

Proximity Sensing in Robot Manipulator Motion Planning : System and Implementation Issues¹

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

Abstract

In this paper, the problem of sensor-based path planning for robot arm manipulators operating among unknown obstacles of arbitrary shape is considered. It has been known that algorithms with proven convergence can be designed for planar and simple three-dimensional robot arms operating under such conditions. However, implementation of these algorithms presents a variety of hardware and algorithmic problems related to covering the robot arm with a "sensitive skin", processing data from large arrays of sensors, and designing lower-level algorithms for step-by-step motion planning based on limited information. This paper describes the hardware and the lower-level control algorithms of a system for robot motion planning in an uncertain environment, and summarizes the results of experiments with the system.

1. Introduction

Ongoing research in robotic motion planning encompasses two major trends. In one approach, also 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, and typically assumes an unchanging and static environment [7]. An overview of research in this area can be found in [1].

Another approach, also considered in this paper, assumes incomplete information about the environment. Such a situation takes place, for example, when the robot has no a priori knowledge about the environment, but is equipped with a sensor that notifies it of impending collision, or proximity to an obstacle. The area to be sensed can be large, as when ultrasonic range sensors [8] or stereo vision [9] is used, or it can be a small local area, such as when proximity sensors are used [4] (for instance, the optical sensors discussed in this paper). Other reported uses of light in proximity sensors include grasping applications [10] where optical sensors are mounted in the robot gripper, and tactile sensors in fiber optic arrays [11].

The problem of moving a 2-degree-of-freedom two- or three-dimensional manipulator in an environment with unknown obstacles of arbitrary shape can be reduced to the problem of moving a point automaton on the surface of a 2-dimensional manifold [2]. The algorithms resulting from this approach, called Dynamic Path Planning (DPP), guarantee convergence, and require the arm to "slide" along or follow the contour of the obstacle. Because the planning at every instance is done based only on local information, the approach can be used in a highly unstructured and even time varying environment.

To realize such algorithms, the robot arm manipulator must have the ability to sense the presence of an obstacle with every point of its body and to identify the points of the body that are in physical or proximal contact with the obstacle. To develop this capability, a sensitive "skin" is needed that would cover the whole body of the arm. Such a skin would be similar, in the case of physical contact, to the skin of humans or animals. In the case of the proximal contact, it would form the kind of aura similar to the hairs on the legs of some insects. Realization of such sensing capabilities encompasses a variety of hardware and data processing issues that, to our knowledge, have not been addressed before.

This paper addresses the hardware and lower-level control issues related to the DPP approach, and, specifically, to the problem of instrumenting a robot arm with sensitive skin and giving it the capability to interpret the sensory data generated by the skin, for the purposes of obstacle avoidance. A scheme is suggested for obstacle following given the response of the proximity sensors on the arm. The considered approach is based on sensory data supplied by infrared proximity sensors that cover the whole body of the robot arm and thus form the arm's "sensitive skin". The results presented treat the problem in the context of motion planning for a planar robot arm.

Suppose that the robot body is covered by proximity sensors that sense a nearby obstacle. Assuming that a global path planning algorithm is in place (see Section 3), a local strategy is needed to generate, at the current position of the robot, its next step. What is needed is the local tangent to the obstacle, at the point of contact with the robot (here, "contact" refers to the fact of sensing an obstacle at some distance). Once this is known, the robot can move

¹ Supported in part by the National Science Foundation Grants DMC-8519542 and DMC-8712357, and a grant from North American Philips Corp.

along this local tangent to follow the contour. To generate the local tangent, information is needed as to which points of the robot body are experiencing the obstruction. Combining this information with approximate distance to the obstacle surface, one can assure that the robot's next move does not lead to collision and/or loss of contact with the obstacle.

Section 2 addresses the hardware issues, describes briefly available options, and justifies the choice of proximity sensors. Section 3 addresses the algorithmic issues of local step planning, and Section 4 summarizes the results of our experimental work.

