Proximity Sensing in Robot Manipulator Motion Planning : System and Implementation Issues*

Edward Cheung and Vladimir Lumelsky
Yale University, Department of Electrical Engineering
New Haven, Connecticut 06520

Abstract

In this paper, results of one implementation effort on a motion planning system for a robot arm operating in an uncertain environment are discussed. It has been known that path planning algorithms with proven convergence can be designed for some planar and three-dimensional robot arm manipulators operating among unknown obstacles of arbitrary shapes. The attractiveness of such systems lies in their ability to operate in a complex, perhaps even time-varying, unstructured environment. Implementation of these algorithms, however, presents a variety of hardware and algorithmic problems related to, first, covering the arm with arrays of sensors to form a "sensitive skin"; second, processing real-time sensor data; third, designing complementary algorithms for step-by-step motion planning based on limited local information; and finally, integrating these components, together with global planning algorithms, in a single system. We discuss various trade offs and solutions implemented in the sensor-based planning system for a planar arm manipulator developed in our laboratory, and present a summary of our experiments with it.

1. Introduction

Ongoing research in robotic motion planning encompasses two major trends. In one approach, often called the Piano Movers Problem, complete information about the robot and its environment is assumed. A priori knowledge about the obstacles is represented by an algebraic description, such as a polygonal representation; the environment is assumed to be static [1]. An overview of research in this area can be found in [2].

In the second approach -- to which this paper also belongs -- information about the environment is assumed to be incomplete. Such situations take place when the robot has no á priori knowledge about the environment, but is equipped with sensors that notify it of impending collisions or proximity to obstacles. Depending on a specific sensor, the area that is being sensed can be large, as when ultrasonic range sensors [3] or stereo vision [4] is used, or it can be a small local area, such as when proximity sensors are used. Within the latter type, optical sensors are especially popular. Applications include obstacle sensing

* Supported in part by the National Science Foundation Grants DMC-8519542 and DMC-8712357, and a grant from North American Philips Corp.

sensing [5] (see also this paper), object grasping [6,7] where optical sensors are mounted in the robot gripper, and tactile sensing [8].

One area of sensor-based motion planning that eluded automation so far involves robot arm manipulators that operate in situations where many or all parts of the robot body can be subject to unexpected collisions. The attractiveness of such systems lies in their ability to operate in a highly unstructured and even time-varying environment. Consider, for example, complex logistics and safety issues at a space station where an arm manipulator is expected to operate automatically amidst such moving obstacles as astronauts or arriving cargo. Only a sensor-based system capable of generating continuous purposeful paths among unknown obstacles while guaranteeing collision free motion for every point of the robot body would be acceptable in this application. A first attempt to build such a system is described in this paper.

Work on algorithms for such systems has been an active area of research in recent years [9,10,11,12]. It has been shown, specifically, that algorithms with proven convergence can be designed for motion planning of simple planar [9] and three-dimensional robot arm manipulators operating among unknown obstacles of arbitrary shape [13,14]. In these algorithms, the problem of motion planning for an arm is reduced

to that of a point automaton operating on the surface of a two-dimensional manifold. Resulting path planning procedures are quite elegant and computationaly inexpensive, and require the arm to "slide" along or follow the contours of obstacles encountered along its way. Although at present convergence is guaranteed only for an environment with still obstacles, the fact that no preliminary information about the obstacles is needed opens a possibility to apply this approach to time-varying environments.

While extending the theory of motion planning with uncertainty to more complex multi-link three-dimensional arm manipulators is an important area of current research, it is equally important to address the implementation issues. It is clear that, besides global planning algorithms, the following three basic components will have to be developed before such systems become feasible: (1) The robot must have a physical ability to sense the presence of an obstacle with every point of its body and to identify the points of the body that sense the obstacle. (2) Adequate signal-processing capabilities are vital for processing large amounts of sensor data in real time. (3) Complementary algorithms for local planning are needed, to guide the arm maneuvering around obstacles based on the local sensory data and in accordance with the global planning strategy.

Finally, these components have to be integrated, together with the subsystem for global planning, in a common structure. The integration process is vital: for example, depending on the strategy for local planning, more strict or less strict requirements may be imposed on the selection and calibration of sensors.

Realization of such a system encompasses a variety of hardware and data processing issues that, to our knowledge, have not been addressed before. This paper describes the first attempt to build such a complex real-time motion planning system. The current system presents a planar version implemented on the two main links and corresponding revolute joints of a PUMA 562 robot arm manipulator. The arm's sensory capability is provided by a "sensitive skin" formed by dozens of active infrared proximity sensors that cover the whole body of the robot arm. The resulting effect is a kind of aura similar to that produced by hair on the legs of some insects.

Section 2 below presents a brief overview of the whole system. Then, the next three sections discuss the system's components in more detail. In Section 3, some options considered in the course of developing the sensor hardware are briefly discussed and the implemented solutions are outlined. The computer and control architecture are reviewed in Section 4. Algorithmic issues of local step planning are addressed in Section 5. Section 6 summarizes the results of our experimental work. Finally, the Conclusion is presented in Section 7.

2. Overall System : Summary

The general flowchart of the implemented path planning system, depicted in Figure 1, shows the interaction and hierarchy of the various components in the system; in the diagram, the software blocks are shown by straight rectangles, and the hardware items by curved rectangles.

The distribution of infrared sensors on the sensitive skin is such that no approaching obstacle can penetrate the area around the arm within the distance of about 10-15 cm without being detected by at least one sensor. Given the implemented algorithms for local planning (see below), and given the range of velocities, update rates, and accuracies of typical industrial manipulators, this distance is sufficient to guarantee that no collision takes place during the motion. Below, the term "contact" refers to proximity sensing of obstacles. The resulting system is quite robust in that it is insensitive to the variability in characteristics of individual sensors and requires no elaborate sensor selection and calibration procedures.

The system continuously interrogates the skin sensors. The sample rates and the

corresponding hardware are designed so as to fit comfortably within the update rate of the arm, which in most industrial arm manipulators, as well as in our arm, is about 50 position points per second. The sensor information is then passed to the component of the system responsible for local planning (Figure 1).

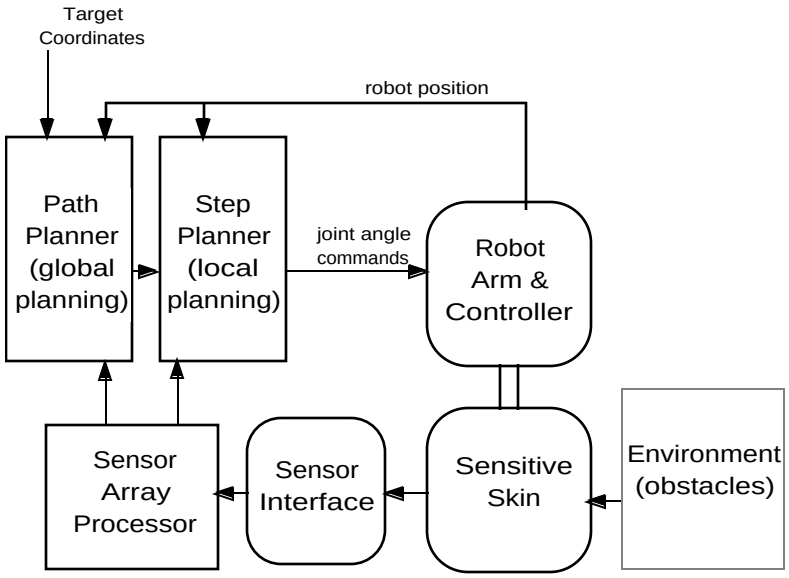

Figure 1. Information flow diagram of the path planning system

The basic cycle of local planning occurs at the same rate as the update rate of the sensors on the arm and includes on-line generation of the next arm positions. If the arm moves in free space so that no skin sensor detects an obstacle, the consecutive arm positions are generated so as to move the arm directly to the predetermined target position. What is meant by "directly" depends on the desired trajectory - the one that the arm would go through if it encountered no obstacles on its way. This can be, for example, a straight-line motion of the arm end effector in the work space, or a straight line in the space of arm joints, or any other predetermined curve.

