mathbf{z}}_f \geq \mathbf{J}\,\dot{\mathbf{q}}_f \qquad (7)$$

where $\dot{\mathbf{q}}_f$ is the filtered robot joint velocity. This equation limits the robot joint velocities such that the resulting perpendicular velocities at each sensor are less than or equal to the value determined from Equation (6).

There are many solutions to this problem including the trival solution $\dot{\mathbf{q}}_f = 0$. For our circumstances, we are interested in the solution which provides the largest bounds for $\dot{\mathbf{q}}$. One possible solution would be

$$\begin{aligned} \dot{\mathbf{q}}_{jf} &= \min_{i=1,\ldots,m}\left(\frac{\dot{\mathbf{z}}_{if}}{w_i \mathbf{J}_{ij}} \quad | \quad \mathbf{J}_{ij} > 0\right) \\ \dot{\mathbf{q}}_{jf} &= -\min_{i=1,\ldots,m}\left(\frac{-\dot{\mathbf{z}}_{if}}{w_i \mathbf{J}_{ij}} \quad | \quad \mathbf{J}_{ij} < 0\right) \qquad (8) \end{aligned}$$

for $j = 1, \ldots, n$. The symbol | denotes an "if" condition, and w_i is the number of nonzero elements in row i of $\mathbf{J}$. The filtered perpendicular velocity at each sensor, $\dot{\mathbf{z}}_{if}$, is always greater than or equal to zero. Equation (8) results in a rather conservative solution.

As shown in the previous section, the WHAP sensor's output is related to the distance from the object by a nonlinear function over the entire range of the sensor. If the control is to respond linearly to the distance from the object, the expression for the sensor reading c_i must be inverted in order to determine the distance d_i. The WHAP sensors have been designed so that the nonlinear curve in the sensor expression (the "knee" in Figure 3) occurs at distances less than d_{min}. We will assume that between d_{min} and d_{max} the sensor reacts linearly as given by

$$c_i = \frac{c_{imax} - c_{imin}}{d_{max} - d_{min}}(d_i - d_{min}) + c_{imin} \qquad (9)$$

The filtered velocity can then be written as a function of the sensor readings.

$$\dot{z}_{if} = \frac{\dot{z}_{ic}}{c_{imax} - c_{imin}}(c_i - c_{imin}) \qquad (10)$$

At this point, we have a two step process. First, we solve for the maximum perpendicular velocities using Equation (10), and then we solve for the maximum and minimum joint velocities using Equation (8). These two equations can be combined into a single equation. Equation (7) can be written as

$$\mathbf{e}_{\dot{z}} \leq \mathbf{J}\,\mathbf{e}_{\dot{q}} \qquad (11)$$

where $\mathbf{e}_{\dot{z}} = \dot{\mathbf{z}}_r - \dot{\mathbf{z}}_f$, and $\mathbf{e}_{\dot{q}} = \dot{\mathbf{q}}_r - \dot{\mathbf{q}}_f$. Using Equation (10),

$$\mathbf{K}\,\mathbf{e}_c \;\leq\; \mathbf{J}\,\mathbf{e}_{\dot{q}} \qquad (12)$$

where $\mathbf{K}$ is a diagonal $m \times m$ matrix

$$\mathbf{K}_{ii} = \frac{\dot{\mathbf{z}}_r}{\mathbf{c}_{imax} - \mathbf{c}_{imin}} \qquad (13)$$

and $\mathbf{e}_c = \mathbf{c}_{max} - \mathbf{c}$. It is important to note that $\mathbf{c}_{max}$ is the reference WHAP reading when the obstacle is at a distance equal to d_{max}. Therefore, Equation (12) shows that WHAP sensor errors can be directly mapped to errors in the joint velocities. These joint velocity errors can in turn be used to scale the maximum robot joint velocities.

There are an infinite number of solutions $\mathbf{e}_{\dot{q}}$ to (12). We are interested in a solution which closely bounds $\mathbf{e}_c$. Also, for teleoperated operations it is advantageous to choose a solution which considers each joint independently since the controller can not depend on the operator to direct the joints in such a fashion that one joint cancels the motion of another joint. Therefore, a simple solution to (12) is

$$\begin{aligned} \mathbf{e}_{j\dot{q}pos} &= \max_{i=1,\ldots,m}\left(\frac{\mathbf{K}_{ii}\,\mathbf{e}_{ic}}{\mathbf{J}_{ij}} \quad | \quad \mathbf{J}_{ij} > 0\right) \\ \mathbf{e}_{j\dot{q}neg} &= \max_{i=1,\ldots,m}\left(\frac{-\mathbf{K}_{ii}\,\mathbf{e}_{ic}}{\mathbf{J}_{ij}} \quad | \quad \mathbf{J}_{ij} < 0\right) \qquad (14) \end{aligned}$$

for $j = 1, ..., n$. Note that $\mathbf{e}_c$ and $\mathbf{K}_{ii}$ are always greater or equal to zero. As we can see the final solution has maximum positive and negative error values for each joint. This allows the controller to limit motion in one direction and still enable motion in the other direction.