2. Hardware issues; Proximity Sensor

2.1 General

Apart from providing the robot arm with information about an approaching obstacle, the sensor system should also indicate the location(s) on the arm body where the obstruction takes place. This suggests an array of distance sensors. A detection range of about five inches is considered to be adequate. The sensor should remain effective at shallow angles between the obstacle and the arm, and should have no "dead zones" on the arm body. An obstacle must be positively detected if it is located within the detectable range of the sensors, otherwise, if the dead zone comes in contact with an obstacle, a collision may take place. A brief survey of sensor options follows.

The first 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. 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, but these methods are generally limited to detection distances of less than one inch, and are 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, a burst of ultrasonic energy is transmitted, reflected from an obstacle and then received, making use of the time of flight for distance measurement [5]. 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.

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 [5,8]. This effect can be only partially compensated by the active damping of the transducer [6]. 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, it 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, a radiation of shorter wavelength can be used, such as light. The wavelength of visible light is in the range of hundreds of nanometers, tending obstacles to appear matte, scattering light equally in all directions [3]. If the matte surface to be sensed 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. This insensitivity to obstacle orientation can be exploited to detect the presence of objects. One drawback to using light for sensing is that objects of dark color may not be sensed. Also, optical mirrors will continue to exhibit specular reflection, possibly resulting in a lack of returning radiation.

In the developed sensor system, the amplitude of the reflected light is used for proximity measurement. An alternative approach, often used in auto focus cameras, is to emit a beam of light and use triangulation to determine distance. The advantage gained by 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.

2.2 The Sensor Hardware

Based on these considerations, a sensor system with infrared proximity sensors was chosen. The transmitted light is infrared (IR), at 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, sixteen sensor pairs, each consisting of a phototransistor and an infra red 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 1, and its functional block diagram - in Figure 2; its more complete schematics is described in the Appendix.

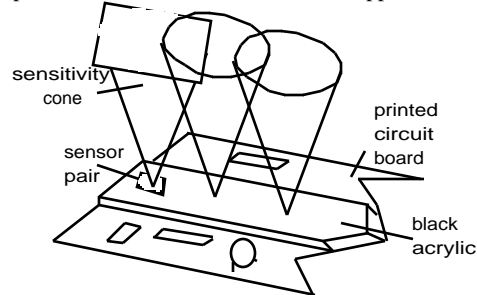


Figure 1. A sketch of the sensor module.

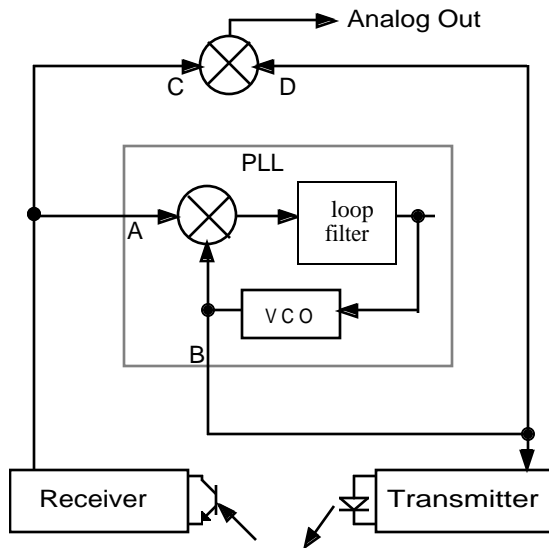


Figure 2. Sensor module functional block diagram

The transmitter block amplifies the output of the voltage controlled oscillator (VCO) (marked B in Figure 2). 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 phototransistors. The current generated in a phototransistor 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 phototransistor 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 five inches 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. A description of the PLL operation can be found in [13].