Once one or more skin sensors sense an obstacle, the subsystem for local planning starts generating arm steps such as to result in the arm maneuvering around the obstacle. One obvious consideration in generating the size and direction of each step is to provide collision-free motion. What is meant by "collision-free" can be defined either probabilistically or deterministically. In the former case, the safety of a contemplated position is inferred indirectly, by extrapolating from available information. This is done sometimes in wall following by autonomous vehicles, where the next step is done parallel to what is expected to be the wall, based on the previous steps. In the application at hand, distances between the arm and obstacles are short, speeds are relatively high, and "wall" shapes are unpredictable. This deems the probabilistic techniques undesirable.

In deterministic techniques, the safety of the next position is assured by direct sensing. This can be done in one of two ways. One strategy is similar to that of a blind person with a cane, who first tests all candidate positions, and only then makes a step. This technique is appropriate for computer simulations [15] but is not very practical for our application. An alternative technique, also accepted in this work, capitalizes on the information provided by proximity sensing - that an envelope of certain thickness (in our case, about 10cm) around the arm is free of obstacles. Based on this information, the next step is calculated so that after accomplishing it all points of the arm body are still inside the same safe envelope. Then, a new safety envelope appears, and the process repeats.

Another consideration in step planning comes from algorithms for global planning

[9] which require that the arm maintain a contact with the obstacle through the whole operation of maneuvering around the obstacle. This corresponds to the arm motion along the local tangent to the obstacle at the point of contact. To generate the local tangent, the local planning subsystem takes into account the location of the point of contact on the robot body and the approximate distance to the corresponding obstacle surface. Hereafter, unless otherwise noted, "local tangent" refers to the obstacle boundary in configuration space [1].

If more than one point of the arm body are simultaneously in contact with (possibly more than one) obstacles, all such contacts are evaluated and one local tangent is chosen such as to guarantee safe motion. The general direction for maneuvering and the conditions for leaving the obstacle in order to proceed toward the target position come from the subsystem that is responsible for global planning, Figure 1. The latter is based on the algorithms for path planning with uncertainty described in [9]. These algorithms (see also Section 5.1) are based on topology of the configuration space, which is a function of the arm kinematics rather than of obstacle geometry. Consequently, the geometry or the number of obstacles that the arm is interacting with at any given moment are not essential for the system operation.

3. Sensitive Skin

3.1 General

The function of the arm sensors is to provide the motion planning system with two kinds of information - the fact of approaching obstacles and locations on the arm body where these obstructions take place. There are many considerations that can guide selection of an adequate sensing medium. For the purpose of this work, these were divided into two categories -- those that were deemed essential for demonstrating the feasibility of our general approach to motion planning, and those that could be sacrificed at the current stage. For example, a capability to handle various shapes and distributions of obstacles in the workspace was considered to be essential, whereas a capability to handle obstacle materials with a wide range of surface reflectivity - not essential. Below, criteria that guided our design of the sensor system are reviewed in more detail.

The system must guarantee that no obstacle approaching the arm remains undetected. Clearly, if any point of the arm body is subject to collision, the only way to guarantee obstacle detection is to supply every point of the body with the sensing capability. This suggests a sort of a "sensitive skin" an array of sensors covering the whole arm body. The density of sensors on the skin should be such that no dead zones appear on the arm body; otherwise, if an obstacle approaches the arm in an area of a dead zone, a collision may take place.

What sensing range - specifically, the maximum sensing distance of the sensors - would be adequate? On the one hand, the sensing distance should be limited, to reduce the size and weight of the electronics. Larger range requires more transmitted power and/or more sensitive receivers. This leads to larger components in the transmitter, and more complicated and larger receivers. This in turn will limit the sensor density on the skin and may result in dead zones on the arm body.

On the other hand, the smallest acceptable maximum sensing distance depends on the sensor sampling rate and the maximum linear speed of the arm. The latter depends on the arm specifications and/or intended applications, and is more likely to occur near the arm end effector. Clearly, between two successive scans of all skin sensors no point on the arm body should move more than the maximum sensing distance. This means that the product of the maximum sensing distance and the scan rate must be no less than the maximum linear speed of the arm:

Maximum arm speed < Maximum sensing distance x Sensor scan rate

The factors to be considered are thus the arm speed range, sensor reaction time (which determines the sensor scan rate), and sensor density on the arm skin. The result of this analysis is that for typical arm manipulators the maximum sensing distance should be about 10-15 cm.

The next choice to be made is between passive sensors, such as tactile or vision, and active sensors that operate by emitting a form of energy and sensing the reflected signal. For the considered application, passive sensors do not seem to present a viable alternative. For example, tactile sensing of obstacles in practice would amount to numerous collisions, whereas covering the arm body with sensors, each of which would provide a vision capability, is not practical [1]. Thus, a viable choice is likely to be among active sensors.

Two major types of active sensing make use of optical or acoustical radiation. In addition, inductive and capacitive methods can be used to accomplish sensing; these methods are generally limited to detection distances of less than 2 cm, and are heavily dependent on the material of the obstacle. Commercial inductive and capacitive sensors are mainly used in industrial environments where the applications require durability and involve objects of known composition that have to be sensed at very close range. With acoustic sensing (another popular term - sonar), a burst of ultrasonic energy is transmitted, reflected from an obstacle and then received, making use of the time of flight for distance measurement [16]. A drawback to this type of sensor is that, because the wavelength of sound is relatively long, some obstacles can exhibit specular (mirror like) reflection. Consequently, large flat surfaces may be undetected by the sensor because of the lack of reflected signal. This effect is especially pronounced at shallow angles between the energy beam and the obstacle - a case very common in the considered application.

---

[1]. Note that if one contemplated a vision system attached to the walls of the workcell, one would need a very large number - in principle, an infinite number - of "eyes" in order to guarantee that neither the arm itself nor the obstacles would ever occlude any possible contact between the arm links or the arm and an obstacle.

In addition, commonly available acoustical sensors, such as the popular Polaroid sensor, operate poorly when obstacles are placed closer than 0.9 feet from the sensor [17,18]. This effect can be only partially compensated by the active damping of the transducer [18]. This dead zone necessitates the path planner to restrict the distances of obstacle detection to larger than 0.9 feet from the sensor, which is not realistic in the case of motion planning for industrial arm manipulators. For example, such a system may render the target position unreachable if it can be reached only by passing between two obstacles located at about two feet from each other.

To overcome some of these shortcomings, sources of radiation of shorter wavelength can be used. Among those, sensors based on electromagnetic radiation were considered less desirable because of their bulkiness: for example, focusing electromagnetic radiation requires the use of focused antennae, such as parabolic dishes. Visible or infrared light seem to present a better choice: the wavelength of visible light is in the range of hundreds of nanometers, tending most surfaces to scatter light equally in all directions and thus appear matte [19].

If a sufficiently matte surface is illuminated by a narrow beam of light, the amount of reflected radiation will depend mainly on the distance between the surface and the

receiver. Although this does not produce a fully universal device, the insensitivity to obstacle orientation can be successfully exploited in detecting the presence of objects in a wide range of materials, shapes, and orientations. One drawback to using light for sensing is that objects of very dark color may not be sensed. Also, non matte surfaces and optical mirrors may exhibit specular reflection, possibly resulting in a lack of returned radiation.

Based on these considerations, a sensor system using light as the sensing medium was chosen. Given this, the next issue is extracting from the reflected light distance information needed for the operation of the planning system. Of the several techniques that can be used for this purpose, four that are relevant for the task at hand are reviewed here.

