Laboratory Research in Autonomous Sensory Perception

Final Report for Research Performed Under ONR grant no. N00014-91-J-1621
June 1, 1991 — September 30, 1991

John S. Bay
Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0111

(703) 231-5114
FAX: (703) 231-3362
EMAIL: bayj@vtvm1.cc.vt.edu

Summary

This report documents the research performed in autonomous sensory perception through automatic sensor-based decision-making in a robot. The robot's task is to touch simple curved surfaces and estimate their geometric parameters in a *completely autonomous fashion* with inexpensive sensors. That is, after initial contact, it is to intelligently explore the object so as to understand its shape as quickly as possible.

Equipment funded under the grant included several simple sensors and sensor parts, such as a wrist-mounted force/torque sensor, a track-ball transducer, and data acquisition electronics, as well as the high-speed computer host which acts as controller. This equipment was successfully integrated with a Merlin 6540 industrial robot which was programmed to maintain force-controlled contact with miscellaneous curved objects. Also funded was one month of Principal Investigator time and three months of graduate assistant support.

The system properly executed the exploration procedure, which is based theoretically upon the estimator uncertainty measure provided by a Kalman filter covariance matrix. Initial results show fast estimation of spherical surfaces with slower estimates of planar surfaces. Since automated exploration of the spherical objects produced faster estimates than manually programmed exploration, it is judged that the algorithm under test performed well in an actual experimental trial. Unexpected encouraging results were also obtained in the possible application of the autonomous learning algorithm for other tasks, such as the fully autonomous self-calibration of the robot arm itself.

Poor or incomplete results were obtained in the distinction of closely similar objects and for the estimation of surface curvatures. These problems are partly attributed to mechanical factors in the force-controlled contour following process. These factors will be separately addressed by graduate thesis projects currently underway.

Objectives

Funding for this project was provided for two purposes: *i)* to provide the experimental apparatus necessary for concentrated future study of autonomous robotic sensory perception under uncertain environments, and *ii)* to support work toward the experimental proof-of-concept of an autonomous machine learning technique based on the Kalman filtering of sensor data.

The first goal is intended to provide the long-term benefit of the experimental apparatus needed in subsequent studies. The second is support for the initial trials of those studies. The proof-of-concept is necessary before the continued theoretical development of the autonomous sensory perception and learning procedures discussed in the next section can proceed.

To satisfy these objectives, this report describes the completed work in installation and programming of the equipment. It also describes the results, successful and unsuccessful, of the experimentation. The experiments involve the automatic, unguided and unsupervised exploration of geometric objects with simple robotic sensors for the purpose of geometric modeling. It will be revealed that while the learning and sensory perception technique appears to perform better than anticipated, the mechanical difficulties of robotic touch-control represent obstacles to certain operations.

Motivation

The motivation for this experimental proof-of-concept comes from the theoretical work presented in [1]. This concept is designed to free an artificially intelligent machine or robot from human supervision during learning and manipulation tasks.

Briefly, the scenario proposed is a learning robot given the task of acquiring spatial data and estimating the geometric model of unknown, arbitrarily shaped objects. The technique enables the machine's controlling computer to compute a cost function which represents a measure of uncertainty in its estimates. This confidence measure then acts as a guide, "pointing" to workpiece features which are less confidently known and are therefore fruitful regions for continued tactual exploration.

The resulting system acts autonomously, teaching itself the necessary information-gathering maneuvers to uniquely describe the object's shape, position, and orientation. An analogous procedure is the study of cognition through visual cues. It has been shown that humans, when given the task of recognizing a familiar face, tend to gaze at clearly defined facial features in a predictable sequence. Often, this sequence is: eyes, nose, mouth, hairline, and back to eyes. Humans "know" how uncertainty in information can be resolved with efficient exploration using the available senses. The robotic system under study is to be given a similar capability for its touch senses.

On a more technical level, the perception technique is built on the Kalman filter estimation algorithm. The Kalman filter is essentially a recursive parameter estimator, here estimating the constant geometric parameters of an unchanging object. The attractive feature of the Kalman filter for this study is its "covariance update" equation.