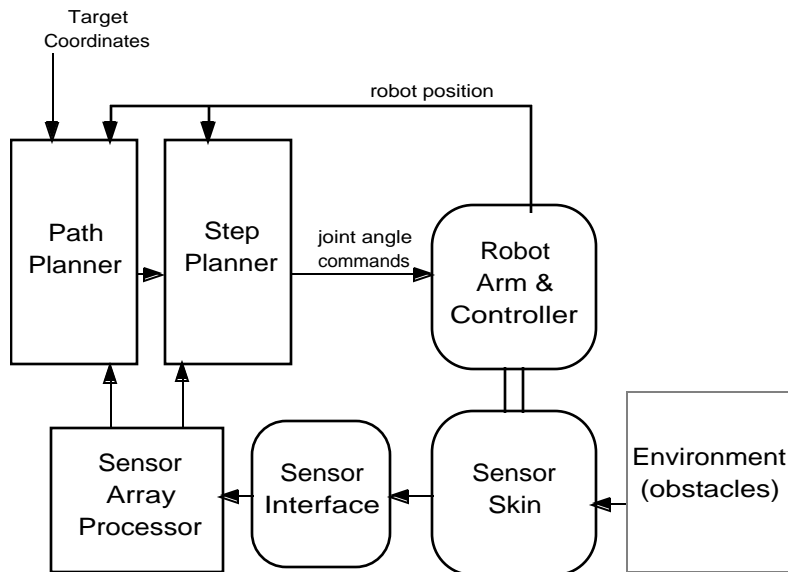


Figure 3. Information flow diagram of the path planning system

The information flow diagram, Figure 3, shows the interaction and hierarchy of the various components in the obstacle avoidance system; in the diagram, the software blocks are shown by straight rectangles, and the hardware blocks by curved rectangles. The three software blocks, 'Path Planner', 'Step Planner', and 'Sensor Array Processor' are implemented in two microcomputers (see Section 2.3). The block 'Sensor Interface' is described below (Figure 4).

The Analog Out signal (Figures 2 and 4) is proportional to the reflected signal, and is used for distance detection. Each sensor module produces its corresponding output, which is selected 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. In the implementation, an extra bit is used to trigger the analog to digital converter (ADC) into conversion. 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 μ sec), a byte of range data is read from the ADC by the sensor controlling microcomputer, and transformed into the distance between the obstacle and the corresponding sensor pair.

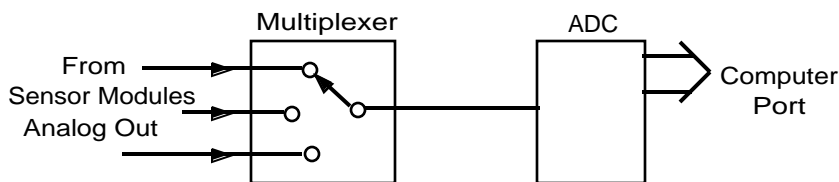


Figure 4. Detail of the Sensor Interface

A typical sensor response is shown in Figure 5 : the test object consists of a 2x2 inch piece of white paper. Typically, the sensor sensitivity becomes insufficient beyond a distance of five inches 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 2x2 inch produce approximately the same response. This results from the fact that at distances at which the sensor operates (about five inches), 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. Increase in obstacle size in this invisible region does not affect sensor reading. Note that the obstacle may then be sensed by another sensor pair (with a different invisible region).

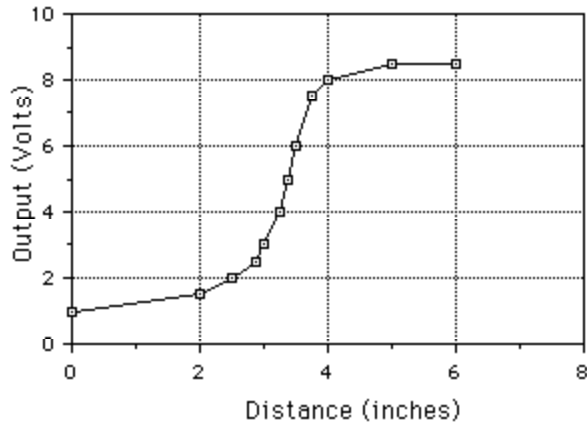


Figure 5. A typical response of the infra-red sensor.

photo here 13

Figure 6. Sensor module

The sensor module is designed so that all optical components and instrumentation are on one printed circuit board. A sketch of the module appears in Figure 1, and a photograph of a part of the actual module is given in Figure 6; note two sensor pairs highlighted by the reflection in the middle of the black acrylic.

2.3 The Computer System

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 backplane, 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, the Sun and the 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 3. As long 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.

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.

3. Step Planning and Contour Following

3.1 General

In addition to the information on contact with an obstacle provided by the proximity sensor, information on the location of the obstacle relative to the arm is also available. It will be shown now how this data is used for contour following. Since the contour following algorithm works in conjunction with the global path planning algorithm [2], the latter is first briefly discussed below.