An approach often used in auto focus cameras is to emit a beam of light and use triangulation to determine distance. The advantage of the triangulation method is that the object colour has a reduced effect on the measured distance; on the other hand, optical mirrors continue to be difficult to detect. The disadvantage of using an array of optical triangulation sensors is the need to use optical components such as small lenses and lateral effect photo detectors (detectors that sense the position of a received spot of light). Due to the small size of these diodes (several square millimeters in area), assembly of the sensor array elements requires the placement of many lenses to an accuracy of a millimeter. The sensor system may therefore be not robust enough and be susceptible to vibration of the robot arm.

The concept of time-of-flight can also be used with an optical sensor to determine obstacle distance [20]. Similar to acoustic sonar, some of the light emitted from the transmitting element is then received back after having been reflected off an obstacle. The time that it takes for the burst of light to return is proportional to the obstacle distance. Light travels approximately one foot per nanosecond, therefore very high speed electronic circuitry is needed to detect obstacles in the range of interest (from zero up to about 15 cm). The complexity of building an array of such sensors is the reason that this approach was rejected.

A method presented in [7] utilizes extended Kalman filtering to allow a distance measurement. The problem with using this method for our use is the need for a learning and calibration stage for each sensor, requiring known motion of the sensor in the sensing direction. This may not be easy for sensors covering a robot arm with irregularly shaped links, or when the sensing direction is not perpendicular to the arm link. Furthermore, objects of varying size cannot be handled since the method calls for objects of standard shape, and size. A large increase in system complexity to incorporate this method is therefore still insufficient to insure immunity to variations in the obstacles encountered.

A more simple way of using the reflected light, which was used in our system, is to use the amplitude of the reflected light for proximity measurement. This form of sensor is readily available commercially, although its output is typically digital, going 'high' if the reflected light exceeds a certain threshold that can be set by the user. Analog output sensors are also available, although they are in general too bulky to be incorporated into a sensor array that is to cover the surface of an arm. The sensor that was eventually built in our laboratory uses the reflected light for proximity measurement, and is specifically designed to fit on the surface of a robot arm; its more detailed description is given in the next section.

3.2 Sensor Hardware

The transmitted infrared (IR) light used in the sensor system has a wavelength of approximately 875 nm. This light is modulated at a frequency of 10 kHz, and is then coherently or synchronously detected after reception to improve immunity to other light

sources, such as ambient light. The reflected light is demodulated by a multiplier, amplified, and then passed to a digitizer.

In the current design, 16 *sensor pairs*, each consisting of a photo transistor and an infrared LED (IRLED), are time multiplexed together, forming one *sensor module.* The entire *sensor system* is comprised of a number of these sensor modules mounted on the arm. A sketch of the sensor module is given in Figure 2, and its functional block diagram - in Figure 3, and a photograph of a part of the actual module is shown in Figure 4; its more complete schematics is described in the Appendix. Note two sensor pairs in Figure 4 highlighted by the reflection in the middle of the black acrylic.

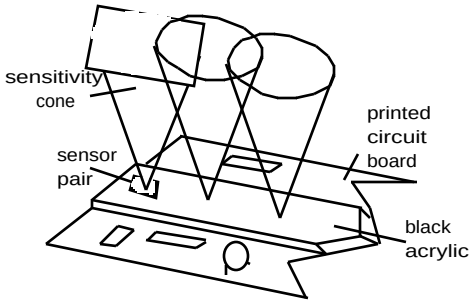

Figure 2. A sketch of the sensor module.

The sensor module is designed so that all optical components and instrumentation are on one printed circuit board. Each sensor module includes a transistor block and a receiver block. The transmitter block amplifies the output of the voltage controlled oscillator (VCO) (marked B in Figure 3). The block consists of an analog multiplexer that selects one of the sixteen IRLED that is to be flashing, and a transconductance amplifier. The diode current is at a duty cycle of 5%, with 'on' currents of 1 Ampere. The receiver block consists of a transresistance amplifier, a filter, and a multiplexer that selects one of the sixteen photo transistors. The current generated in a photo transistor is produced by the incident light consisting of the reflected infrared light from the transmitter, as well as unwanted room lighting. A large portion of the room lighting's presence is removed by high pass filtering of the photo transistor current in the receiver block. The output of this block is connected to the phase locked loop (PLL) input A.

As soon as enough light is reflected (this occurs when an obstacle is about 15 cm away), the PLL will lock into phase with the reflected signal, which has the same frequency as the VCO. The second multiplier, external to the PLL, with inputs C and D, multiplies the output of the receiver and VCO, producing a DC signal if there is a reflected signal that is in phase with the VCO signal. The PLL and the second multiplier are implemented in a commonly available tone decoder chip made by the National Semiconductor Corporation. A description of the operation of a typical phase locked loop can be found in [21]. Each sensor module contains one complete detector consisting of the PLL and second multiplier as shown in Figure 3. Five such detectors are used in the system, each with 16 sensor pairs. These five detectors are read by the sensor controlling computer every 1.25 mseconds, after which five new sensor pairs are addressed. It takes 20 msec to scan the entire arm since all detectors function independently.

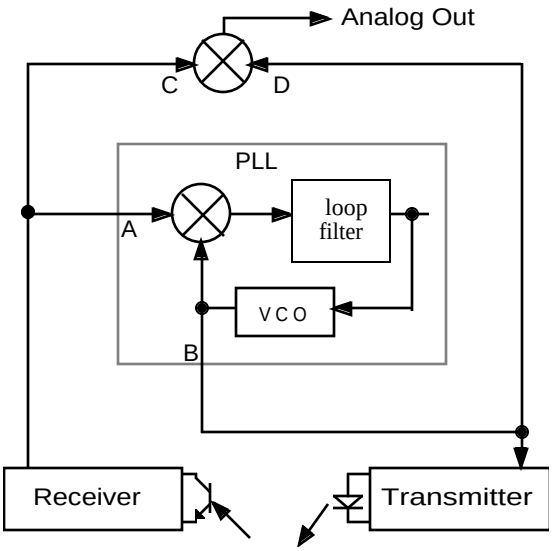

Figure 3. Sensor module, functional block diagram.

The Analog Out signals generated in sensor modules are then multiplexed in the sensor interface, Figures 1 and 5. This signal's voltage decreases with the increasing reflected light and is used for distance detection. Each sensor module is separately calibrated by placing a target consisting of a 12x12 cm piece of white paper at the desired maximum sensing distance. The Analog Out signal then adjusted for full scale output. Refer to the Appendix for a more detailed explanation and a circuit schematic.

photo here, 13 lines

Figure 4. Sensor module

The outputs of all sensor modules are polled by the microcomputer controlling the sensor system. For a system with five sensor modules which has been implemented in this project, digital control consists of seven bits of digital data. The three most significant bits are used to select a particular sensor module, and the 4 least significant bits are connected to all sensor modules and used to select a particular sensor pair. Also, one extra bit is used to trigger the analog to digital converter (ADC). The ADC is connected to the output of the one-of-five analog multiplexer that selects a particular module. Thus a byte (8 bits) of data is written to the sensor system, and after the ADC has had enough time to convert (about 10 µsec), a byte of range data is read from the ADC by the sensor controlling microcomputer, and translated into the proximity between the obstacle and the corresponding sensor pair.