The covariance of the parameter estimate is updated at each measurement interval in order to provide a measure of *un*certainty in the current estimate. If the covariance matrix is relatively large, the user knows that estimation is not yet reliable. Also, the filter equations then place more emphasis on incoming sensor data when computing an updated estimate. As the covariance decreases, the estimate is more reliable, so incoming data is de-emphasized; the parameter estimate has evolved from a long history of data and should not change much from individual sensor readings.

The learning sensory perception algorithm used here relies on the fact that the covariance matrix of uncertainty is an explicit mathematical function of the geometric coordinates of the sensor. In particular, the covariance depends on the coordinates of the contact point and the surface normal direction at the point of contact. These coordinates can be directly measured with simple sensors.

We provide the controller the capability to explore objects by two computations. First, we find the rate of change of a scalar counterpart to the covariance (in order to avoid matrix derivatives). This gives a measure of the time rate of change in uncertainty. Then we compute the *spatial* (directional) derivative of the result. Examining the directional derivative for extrema, we can pinpoint the direction in which the sensor should travel in order to *maximize* the *decrease* in our uncertainty. Naturally, rapidly decreasing uncertainty represents rapidly increasing rates of learning.

Because the algorithm provides the system with the best directions in which to explore, the user need only start with a random position, with the sensors in contact with the object, and let the computations take over.

Although our scenario provides only touch senses, the computations would also work with visual or any other spatial data. Furthermore, although we seek only fixed geometric parameters of objects, it is also possible to apply the technique to other learning problems, as we found in this research (see Results).

The mathematical details of the operation are discussed in [1] and [2]. The technique is shown in those publications to work exceptionally well with simulated objects and sensor data, but prior to the grant period, no experimental verification had been attempted.

Apparatus

The first step in the experimentation is the installation, testing, and programming of the robotic apparatus. The robot used was a Merlin 6540 industrial robot, on indefinite-term loan to the investigator from Martin Marietta Aero and Naval Systems, Baltimore, MD. It weighs 1000 lbs., has a 50 lb. payload, a 40 in. reach, repeatability of 0.001 in., and absolute accuracy of 0.05 in. Its controller has an on-board computer with a 68000 CPU board in a Versabus card cage.

Although the robot is provided with its own specialized robotics version of the BASIC language, such programming is slow. The control interval for this standard configuration is a minimum of 128 ms.

For the estimation and other computations required for real-time experiments, a high-speed host was installed. Under project funding, bus-conversions, communications, and control hardware and software were implemented so that a 33 MHz, S0386-based IBM-compatible computer (hereafter "PC") could take over the control duties for the robot. Through a shared dual-port memory, individual robot axes were independently controlled through programs written in the C-language. With this programming, all functions of the manufacturer's controller were handled with cycle times of 4 ms. This is the minimum possible time, due to the actual motor driver controls. Further processing and apparatus will later increase the time delays necessary for computation.

For contact sensing and measurement of the surface normal direction, a wrist-mounted force/torque sensor was purchased and installed. This sensor is built with strain-gage transducers and provides signal-conditioned, formatted data on the three axes of applied force and three axes of applied torque. If the contact probe is constructed to be frictionless (to be discussed below), the direction of contact force is

coincident with the surface normal direction. In addition, the force sensor provides the signal by which surface contact is maintained, since nonzero force indicates contact, and zero force indicates loss of contact.

The force/torque sensor data is transmitted to the PC every 10 ms. This provides a new lower limit on force control computation time.

In order to implement frictionless probes, two specialized end effectors were fabricated, illustrated in Figure 1. The first is a simple bracket which holds a ball-point pen tip or a felt-tip plotter pen. The small areas of contact provided by pen tips reduce sliding friction and provide the added benefit of actually drawing the line of travel on the probed object. The second is an industrial track-ball transducer, which contains a rolling sphere of known radius (2.5 in.). Since the ball rolls on the surface, friction is negligible. However, because the possible contact locations on the ball's surface are infinite in number, the force sensor must be used in conjunction with the track-ball so that the force direction uniquely defines the contact point on the ball.