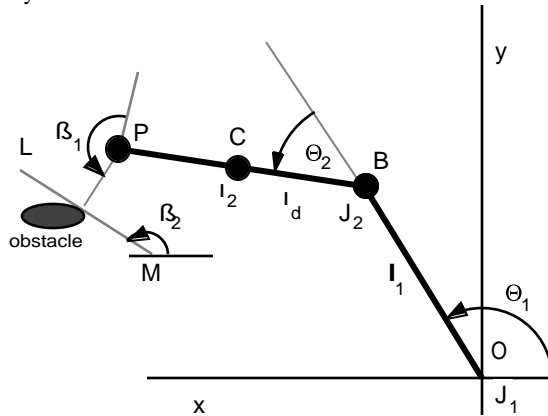


Figure 7. A sketch of the 2-link arm with revolute joints.

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 [2]), its main idea can be presented in terms of the planar configuration space, Figure 8. Assume that the whole robot body is covered with tactile sensors so that any point of the body can detect contact with an obstacle, and the location of that point is known. [Actually, the algorithm can work with any type of sensing].

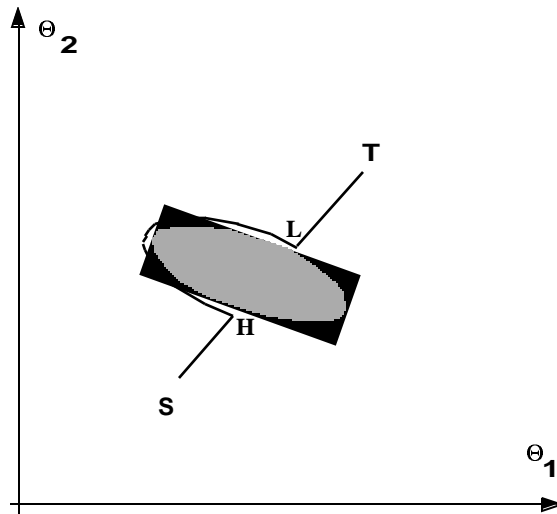


Figure 8. Configuration space of robot arm of Figure 7.

In the path planning algorithm, the automaton moves directly to T from its start, S, 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.

To realize contour following required by the path planning algorithm, a 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.

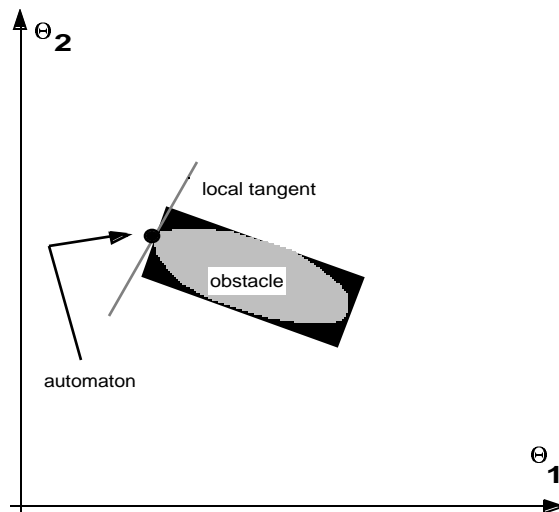


Figure 9. Using the local tangent for contour following.

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. Hereafter, unless otherwise noted, "local tangent" refers to the local tangent to the obstacle boundary in configuration space, as indicated in Figure 9.

3.2 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.

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. Now we consider each of the types.

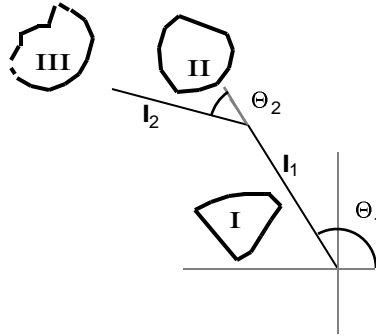


Figure 10. Obstacle types.