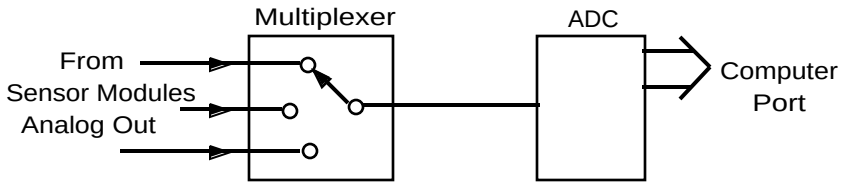

Figure 5. Detail of the Sensor Interface

A typical sensor response is shown in Figure 6: the test object consists of a 12x12 cm piece of white paper. Typically, the sensor sensitivity becomes insufficient beyond a distance of 15cm from the obstacle. Factors such as obstacle color, size, and surface texture affect the proximity reading. The darker the color, or the smaller the obstacle, the closer the obstacle will be before it is sensed. On the other hand, obstacles larger than 12x12 cm produce approximately the same response. This results from the fact that at distances at which the sensor operates (about 15 cm), the emitted light of the LED illuminates only a spot on the obstacle, causing the rest of the obstacle to be invisible to the sensor. Increasing the obstacle size in this invisible region does not affect sensor reading. Instead, the obstacle may then be sensed by another sensor pair (with a different invisible region). No attempt was made to reach uniform sensor reading from sensor to sensor, through a special selection of sensors or by calibrating the sensor pairs with respect to each other. Given the fact that the arm is only required to avoid collisions on its way around obstacles, rather than to maintain a constant distance from obstacles, the operating tolerances of off-the-shelf optical components were found close enough to produce a robust system.

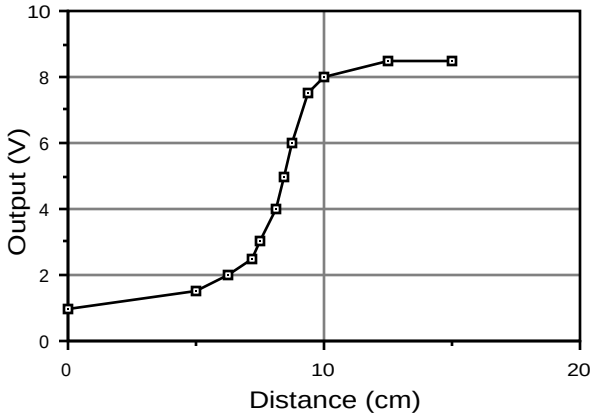

Figure 6. A typical response of the infrared sensor.

4. Computer Architecture

The two microcomputers used in the path planning system are manufactured by Pacific Microcomputers. These single board micros each contain a MC68020 CPU operating at 16Mhz, an MC68881 math coprocessor, 1Mbyte of on board DRAM, and a parallel input/output (I/O) board that 'piggybacks' onto the main board. The boards are plugged into a VME back plane, connected to a Sun Microsystems workstation, which acts as the development system. Programs are written in the C programming language and compiled by the Sun's compiler. Thereafter, programs can be downloaded onto the micro's boards via the VME bus, and then executed. The three computers, one Sun and two micros, can run programs that pass data among themselves, enabling the micros to handle the work involved, and permitting the Sun to log data or act as a watchdog to alert the user in case of a failure. A more detailed description of the use of each microcomputer follows.

One microcomputer, called 'Senscon', is used to interface to the sensors. This computer reads the 80 sensor pairs available in the current system, and puts the analog range data into memory, making the data easily available to 'Planner', the micro that handles path and step planning. It takes Senscon about 20 msec to poll the entire array of arm's sensors, which includes the time needed to calculate all the local tangents associated with each sensor on the arm. The use of these local tangents, as well as the path and step planning, is described in Section 5.1.

The 80 sensor pairs are arranged in 5 groups of 16 pairs each. Each group of 16 has its own detector (Figure 3) that operates independently of and in parallel with the other detectors, and together comprise of a sensor module. At any one time, the five sensor pairs are being polled sequentially, after 1.25 msec of settling time. It therefore takes 20 msec (16x1.25) to poll the entire sensor skin.

As soon as a sensor is polled, its information becomes immediately available to Planner. Besides handling the step and path planning, Planner serves as an interface to the robot arm. For the two degrees of freedom to be controlled, Planner writes two bytes of data to the interface of the robot, each byte representing the commanded velocity of the corresponding robot joint. The commanded velocities are interpreted by the robot's operating system, which then drives the arm motors to the desired velocity. As feedback, the robot arm returns to Planner its current position. Planner then uses the sensor data from Senscon, the desired Target data from the Sun computer, and the current robot position to plan and command the next step for the robot arm. The calculation of each step occurs at intervals of 1.6 msec, and is done asynchronously with the operation of Senscon, so that each step is planned based on the most recent information from all the sensors.

The overall processing speed achieved by the planning and control system are quite satisfactory. Since the sample rate of the PUMA robot is 36 msec (which is quite close to that of the majority of existing industrial robot arm manipulators), and the longest operation in our system takes 20 msec, the required data processing fit rather easily into the real-time operation. It seems that faster rates can be realized if needed, within the framework of the same system.

Overseeing the operation of the two micros is the Sun development station. Using the Sun, the user can enter the desired Target position of the robot, which is then passed to Planner. The Sun can also abort the operation if need be. A small graphics package has also been developed to graphically document the movement of the robot as it maneuvers around obstacles.

5. Step Planning and Contour Following

5.1 General

In addition to the information about a contact with an obstacle provided by the proximity sensor, information on the location of the obstacle relative to the arm is also available. This is because the computer system that addresses each individual sensor knows the placement of every sensor on the arm. If a sensor senses an obstacle, the obstacle's position with respect to the arm can be derived from the placement of the sensor on the arm. It will be shown now how the local planning subsystem uses these data for contour following. Since contour following is done in conjunction with the global path planning algorithm [9], the latter is first briefly discussed below.

Consider a simple two-link planar robot arm with two revolute joints, Θ_1 and Θ_2, Figure 7. Link 1 and link 2 of the arm are represented by the line segments O-B and B-P, of lengths l_1 and l_2, respectively; J_1 and J_2 are the arm joints. Point P represents the wrist; point B, which coincides with joint J_2, represents the arm elbow; point O is fixed and represents the origin of the reference system. If, on its way to the target position, one

or both arm links come into contact with an obstacle, the arm maneuvers around the obstacle according to the path planning algorithm.

The robot arm of Figure 7 can be represented by a point in the configuration space (Θ_1, Θ_2). Because the values ($\Theta_i + 2\pi$), i=1,2, correspond to the same position of each link, the configuration space represents a common torus. Or, it may represent a subset of the torus, if the range of a joint is limited. Correspondingly, any obstacle in the work space has its image in the configuration space. Then, the path planning problem becomes that of moving a point automaton on the surface of a common torus. Although the actual algorithm takes into account the topology of the torus (for more detail, see [9]), its main idea can be presented in terms of the planar configuration space, Figure 8. Assume for now that the whole robot body is covered with tactile sensors so that any point of the body can detect a contact with an obstacle, and the location of that point is known. (Actually, it is of no importance to the algorithm whether the sensing is tactile or proximal).

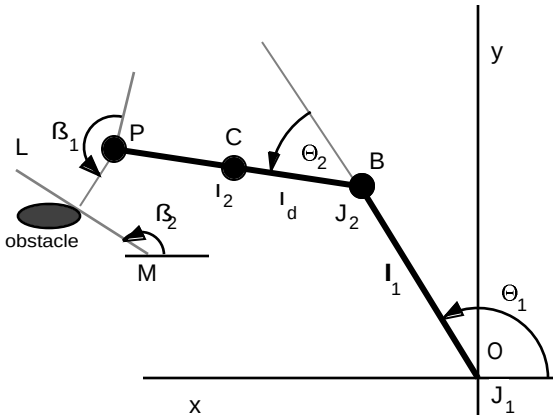

Figure 7. A sketch of the 2-link arm with revolute joints.

In the path planning algorithm, the automaton moves from the start position S to the target position T, along the line segment S - T. If an obstacle is encountered, the point of contact is designated as a hit point, H. The automaton then turns in a prespecified local direction (e.g. left, as in Figure 8) and follows the contour of the obstacle until the line segment S - T is again met at a distance from T shorter than the distance between the lastly defined hit point and T. At this point, called a leave point, L, the automaton follows the line S - T towards T, unless another obstacle is met causing the process to repeat. The algorithm has been shown to converge to the target if it is reachable, or to conclude in finite time that the target is not reachable if such is the case [9].