The WHAP filter is defined by looking at the boundary conditions. When no obstacles are present, $\mathbf{e}_c = 0$, which implies that both $\mathbf{e}_{j\dot{q}pos}$ and $\mathbf{e}_{j\dot{q}neg}$ are zero and $\dot{\mathbf{q}}_f = \dot{\mathbf{q}}_r$. Since we want the robot to move freely when no obstacles are present, this boundary condition corresponds to a WHAP filter gain of unity. When an obstacle is a distance d_{min} from sensor i, $\mathbf{e}_{ic} = \mathbf{c}_{imax} - \mathbf{c}_{imin}$, $\mathbf{e}_{j\dot{q}pos} = \dot{\mathbf{q}}_r$, and $\dot{\mathbf{q}}_f = 0$. Since we want the robot to be constrained in the direction of the obstacle, this boundary condition corresponds to a WHAP filter gain of zero. Between these values is a linear function.

One last consideration is the computation of $\mathbf{J}$. In general, $\mathbf{J}$ is a triangular matrix. This results from the fact that the velocities at sensor locations on link l are not affected by the motions of joints greater than l (neglecting dynamic Coriolis dynamic effects). Also, we can take advantage of the fact that the Jacobian matrix $\mathbf{J}$ may be decomposed into two matrices. For example, let us consider k sensors on link l. These sensors represent a subset of the m sensors defined above. The differential relationship may be written as

$$\dot{\mathbf{z}}_{cl} = \mathbf{J}_{cl}\ \mathbf{J}_{ql}\ \dot{\mathbf{q}}_{ql} \qquad (15)$$

where $\dot{\mathbf{z}}_{cl}$ is a k dimensional row vector, $\mathbf{J}_{cl}$ is a $k \times 6$ matrix representing the transformation from link coordinate frame l to the z direction of each sensor, and $\mathbf{J}_{ql}$ is a $6 \times l$ matrix representing the transformation from the previous l joints to the coordinate frame l. Considering only the z velocity component, the ith row of $\mathbf{J}_{cl}$ is [15]

$$\begin{aligned}(\mathbf{J}_{cl})_i \ = \ [\ \mathbf{a}_{ix} \quad \mathbf{a}_{iy} \quad \mathbf{a}_{iz} \quad (\mathbf{p}_i \times \mathbf{a}_i)_x \\ (\mathbf{p}_i \times \mathbf{a}_i)_y \quad (\mathbf{p}_i \times \mathbf{a}_i)_z\ ] \qquad (16)\end{aligned}$$

where $\mathbf{a}_i$ and $\mathbf{p}_i$ are the approach and position vectors of each sensor with respect to the lth coordinate frame. Since the sensors are fixed with respect to the lth coordinate frame, $\mathbf{J}_{cl}$ is a constant matrix. The Jacobian $\mathbf{J}_{ql}$ is the manipulator Jacobian with respect to the lth coordinate frame [15]. This matrix is dependent on the previous l joint angles and must be updated in real-time.

Therefore, the real-time algorithm is to first update the manipulator Jacobian matrices $\mathbf{J}_{ql}$ for links $l = 1, ..., n$. Then starting at joint 1, calculate $\mathbf{J} = \mathbf{J}_{cl}\ \mathbf{J}_{ql}$ and (14) for each joint. Assuming k sensors per link and ignoring the updating of the manipulator Jacobian, this algorithm requires $3kn(n+1)$ multiplies, $2.5kn(n+1)$ additions, $0.5kn(n+1)$ divides, and $0.5kn(n+1)$ comparisons. The true advantage of this algorithm is that it is easily parallelized because of the maximum operation. Each link can be computed separately on different joint level CPUs with the final maximum being computed on the robot control CPU.