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:

$$\begin{aligned} c_x &= l_1 \cos(\theta_1) + l_d \cos(\theta_1 + \theta_2) \\ c_y &= l_1 \sin(\theta_1) + l_d \sin(\theta_1 + \theta_2) \end{aligned} \quad (1)$$

Take total derivatives:

$$\begin{aligned} dc_x &= -l_1 \sin(\theta_1) d\theta_1 + dl_d \cos(\theta_1 + \theta_2) - l_d \sin(\theta_1 + \theta_2) d\theta_1 - l_d \sin(\theta_1 + \theta_2) d\theta_2 \\ dc_y &= l_1 \cos(\theta_1) d\theta_1 + dl_d \sin(\theta_1 + \theta_2) + l_d \cos(\theta_1 + \theta_2) d\theta_1 + l_d \cos(\theta_1 + \theta_2) d\theta_2 \end{aligned} \quad (2)$$

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

$$dl_d = \frac{[-l_1 \cos \theta_1 - l_d \cos(\theta_1 + \theta_2)] d\theta_1 - l_d \cos(\theta_1 + \theta_2) d\theta_2}{\sin(\theta_1 + \theta_2)} \quad (3)$$

and

$$dl_d = \frac{[l_1 \sin \theta_1 + l_d \sin(\theta_1 + \theta_2)] d\theta_1 + l_d \sin(\theta_1 + \theta_2) d\theta_2}{\cos(\theta_1 + \theta_2)} \quad (4)$$

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

$$\begin{aligned} & \left[\cos(\Theta_1 + \Theta_2) \left(-l_1 \cos \Theta_1 - l_d \cos(\Theta_1 + \Theta_2) \right) \right. \\ & \left. - \sin(\Theta_1 + \Theta_2) \left(l_1 \sin \Theta_1 + l_d \sin(\Theta_1 + \Theta_2) \right) \right] d\Theta_1 \\ & = l_d \left[\cos^2(\Theta_1 + \Theta_2) + \sin^2(\Theta_1 + \Theta_2) \right] d\Theta_2 \end{aligned} \quad (5)$$

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

$$\begin{aligned} \frac{d\Theta_2}{d\Theta_1} &= \frac{1}{l_d} \left(-l_1 \cos \Theta_1 \left(\cos(\Theta_1 + \Theta_2) \right) - l_1 \sin \Theta_1 \left(\sin(\Theta_1 + \Theta_2) \right) - l_d \right) \\ &= - \left(\frac{l_1}{l_d} \left\{ \cos \Theta_1 \left(\cos(\Theta_1 + \Theta_2) \right) + \sin \Theta_1 \left(\sin(\Theta_1 + \Theta_2) \right) \right\} + 1 \right) \end{aligned} \quad (6)$$

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

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

Type III are those obstacles that obstruct the arm wrist or elbow (points P and B, Figure 7). To find the local tangent in this case, the inverse Jacobian of the arm is used [12]. 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:

$$\begin{aligned} d\Theta_1 &= \frac{l_2 \cos(\Theta_1 + \Theta_2) dx + l_2 \sin(\Theta_1 + \Theta_2) dy}{l_1 l_2 \cos \Theta_2} \\ d\Theta_2 &= \frac{\left[-l_1 \cos \Theta_1 - l_2 \cos(\Theta_1 + \Theta_2) \right] dx - \left[l_1 \sin \Theta_1 - l_2 \sin(\Theta_1 + \Theta_2) \right] dy}{l_1 l_2 \cos \Theta_2} \end{aligned} \quad (8)$$

Sliding of the wrist P along the obstacle corresponds to its moving along the line segment LM, Figure 7. Define β_3 as the angle between the line perpendicular to link 2 and the line from P to the obstacle, and β_4 as the angle between the line LM and the positive x-axis. Then,

$$\begin{aligned} \frac{dy}{dx} &= \tan \beta_2 \\ \text{or} \\ dy &= dx \tan \beta_2 \end{aligned} \quad (9)$$

where $\beta_2 = \Theta_1 + \Theta_2 + \beta_1 - \pi$. Substituting the expression for dy from (9) into (8), obtain the ratio $d\Theta_2 / d\Theta_1$:

$$\frac{d\Theta_2}{d\Theta_1} = - \left(\frac{l_1 \cos \Theta_1 + l_2 \left[\cos \Theta_1 \cos \Theta_2 - \sin \Theta_1 \sin \Theta_2 \right] + \left[l_1 \sin \Theta_1 + l_2 \left[\sin \Theta_1 \cos \Theta_2 + \cos \Theta_1 \sin \Theta_2 \right] \right] \tan \beta_2}{l_2 \left[\cos \Theta_1 \cos \Theta_2 - \sin \Theta_1 \sin \Theta_2 + \left[\sin \Theta_1 \cos \Theta_2 + \cos \Theta_1 \sin \Theta_2 \right] \tan \beta_2 \right]} \right) \quad (10)$$