To realize contour following required by the path planning algorithm, a local procedure is needed to plan the next little step along the obstacle boundary at a given location of the robot arm. The input information for the procedure is the current location of the arm and the local tangent to the obstacle boundary in configuration space, Figure 9. The calculation of the local tangent at the contact point to the obstacle in configuration space requires knowing what point(s) of the robot body are in contact with the obstacle in the work space. Note that contour (obstacle) following requires no information about the shape or position of the obstacle in work space or in configuration space.

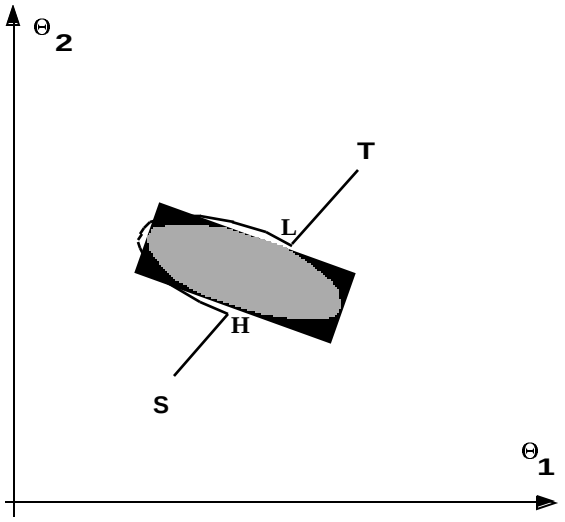

Figure 8. Configuration space of robot arm of Figure 7.

The motion of the point automaton along a local tangent in configuration space corresponds to the arm "sliding" along the obstacle at the point of contact in the work space. This process is done repeatedly during contour following. Note that one or more obstacles can be sensed simultaneously by more than one sensor pair. Below, the procedure for calculating the local tangent is described, followed by the step planning algorithm that uses the local tangent for calculating the next step along the obstacle boundary. Unless otherwise noted, "local tangent" refers to the obstacle boundary in configuration space, as indicated in Figure 9.

5.2 Calculation of Local Tangent

The following derivations are valid for a 2 degrees of freedom revolute - revolute arm, Figure 7, but can be similarly derived for other kinematic configurations.

Whenever an obstacle is encountered, the arm attempts to slide along its surface. This sliding is accomplished by a coordinated move between joints J_1 and J_2, based on the value of the local tangent (Figure 9). To find the latter, estimates of the derivatives $d\Theta_2$ and $d\Theta_1$ are computed at every point along the contour, using the procedure described below.

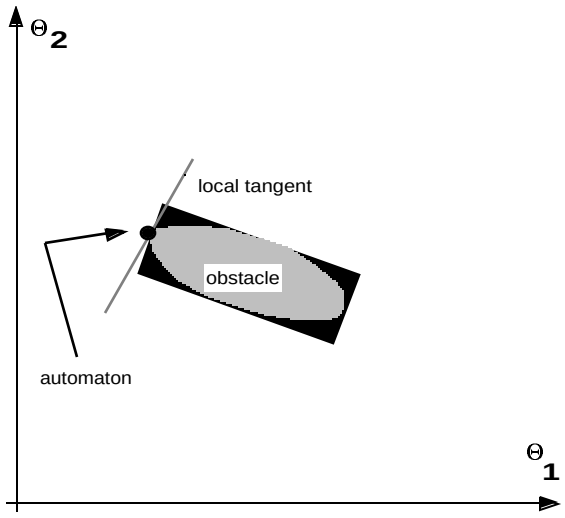

Figure 9. Using the local tangent for contour following.

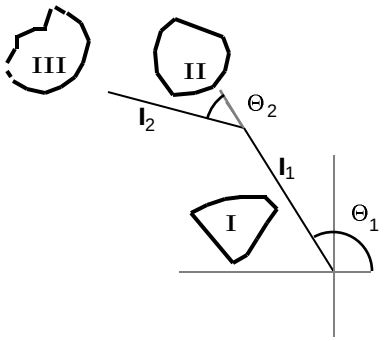

Figure 10. Obstacle types.

Depending on their location relative to the arm, obstacles that may occur in the work space are divided into three categories, easily recognizable by the sensor system: Type I, Type II, and Type III, Figure 10. Note that the notion of the obstacle type is relative: the same physical obstacle can correspond to different types at different moments, depending on what part of the arm interacts with it. Now we consider each of the types.

Type I are those obstacles that obstruct link 1. Since link 2 is irrelevant in such cases, then $d\Theta_1 = 0$ and $d\Theta_2 \neq 0$. Therefore, the local tangent to the obstacle boundary in this case is vertical.

Type II are those obstacles that obstruct link 2. Assume that link 2 of the arm is obstructed at point C by a Type II obstacle, at the distance l_d from the joint J_2, Figure 7. Then, the estimates of $d\Theta_1$ and $d\Theta_2$ at C can be found as follows. Write the expressions for the x and y coordinates of C, c_x and c_y, respectively:

$$c_x = l_1 \cos(\Theta_1) + l_d \cos(\Theta_1+\Theta_2) \qquad (1)$$
$$c_y = l_1 \sin(\Theta_1) + l_d \sin(\Theta_1+\Theta_2)$$

Take total derivatives:

$$dc_x = -l_1\sin(\Theta_1)d\Theta_1 + dl_d\cos(\Theta_1+\Theta_2) - l_d\sin(\Theta_1+\Theta_2)d\Theta_1 - l_d\sin(\Theta_1+\Theta_2)d\Theta_2 \qquad (2)$$
$$dc_y = l_1\cos(\Theta_1)d\Theta_1 + dl_d\sin(\Theta_1+\Theta_2) + l_d\cos(\Theta_1+\Theta_2)d\Theta_1 + l_d\cos(\Theta_1+\Theta_2)d\Theta_2$$

Since C is a stationary point, $dc_x = dc_y = 0$. Find dl_d from both equations of (2):

$$dl_d = \frac{[-l_1\cos\Theta_1 - l_d\cos(\Theta_1+\Theta_2)]d\Theta_1 - l_d\cos(\Theta_1+\Theta_2)d\Theta_2}{\sin(\Theta_1+\Theta_2)} \qquad (3)$$

and

$$dl_d = \frac{[l_1\sin\Theta_1 + l_d\sin(\Theta_1+\Theta_2)]d\Theta_1 + l_d\sin(\Theta_1+\Theta_2)d\Theta_2}{\cos(\Theta_1+\Theta_2)} \qquad (4)$$

Equating the right hand sides and eliminating the denominators in (3) and (4), obtain:

$$\left[\cos(\Theta_1+\Theta_2)\left(-l_1\cos\Theta_1 - l_d\cos(\Theta_1+\Theta_2)\right) - \sin(\Theta_1+\Theta_2)\left(l_1\sin\Theta_1 + l_d\sin(\Theta_1+\Theta_2)\right)\right] d\Theta_1 = l_d\left[\cos^2(\Theta_1+\Theta_2) + \sin^2(\Theta_1+\Theta_2)\right] d\Theta_2 \qquad (5)$$

Now, the ratio $d\Theta_2 / d\Theta_1$ is found as

$$\frac{d\Theta_2}{d\Theta_1} = \frac{1}{l_d}\left(-l_1\cos\Theta_1\left(\cos(\Theta_1+\Theta_2)\right) - l_1\sin\Theta_1\left(\sin(\Theta_1+\Theta_2)\right) - l_d\right)$$

$$= -\left(\frac{l_1}{l_d}\left\{\cos\Theta_1\left(\cos(\Theta_1+\Theta_2)\right) + \sin\Theta_1\left(\sin(\Theta_1+\Theta_2)\right)\right\} + 1\right) \qquad (6)$$

After simplification, the local tangent to a Type II obstacle at point C is given by:

$$\frac{d\Theta_2}{d\Theta_1} = -\left(\frac{l_1}{l_d}\cos\Theta_2 + 1\right) \qquad (7)$$