From an operator stand point, the filtering scheme provides several benefits. First, it does not restrict motion in the region between d_{min} and d_{max}. It only reduces the arm's speed as it approaches an obstacle. Second, this scheme does not provide repulsive forces which would move the arm away from the obstacle when the commanded input velocity is zero. Our experience has been that the operator does not feel in total control when the built up repulsive force causes the arm to move to an equilibrium state after releasing the input device. Third, this scheme does not require computationally expensive pseudo-inverse equations as required by some repulsive least squares control schemes. This is especially important as the number of sensors becomes increasingly large and there are potentially several obstacles which the arm could contact simultaneously. Finally, obstacles on opposite sides of the arm will not cancel each other as would occur with a least squares approach.

5 Application to a PUMA 560

The control system for a teleoperated PUMA robot is shown in Figure 5. The operator gives a velocity command $\dot{\mathbf{x}}_e$, in the world or tool coordinates, using a six degree of freedom force input device, called the Force-Torque Ball (FTB). This command is transformed into the joint space command $\dot{\mathbf{q}}_r$ which is scaled by the WHAP filter prior to being communicated to the robot controller as command velocity $\dot{\mathbf{q}}_f$.

The WHAP filter reduces the desired velocity based upon the direction and magnitude of the joint error vector $\mathbf{e}_{\dot{q}}$. This error vector iş calculated using the Jacobian transformation in Equation (15) and the column maximization technique in Equation (14). The Jacobian transformation computes a linearized error in robot joint space which increases as the arm approaches the obstacle. The column maximization operation makes the decision as to which sensor is providing the most important information during each controller cycle. Because the maximization algorithm is executed each robot control cycle, the "important" sensor can change smoothly as the arm is moved past an obstacle. This simple method of selecting a sensor works well because the overlapping electric fields of the WHAP sensors provide simultaneous sensing of an obstacle by a number of sensors. This seamless transition results in smooth motion past an obstacle, in spite of gross motion of the robot with respect to the obstacle.

This control algorithm has been implemented on a VME-bus computing platform. The WHAP sensor data is updated every 10ms by a set of boards that perform the synchronous detection and analog-to-digital conversion. A 68030 CPU board running VxWorks reads the sensor data over the bus and performs the Jacobian transformations from sensor space to robot joint space and the column maximization operation. A linked list of sensor transformations is maintained for easy addition of sensors. A second CPU board reads the FTB input through a serial port, translates the input from world space to robot joint space, and uses the WHAP perturbation from the first CPU board to scale the desired velocity command to produce the command velocity . The PUMA controller is running a SLAVE program which communicates serially with the VME system and exchanges the desired and actual robot joint positions every 28 ms.

Figure 6 shows the error in WHAP sensor reading and the associated WHAP filter value as the arm approaches a 60 mm diameter pipe. The breakpoints for c_{min} and c_{max} were chosen such that the WHAP filter begins to decrease at a distance of about 100mm, and becomes zero (stopping all motion in the direction of the obstacle) at about 30mm. By comparing Figures 3 and 6, the reader can see that we are using the approximately linear region of the WHAP sensor output.

Note that when a joint velocity is reduced to zero by the WHAP filter, the resulting end-effector velocity will be different than that specified by the operator. A similar error would also occur using a repulsive pseudo-inverse technique on a 6 degree of freedom robot. This difference occurs due to having an inadequate number of degrees of freedom to both satisfy the desired end-effector velocity and move away from the obstacle.

For a redundant robot, the above control structure could be modified to reduce and possibly eliminate the error in the end-effector velocity. The inverse manipulator Jacobian and the WHAP filter would be replace with a weighted pseudo-inverse of the manipulator Jacobian

$$\begin{aligned} \dot{\mathbf{q}}_c &= \mathbf{W}(\mathbf{J}_e\mathbf{W})^{+}\dot{\mathbf{x}}_e \\ &= \mathbf{W}(\mathbf{J}_e\mathbf{W})^T[(\mathbf{J}_e\mathbf{W})(\mathbf{J}_e\mathbf{W})^T]^{-1}\dot{\mathbf{x}}_e \quad (17) \end{aligned}$$

where the diagonal $n \times n$ matrix $\mathbf{W}$ elements are the WHAP filter values for each joint. When the WHAP filter values are all unity, (17) will result in an unconstrained least squares solution. When a WHAP filter value of joint j is a value between zero and one, the joint velocity $\dot{\mathbf{q}}_j$ will be reduced. When the WHAP filter value of joint j is zero, column j of the manipulator Jacobian will be zero and the resulting joint velocity $\dot{\mathbf{q}}_j$ will be zero.