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

$$\frac{d\Theta_2}{d\Theta_1} = - \left(\frac{\frac{l_1}{l_2} + \cos \Theta_2 + \sin \Theta_2 \tan(\Theta_2 + \beta_1)}{\cos \Theta_2 + \sin \Theta_2 \tan(\Theta_2 + \beta_1)} \right) \quad (11)$$

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.

3.3 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.

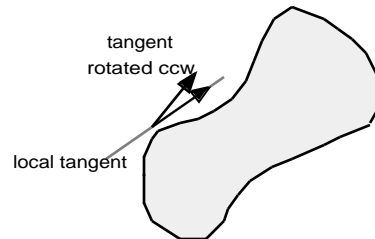


Figure 11. Configuration Space: ccw rotation of local tangent causes the robot to move away from the obstacle.

If the chosen local direction is "left", 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 safe rotation within one step - that is, such that it guarantees that no collision takes place after the rotation is completed - is determined by the distance at which the obstacle is detected and by the arm geometry. In other words, for this idea to work, the distance cannot be zero, which dictates the use of proximity sensors. 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 can be 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 distance to the obstacle is at some preset nominal value.

The proportional feedback in the sensor-based control loop results in constant distance tracking between the arm and the obstacle, thus improving contour following. The proportional feedback gain, K_p , is expressed in units of degrees per error in volts; here, the error is defined as the difference between the desired reference level of 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.

Several values of the proportionality constant K_p were tested in our experiments. A wide range of values of K_p was 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. For more detail, refer to Section 4.

If more than one sensor pair senses one or more obstacles simultaneously, more than one local tangent will be calculated. Then, one of these tangents is selected for planning the next step. The following two examples illustrate this point.

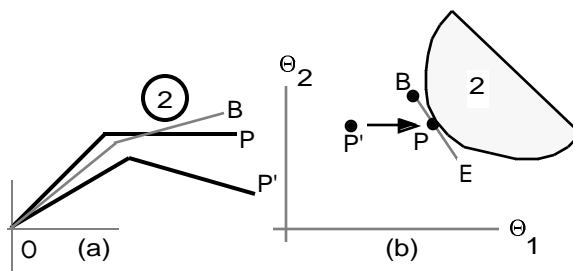


Figure 12. Arm interaction with a single obstacle: a) work space, b) configuration space.

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.

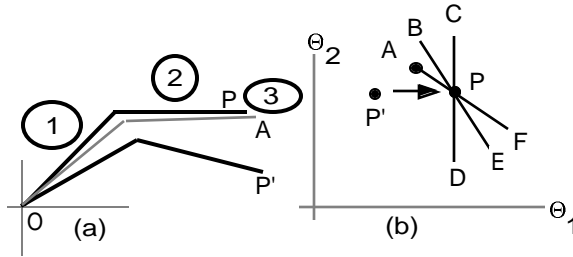


Figure 13. Arm interaction with multiple obstacles:
a) work space, b) configuration space.

2. Interaction with multiple obstacles.

For the case with multiple obstructions, suppose the robot arm is moving in the direction of increasing Θ_1 , Figure 13b. 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 does not necessitate 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.

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.

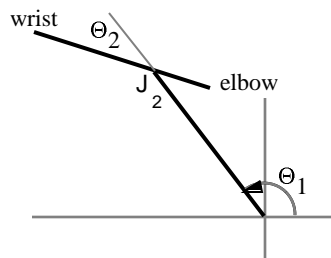


Figure 14. Example of an extended elbow on link 2.