Type III are those obstacles that obstruct the arm wrist, point P, Figure 7. In addition, for certain arms such as the PUMA 562 that has an elbow that does not correspond to the arm joint J_2, obstacles obstructing the elbow can also be considered of Type III. This is explained in more detail at the end of Section 5.3 and Figure 14. To find the local tangent in this case, the inverse Jacobian of the arm is used [22]. Denote dx = dP_x and dy = dP_y in case the wrist is obstructed, and dx = dB_x and dy = dB_y in case the elbow is obstructed. Then, the following relationship holds:

$$d\Theta_1 = \frac{l_2\cos(\Theta_1+\Theta_2)\,dx + l_2\sin(\Theta_1+\Theta_2)\,dy}{l_1 l_2\cos\Theta_2}$$

$$d\Theta_2 = \frac{\left[-l_1\cos\Theta_1 - l_2\cos(\Theta_1+\Theta_2)\right]dx - \left[l_1\cos\Theta_1 - l_2\cos(\Theta_1+\Theta_2)\right]dy}{l_1 l_2\cos\Theta_2} \qquad (8)$$

Sliding of the wrist P along the obstacle corresponds to its moving along the line segment LM, Figure 7. Define $ß_1$ as the angle between the line perpendicular to link 2 and the line from P to the obstacle, and $ß_2$ as the angle between the line LM and the positive x-axis. Then,

$$\frac{dy}{dx} = \tan ß_2$$

or (9)

$$dy = dx\,\tan ß_2$$

where $ß_2 = \Theta_1 + \Theta_2 + ß_1 - \pi$. Substituting the expression for dy from (9) into (8), obtain the ratio $d\Theta_2 / d\Theta_1$:

$$\frac{d\Theta_2}{d\Theta_1} = -\left(\frac{l_1\cos\Theta_1 + l_2\left(\cos\Theta_1\cos\Theta_2 - \sin\Theta_1\sin\Theta_2\right) + \left[l_1\sin\Theta_1 + l_2\left(\sin\Theta_1\cos\Theta_2 + \cos\Theta_1\sin\Theta_2\right)\right]\tan ß_2}{l_2\left[\cos\Theta_1\cos\Theta_2 - \sin\Theta_1\sin\Theta_2 + \left(\sin\Theta_1\cos\Theta_2 + \cos\Theta_1\sin\Theta_2\right)\tan ß_2\right]}\right)$$

(10)

After simplification, the expression for the local tangent in the case of a Type III obstacle appears as:

$$\frac{d\Theta_2}{d\Theta_1} = -\left(\frac{\frac{l_1}{l_2} + \cos\Theta_2 + \sin\Theta_2\tan(\Theta_2 + ß_1)}{\cos\Theta_2 + \sin\Theta_2\tan(\Theta_2 + ß_1)}\right) \qquad (11)$$

Summarizing, for a Type I obstacle, the local tangent is vertical, and for Type II and Type III obstacles it is given by the expressions (7) and (11), respectively.

5.3 Step Planning Algorithm

Every point of contact between the arm body and an obstacle has an associated sensor pair that does the corresponding obstacle detection. For every sensor pair that detects an obstacle, a local tangent is calculated based on the method described above. The motion of the point automaton along a local tangent in configuration space corresponds to the arm "sliding" along the obstacle at the point of contact in the work space. This process is done repeatedly during contour following.

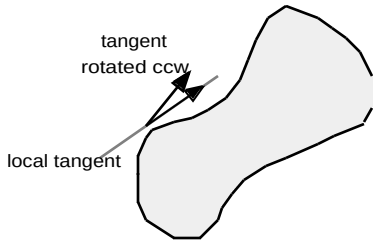

Figure 11. Configuration Space: ccw rotation of local tangent causes the robot to move away from the obstacle.

If, for example, the chosen local direction is "left", as in Figure 8, meaning that the robot arm should maneuver around the obstacle in a clockwise (cw) fashion, a counterclockwise (ccw) rotation of the calculated local tangent will cause the robot arm to move away from the obstacle, Figure 11. The amount of guaranteed collision-free rotation within one step is determined by the distance at which the obstacle is detected and by the arm geometry. In order for this strategy to work, the distance cannot be zero, which dictates the use of proximity sensors. As for the actual distance between the arm and an obstacle during the contour following, it is unimportant as long as, on the one hand, the arm does not touch the obstacle and, on the other hand, it stays in contact with it. If the arm is too close to the obstacle, the local tangent can be rotated ccw to increase the distance to the obstacle. The reverse is done if the sensed obstacle is still a long distance from the arm. No adjustment is made to the tangent to be followed if the output of the sensor is at some preset nominal value. This assures that the distance between the arm and the obstacle is within the sensing range.

On the lower level of control during contour following, an important parameter in the loop is the proportional feedback gain, K_p. This gain is expressed in units of degrees per error in volts; here, the error is defined as the difference between the desired reference level, set at 3 volts, and the sensor output voltage, V_s. Then, the rotation of the local tangent in degrees, Rot, is found as Rot = K_p (3 - V_s); the sign of Rot determines the direction of rotation. If, for example, K_p is set to 10 degrees/volt, and the sensor detecting the obstacle has an output voltage V_s= 2.5 volts, then the local tangent will be rotated ccw 5 degrees. The resulting value of the local tangent is equal to the sum of the value Rot and the angle between the original local tangent and the positive Θ_1 axis.

In our experiments, a wide range of values for K_p were found to result in an acceptable behavior. Outside of this range, very large values produce an oscillatory behavior during contour following, whereas very small values lead to sluggish following of abrupt contours. See an example in Section 6.

A more difficult situation in contour following develops when more than one sensor pair senses one or more obstacles simultaneously. Then, a number of local tangents are calculated, of which one is selected for executing the next step. The following analysis of two possible cases illustrates this point.

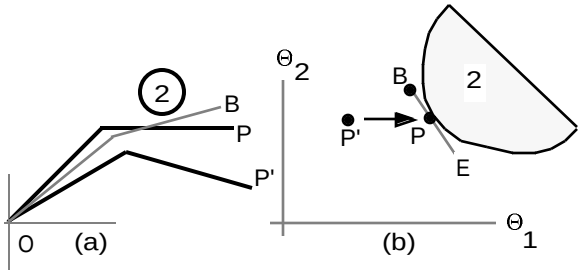

Figure 12. Arm interaction with a single obstacle:
a) work space, b) configuration space.

1. Interaction with a single obstacle.

Suppose the robot arm is moving from point P' towards point P, in the direction of increasing Θ_1. At this point the arm is obstructed by obstacle 2, which presents an obstacle of Type II, Figure 12a. The local tangent at P is EB, figure 12b. In the vicinity of point P, a further increase in Θ_1 results in crossing the local tangent, which is likely to lead to a collision with the obstacle, Figure 12a. If the local direction is "left", the next move along EB towards point B will cause the arm to slide along obstacle 2 to the position indicated by the dotted line in Figure 12a. Continual recalculation and motion along the resultant tangent constitutes the process of contour following.

2. Interaction with multiple obstacles.

For the case with multiple obstructions, suppose the robot arm is moving from point P' to P, in the direction of increasing Θ_1, Figure 13. At point P, the arm is now obstructed by three obstacles that are sensed by the sensor system, Figure 13a. A local tangent is then calculated for each obstacle. Obstacles 1, 2, and 3, of Type I, II, and III, produce tangents CD, BE, and AF, respectively. For the local direction "left", the robot can move towards one of the points A, B or C. Moving to point C necessitates crossing the lines BE and AF, with a large likelihood of penetrating the obstacles 2 and 3 associated with these two local tangents, an unacceptable situation. Similarly, moving towards point B could cause collision with obstacle 3, because the local tangent associated with it (AF) is crossed. Moving towards point A would not cause the crossing of any local tangents, which means that no collision will take place, and is therefore the chosen move. As a result of this move, the arm loses contact with obstacles 1 and 2, but remains in contact with obstacle 3. Analogously, a local direction "right" would necessitate a move towards point D for correct contour following.