6 Conclusion

Both the capacitive sensor and the obstacle avoidance control algorithm are important contributions towards safe and reliable robotic systems. The capacitive sensor offers many advantages over sonar and infrared systems. The distribution of the electric field allows broad coverage of a robot without a large number of sensors. The sensor hardware is reliable and inexpensive, and it may be fabricated using flexible printed circuit boards. Capacitance measurements are instantaneous and insensitive to color, texture, and the surface of the approaching obstacle. The obstacle avoidance control algorithm is formulated for any number of degrees of freedom of the robotic arm and is less computationally intensive than past obstacle avoidance control algorithms. Based on the WHAP sensors readings and the sensors' geometric positions on the robot arm, the control algorithm modifies the operator's velocity input so that the robot arm comes to a smooth stop approximately 30mm from the obstacle. This system is being incorporated into a virtual environment [11] where the the control algorithm keeps the operator from driving into obstacle while a graphical animation of the robot arm and the sensors is used to visually notify the operator of impending collisions.

Acknowledgements

We wish to thank Bob Waldschmidt for fabricating the prototype sensors and analog signal conditioning boards, Jon Bryan for designing and fabricating the data acquisition system, and Jim Akins for keeping the workstation and robot operating.

References

[1] B. R. Davies, "Remediating hazardous waste robotically using a high-level control systems and real-time sensors," *Proc. SPIE Int. Symp. on Optical Tools for Mftg. and Advanced Automation, Telemanipulator Technology Conference*, Boston, MA, September 1993.

[2] F.T. Cheng, et. al., "Obstacle avoidance for redundant manipulators using the compact QP method," *Proc. IEEE Int. Conf. on Robotics and Automation,*, Vol. 3, pp. 262-269, Atlanta, GA, 1993.

[3] R.V. Mayorga, et. al., "A fast approach for the path planning of telerobotic manipulators," *Proc. IEEE Int. Conf. on Robotics and Automation,*, Vol. 2, pp. 289-294, Atlanta, GA, 1993.

[4] S. Seereeram, and J.T. Wen, "A global approach To path planning for redundant manipulators," *Proc. IEEE Int. Conf. on Robotics and Automation,*, Vol. 2, pp. 283-288, Atlanta, GA, 1993.

[5] P. Wenger, P. Chedmail, and F. Reynier, "A global analysis of following trajectories by redundant manipulators in the presence of obstacles," *Proc. IEEE Int. Conf. on Robotics and Automation,*, Vol. 3, pp. 901-906, Atlanta, GA, 1993.

[6] A. A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The Int. J. of Robotics Research*, Vol. 4, No. 3, pp. 109-117, Fall 1985.

[7] E. Cheung and V. Lumelsky, "Development of sensitive skin for a 3D robot arm operating in an uncertain environment," *Proc. 1989 IEEE Int. Conf. on Robotics and Automation*, pp. 1056-1061, Scottsdale, AZ, 1989, .

[8] V. Lumelsky and E. Cheung, "Towards safe real-time robot teloperation: Automatic whole-sensitive arm collision avoidance frees the operator for global control," *Proc. 1991 IEEE Int. Conf. on Robotics and Automation*, pp. 797-802, Sacramento, California, 1991.

[9] D. Rosinski and D. Wegerif, "Sensor-based whole-arm obstacle avoidance for kinematically redundant robots," *Proc. SPIE OE/Technology '92*, Boston, MA, 1992.

[10] J. M. Vranish and D. S. Chauhan, "Tri-mode collision avoidance skin for robot arms in space," *Proc. Third Int. Symposium on Robotics and Manufacturing: Research, Education, and Applications*, Burnaby, British Columbia, CANADA, July 18-20, 1990, pp. 189-195.

[11] J.L. Novak, et. al., "Collision avoidance during teleoperation using whole arm proximity sensors coupled to a virtual environment," *Proc. SPIE Int. Symposium on Optical Tools for Manufacturing and Advanced Automation: Telemanipulator Technology*, Boston, MA, September 1993.

[12] J. L. Novak, and J. T. Feddema, "Whole arm collision avoidance using capacitive sensors on a PUMA 560," *Proc. Fifth Topical Mtg. on Robots and Remote Systems*, ANS, 1993.

[13] J.L. Novak and J.T. Feddema, "A capacitance-based proximity sensor for whole arm obstacle avoidance," *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, pp. 1307-1314, Nice, France, 1992.

[14] K. Glass, et. al., "On-Line collision avoidance for redundant manipulators," *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, Vol. 1, pp. 36-43, Atlanta, GA, 1993.