4. 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 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 2.3. Combined with the DPP global path planning procedure, the two algorithms described above - for step planning and local tangent calculation - proved effective in accomplishing the path planning tasks. In the experiments with various combinations of obstacles, no contact has ever been made with the obstacles, so no collision occurred. No *a priori* information about the obstacles or about the shape of the arm links was given to the robot. Path planning was accomplished based only on the on-line information from the sensor system. In some experiments, the target position was not reachable because of interference with obstacles, and the system successfully concluded that this was indeed the case. Since the PUMA sample rate is 36 msec, and the longest operation in our system takes 20 msec (see Section 2.3), the required data processing fit rather easily into the real-time operation.

In the experimental setup shown in Figure 15, the path of the arm to its target position is obstructed by a rectangular 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. If the robot arm moves unobstructed, its current location in Figures 16a and 16b is connected to its previous location by a thin line. If the robot arm is following the contour of an obstacle, its location is marked by a larger dot.

The experiment documented in Figure 16a consists of two parts. In the first part, the robot is commanded to move from the origin (intersection of the two axes) to the point

photo here 13

Figure 15. Experimental setup with a rectangular obstacle.

cutout of configuration space 17

Figure 16. Configuration space of two experiments with the 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 meant to be a single thick line)

$\Theta_1 = 70^\circ$, $\Theta_2 = 70^\circ$. The second part of the task is then to move back to the origin. In each part the arm follows a different section of the obstacle; when combined, the complete image of the obstacle in the configuration space is produced. In this experiment, the proportionality gain, K_p (see Section 3.3) was set at a value $K_p=10$ degrees/volt chosen from a relatively wide range of values found to produce an acceptable behavior. One can see from Figure 16a that the actual motion of the arm along the obstacle is quite smooth. An example of an unacceptable proportionality gain is presented in the second experiment, shown in Figure 16b. Here, $K_p=25$ degrees/volt; only the second part of the task (see above) is presented. Note that the arm motion along the obstacle becomes somewhat jerky, although it still follows the obstacle contour. Another experimental setup, with a non-convex C-shaped obstacle, is shown in Figure 17.

photo here 15

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

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. 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.
2. V. Lumelsky, Dynamic Path Planning for a Planar Articulated Robot Arm Moving Amidst Unknown Obstacles, Automatica, Sept 1987.
3. D. Ballard, C. Brown, Computer Vision, Prentice-Hall, New Jersey, 1982.
4. M. Winston, Opto Wiskers for a Mobile Robot, Robotics Age, Vol. 3 #1, 1981.
5. Ultrasonic Range Finders, Polaroid Corporation, 1982.
6. G.L. Miller, R.A. Boie, and M.J. Sibilja, Active Damping of Ultrasonic Transducers for Robotic Applications, Proc. IEEE International Conference on Robotics, Atlanta, Georgia, March 1984.
7. 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.
8. A. Elfes, Sonar-Based Real-World Mapping and Navigation. IEEE Journal of Robotics and Automation, June 1987.
9. L. Matthies and S.A. Shafer, Error Modeling in Stereo Navigation, IEEE Journal of Robotics and Automation, June 1987.
10. 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.
11. 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.

12. R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control, MIT Press, 1983.

13. P. Horowitz, W. Hill, The Art of Electronics, Cambridge University Press, Cambridge 1980.

Appendix

The Sensor Schematic

The complete schematic of the sensor module is shown in Figure A. Incident light at a given sensor pair is converted into a current by the phototransistor 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 2.2). 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 μ sec, and 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 receive the same 4 bit address from the computer controlling the sensor system, resulting in the IRLED and the corresponding phototransistor being addressed as a pair.

In addition to the PLL, IC3 also contains the second multiplier that multiplies the VCO signal with the amplified phototransistor current from IC2 (the inputs of this second multiplier is marked C and D in Figure 2). The output of the multiplier, available on the exterior of IC3's 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 infra red light from the IRLED, and serves as the proximity signal. A typical sensor response, is shown in Figure 6 : the test object consists of a 2x2 inch piece of paper. IC3 also has additional circuitry that thresholds the output of the second multiplier. This digital signal is marked 'digital out' on Figure A, and provides for an on-off type indication of the presence of an obstacle.

The repetition rate of IC5, the one-shot timer mentioned above, 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, the light emitted from one module, although shining directly into the detector of some other module, will not affect the response of the other module.