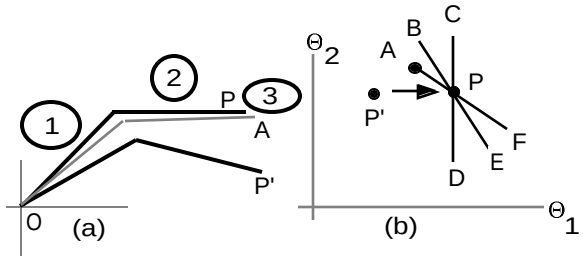

Figure 13. Arm interaction with multiple obstacles:
a) work space, b) configuration space.

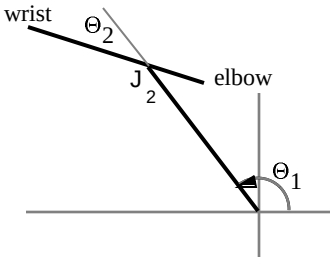

Figure 14. Example of an extended elbow on link 2.

For certain articulated arms, such as the PUMA 562, the second link extends out on either side of the joint J_2, causing the elbow to be susceptible to collision, Figure 14. To find the local tangent for obstacles obstructing the elbow, the same method described above is used, except Θ_2 is now replaced by $\Theta_2 + \pi$. Accordingly, Type II obstacles are those that obstruct the link from J_2 to the tip of the elbow, and Type III are the obstacles that obstruct the tip of the elbow. The quantity l_2 in expression (11) is now the distance from J_2 to the tip of the elbow.

6. Experimental Work

The system described above has been implemented on a PUMA 562 robot arm at Philips Laboratories at Briarcliff Manor, NY. To realize the planar system, only the (major) first two links of the arm were used (Figure 15). In the present configuration, the real time control system is based on the two distributed control boards from Pacific Microcomputers described in Section 4. The arm's skin includes 80 infrared sensors that allow the arm to detect the approaching obstacles from any direction in the plane. No attempt has been made to select sensors in any special way, or to position them on the skin so as to improve their mutual compatibility, or to calibrate the corresponding electronics so as to provide uniform sensitivity. The difference in the distance sensitivity of our sensors is, on the average, about 25%. In spite of such grossly non-flat distribution of sensitivity between individual sensors, the performance of the control/planning system is quite robust.

In a series of experiments that have been carried out, the system proved effective in accomplishing various path planning tasks. No collisions between the arm and the obstacles ever took place. In some experiments, the target position was not reachable because of interference with obstacles, and the system correctly concluded that this was indeed the case.

In each experiment, path planning was done based only on the on-line information from the sensor system -- no á priori information about the obstacles or about the shape of the arm links were given to the control/planning system. No learning of any kind or memorizing of the traversed path was included. Consequently, when the arm encountered the same obstacle for the second time, it treated it as an unknown obstacle. The sensor and planning subsystems were presented with tasks that ranged from sensing an obstacle by a single sensor on the arm's skin to the cases of multiple simultaneous contacts between the sensors on one or both arm links and one or more obstacles.

To test different cases, obstacles were chosen and positioned so as to present all possible obstacle types (Type I, Type II, or Type III, see Section 5.2) to the step planning procedure, depending on what parts of the arm body sensed the obstacle. Accordingly, the calculations of local tangents went through all the cases considered in Section 5.2, and the step planning procedure was forced to handle more difficult cases of multiple contacts, such as that in Figure 13.

Figures 15, 16, and 17 represent some of our experiments. In the experimental setup shown in Figure 15, the path of the arm to its target position is obstructed by a

rectangular obstacle. During the arm's maneuvering around the obstacle, the step planning procedure treats the obstacle as a Type II or Type III obstacle (see Section 5.2), depending on what parts of the arm body sense the obstacle. The screendumps of the configuration space corresponding to two experiments with this setup are shown in Figures 16a and 16b. The horizontal and vertical axes, labeled Theta 1 and Theta 2, correspond to the two joint variables, Θ_1 and Θ_2, respectively. The parts of the path where the robot arm moves unobstructed are shown by a thin line, and those where the robot arm is following the contour of an obstacle are shown by a thick line.

photo here, 13 spaces

Figure 15. Experimental setup with a rectangular obstacle.

The experimental results documented in Figure 16 present two separate tasks. In the first task, the robot has been commanded to move from the start at $\Theta_1 = 0^\circ$, $\Theta_2 = 0^\circ$ (intersection of the two axes) to the target point $\Theta_1 = 70^\circ$, $\Theta_2 = 70^\circ$. The second task is then to move back to the initial start position. Referring to Figure 15, in workspace, the start position of the robot arm corresponds to the arm extending straight up towards the ceiling, and the target position corresponds to the first link rotated counterclockwise by 70° and the second link rotated counterclockwise by 70° with respect to the first link. Thus, the path planning system needs to avoid the rectangular obstacle in order to swing the arm from the start to the target position.

cutout of configuration space, 17 spaces

Figure 16. Configuration space of two tasks executed on the experimental setup of Figure 15: a) K_p=10 degrees/volt; b) K_p=25 degrees/volt. (The double-line trajectory around the obstacle is an artifact produced by the graphics system; it is actually a single thick line).

In each task, the planning system made the arm follow a different section of the obstacle. Namely, the trajectory produced in the first task is represented by the upper curve AB in Figure 16a; the complement of this trajectory in Figures 16a and 16b corresponds to the second task. When combined, the two trajectories happen to produce in this experiment a complete image of the obstacle in the configuration space, Figure 16a. In both tasks, the arm successfully negotiated the unknown obstacle and reached the target positions. During the arm maneuvering around the obstacle, the distance between the arm and the obstacle varied from 2 to 12 cm, and was in the range of 5 to 10 cm for most of the time.

The same two tasks have been repeated in two experiments whose purpose was to study the sensitivity of contour following to the proportionality gain, K_p (see Section 5.3). In the first experiment, depicted in Figure 16a, the gain was set at a value K_p =10 degrees/volt chosen from a relatively wide range of values found to produce an acceptable behavior. Note that the motion of the arm along the obstacle is quite smooth. An example of a proportionality gain that resulted in a degraded performance is provided by the second experiment, shown in Figure 16b. Here, K_p =25 degrees/volt; only the trajectory produced in the second task is presented. Note that the arm motion along the obstacle becomes somewhat oscillatory, although it still follows the obstacle contour.

photo here, 15 spaces

Figure 17. Experimental setup with a C-shaped obstacle.

Another experimental setup, with a non-convex C-shaped obstacle, is shown in Figure 17. Again, at different moments the step planning procedure treated the obstacle as a Type II or Type III obstacle. While following the contour of the concave (hollow) section of the obstacle, the tip of the arm entered the hollow, but there was no contact with the obstacle; the arm successfully reached the target position. Here, K_p =10 degrees/volt, and the contours of the obstacle were followed with smooth motion.

7. Conclusion

Many creatures in nature, including humans, possess a capability to sense an environment with every point of their bodies. They also know how to use the sensor data

to generate purposeful motion. Furthermore, they do it in real time - even in tight situations involving simultaneous interaction with multiple obstacles, where the amount of required on-line processing seems overwhelming. This is a clear example of sensor-based motion planning with uncertainty - a problem that roboticists have been struggling with in recent years. Besides obvious academic interest, one can easily envision many robotics applications (e.g., in highly unstructured environments) where emulating such capabilities would be very attractive.

The described system is, to our knowledge, the very first technical system of this kind. No attempts have been made until now to implement a real-time sensor-based motion planning system for a robot arm manipulator operating among arbitrarily shaped obstacles. Even less is known about a more general case, considered here, when every point of a robot body is subject to potential collisions. In this paper, we, first, attempted to convince the reader that realization of such a system requires significantly more than just a good planning strategy. Second, we identified those functional components that are needed in order to realize sensor-based motion planning, and proceeded to develop those components.