[15] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press, Cambridge, Massachusetts, 1981.

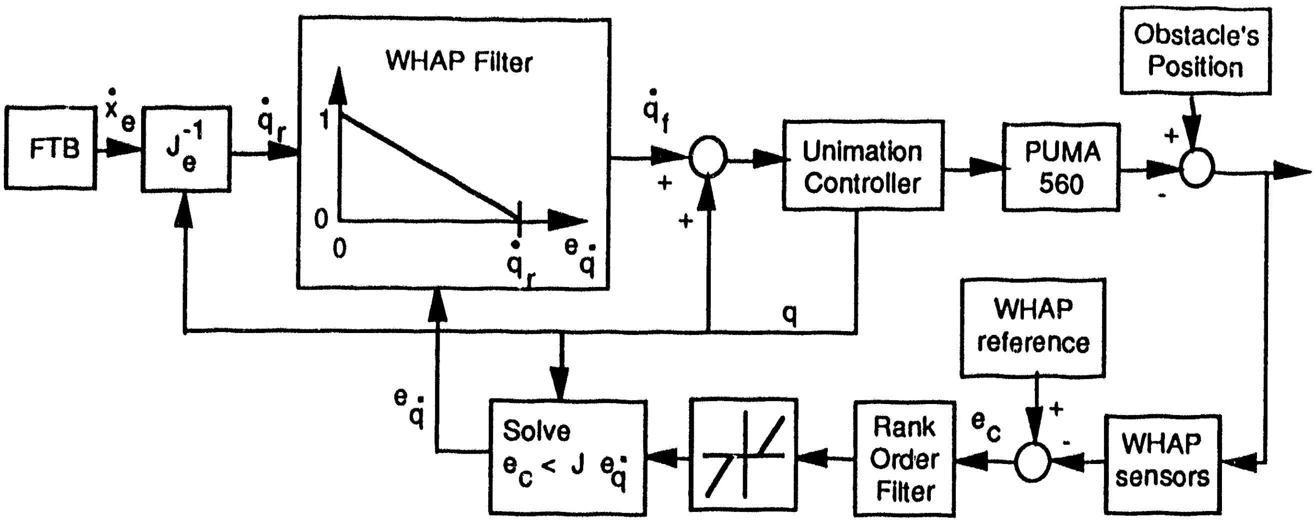

Figure 5. Control Structure for Obstacle Avoidance System.

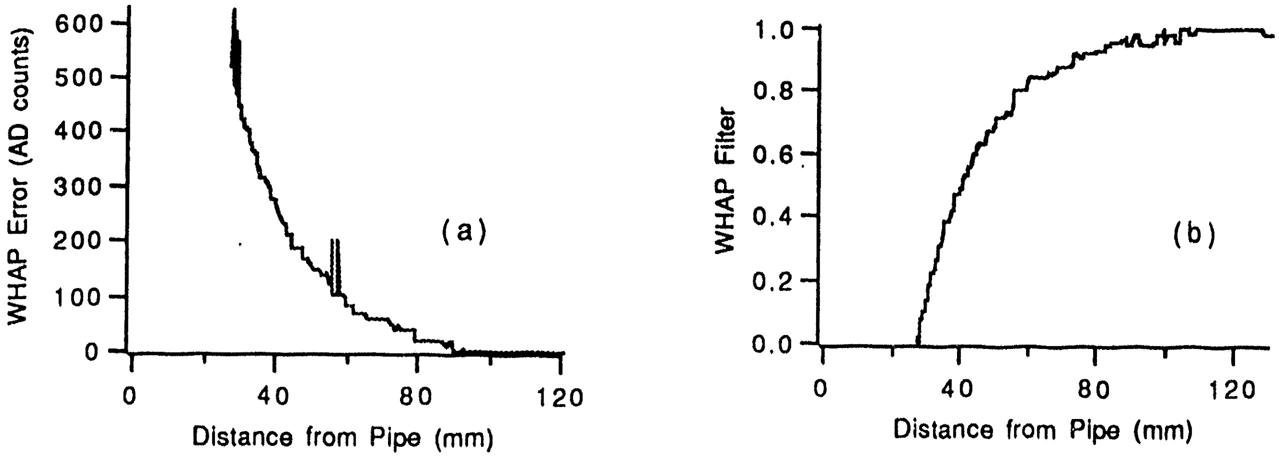

Figure 6. Experimental Results. (a) Raw WHAP sensor readings. (b) WHAP filter.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, 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 product, 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 any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

END

DATE FILMED

4/8/94