The main of these components are (1) hardware that allows every point of the arm body to detect a potential collision, (2) signal-processing hardware/software for real-time processing of data from a large (about 100 sensors or more) sensor arrays, and (3) algorithms for local planning that control the arm maneuvering around obstacles in accordance with sensor data and the global planning strategy. The development, integration into a unified system, and validation of these components presents the main contribution of this paper.

Given the conceptual and implementation novelty of the system, we found its performance surprisingly good. It indeed performs in accordance with the underlying theory. The system is robust: no collision ever took place in our experiments; in spite of significant variation of responses of off-the-shelf sensors, no special selection/calibration procedures were needed. Finally, the system was found to be fast enough to handle real-life speeds of typical robot arms.

A big question, of course, is about a 3D version, which presents a natural continuation goal for this work. This question can be divided into two parts: one, on the availability of 3D global planning algorithms, and the other, on the 3D versions of the components mentioned above. The work on 3D global planning algorithms with guaranteed convergence is an active area of current research; some algorithms that are already available (see Introduction) present good candidates for first implementation efforts.

As to the latter question, our results indicate that the answer is likely to be positive, although a number of solutions in the 3D system are expected to be quite different from those described here. Specifically, the sensitive skin will require an order of magnitude more sensors, and so different control schemes, with higher level of parallelism, are likely to appear. Real-time processing of contact information will require more intensive computations, especially when a significant number of the skin sensors are simultaneously in contact with obstacles - for example, when the arm is attempting to squeeze through a narrow opening between two walls. Finally, the algorithms for local step planning can be readily generalized for the 3D case.

Acknowledgements

The authors would like to thank Ernest Kent, Wyatt Newman, and Thom Warmerdam, all from Philips Laboratories, for good advice and help in experiments with the path planning and obstacle avoidance system.

References

1. J.T. Schwartz and M. Sharir, On the "Piano Movers" Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds, *Advances in Applied Mathematics*, 1983, p. 298 - 351.
2. C.K. Yap, Algorithmic Motion Planning. *Advances in Robotics, vol. 1: Algorithmic and Geometric Aspects*, (J.T. Schwartz and C. Yap, Eds.), Hillsdale, New Jersey: Erlbaum, 1987.
3. A. Elfes, Sonar-Based Real-World Mapping and Navigation. *IEEE Journal of Robotics and Automation*, June 1987.
4. L. Matthies and S.A. Shafer, Error Modeling in Stereo Navigation, *IEEE Journal of Robotics and Automation*, June 1987.
5. M.Winston, Opto Whiskers for a Mobile Robot, *Robotics Age*, Vol. 3 #1, 1981.
6. D.J. Balek and R.B. Kelly, Using Gripper Mounted Infrared Proximity Sensors for Robot Feedback Control, *Proc. 1985 IEEE Conference on Robotics and Automation*, St.Louis, Missouri, March 1985.
7. B. Espiau, J.Y. Catros, Use of Reflectance Sensors in Robotics Applications, *IEEE transactions on Systems, Man, and Cybernetics*, December 1980.
8. J.S. Schoenwald, A.W. Thiele and D.E. Gjellum, A Novel Fiber Optic Tactile Array Sensor, *Proc. 1987 IEEE Conference on Robotics and Automation, Raleigh, North Carolina*, April 1987.
9. V. Lumelsky, Effect of Kinematics on Motion Planning for Planar Robot Arms Moving Amidst Unknown Obstacles, *IEEE Journal of Robotics and Automation, Vol. RA-3 No.3*, June 1987.
10. Y.C. Chen, M. Vidyasagar, Some Qualitative Results on the Collision-Free Joint Space of Planar n-DOF Linkage. *Proc. 1987 IEEE International Conference on Robotics and Automation*, Raleigh, North Carolina, April 1987.
11. A. Flynn and R. Brooks, MIT Mobile Robots - what's next? *Proc. 1988 IEEE Conference on Robotics and Automation*, Philadelphia, Pennsylvania, March 1988.
12. R. Taylor, M. Mason, Sensor-based manipulation planning as a game with nature, *4th International Symposium Robotics Research*, August 1987.
13. V. Lumelsky, K. Sun, On the Motion of Simple 3D Arm Manipulators with Uncertainty. *Proc. 5-th Yale Workshop on Applications of Adaptive Systems Theory*, New Haven, May 1987.
14. V. Lumelsky, Continuous Motion Planning in Unknown Environment for a 3D Cartesian Robot Arm, *Proc. 1986 IEEE International Conference on Robotics and Automation*, San Francisco, 1986.
15. K. Sun, V. Lumelsky, Computer Simulation of Sensor-based Robot Collision Avoidance in an Unknown Environment, *Robotica, Vol.5*, 1987.
16. A. Flynn, Combining Sonar and Infrared sensors for Mobile Robot Navigation, *International Journal of Robotics Research, Vol.7, No.6*, December 1988.
17. *Ultrasonic Range Finders*, Polaroid Corporation, 1982.
18. G.L.Miller, R.A.Boie, and M.J.Sibilia, Active Damping of Ultrasonic Transducers for Robotic Applications, *Proc. IEEE International Conference on Robotics*, Atlanta, Georgia, March 1984.
19. D. Ballard, C. Brown, *Computer Vision* , Prentice-Hall, New Jersey, 1982.
20. I.J. Cox, Blanche: An Autonomous Robot Vehicle for Structured Environments, *Proc. 1988 IEEE Conference on Robotics and Automation*, Philadelphia, Pennsylvania, March 1988.
21. P. Horowitz, W. Hill, *The Art of Electronics*, Cambridge University Press, Cambridge 1980.

22. R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, 1983.
23. *Linear Databook*, National Semiconductor Corporation, 1982.

Appendix
The Sensor Circuit Schematic

The schematic of the sensor module is shown in Figure A. Incident light at a given sensor pair is converted into a current by the photo transistor OP1, that is selected by IC1, an analog multiplexer. This signal is then high pass filtered and amplified by IC2, an operational amplifier, which is then connected to IC3, the PLL (see Section 3.2)[23]. The VCO signal of the PLL is available on the exterior of IC3's package. This signal is connected to IC5, a one-shot timer that generates a clean pulse for transmission. The duration of this pulse is 5 µsec; the pulse is repeated at a rate of about 10kHz. The output of IC5 is boosted by the high current amplifiers T1 and T2, and is then switched to the appropriate IRLED by IC6, an analog multiplexer. The OP2 IRLED is pulsed 'on' at currents of 1 Ampere. Both multiplexers, IC1 and IC6 receive the same 4 bit address from the computer controlling the sensor system, resulting in the IRLED and the corresponding photo transistor being addressed as a pair.

In addition to the PLL, IC3 also contains the second multiplier that multiplies the VCO signal with the amplified photo transistor current from IC2 (the inputs of this second multiplier are marked C and D in Figure 3). The output of the second multiplier, available on the exterior of the IC3 package, is amplified and level shifted by IC4, an operational amplifier. This signal, marked 'analog out (Vout)' on Figure A, is proportional to the intensity of reflected infrared light from the IRLED, and serves as the proximity signal. The digital signal marked 'digital out' on Figure A, provides for an on-off type indication of the presence of an obstacle. The VCO frequency of IC3, is varied from one sensor module to another, allowing interference-free operation under the situation when two different modules are directly facing each other. This situation can occur when the angle between the two links is small. Then, although the light emitted from one module, is shining directly into the detector of some other module, it will not affect the response of the other module.

Adjustment of the individual sensor modules occurs by placing a 12x12 cm piece of white paper at the desired maximum sensing distance and adjusting the 50kΩ potentiometer at IC2 (adjusts sensitivity) and/or adjusting the 10kΩ potentiometer at the emitter of T2 (adjusts output power) until the desired full scale reading is achieved.

Target

Target

Target

Target

Target

Target more

Target

Target

Target

Target

Target

Target