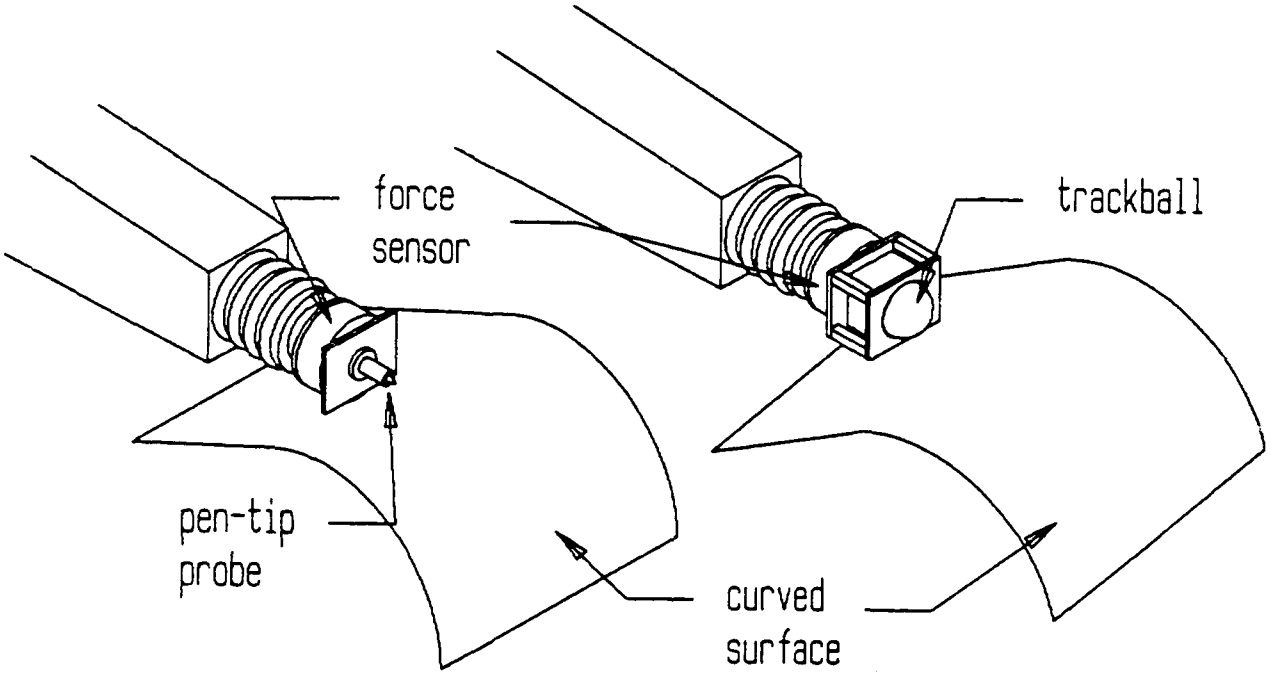


Figure 1. The two end effector probes fabricated and used for this project. The pen tip provides a point contact, while the track-ball provides frictionless contact and arclength information.

A further anticipated use for the track-ball is for the measurement of curvature of the surface of the objects. If the ball were rolled along a surface contour, the traversed arclength measured on the ball would equal the arclength on the surface. Then with the computed angular change in surface normal direction, curvature could be computed for any point on the contour as the ratio of incremental normal direction angle to incremental arclength traversed. This concept was not tested in the laboratory due to difficulties in getting the highly polished ball to roll smoothly over the smooth surface without slipping or breaking contact.

The track-ball data, in the form of two-channels of pulses, was acquired with a Keithley modular data acquisition system with a pulse-counting module. The system was interfaced with the PC through a parallel data port (the force/torque sensor data was serial). Figure 2 shows the entire experimental apparatus and identifies the portions which were bought and/or installed with grant funds.

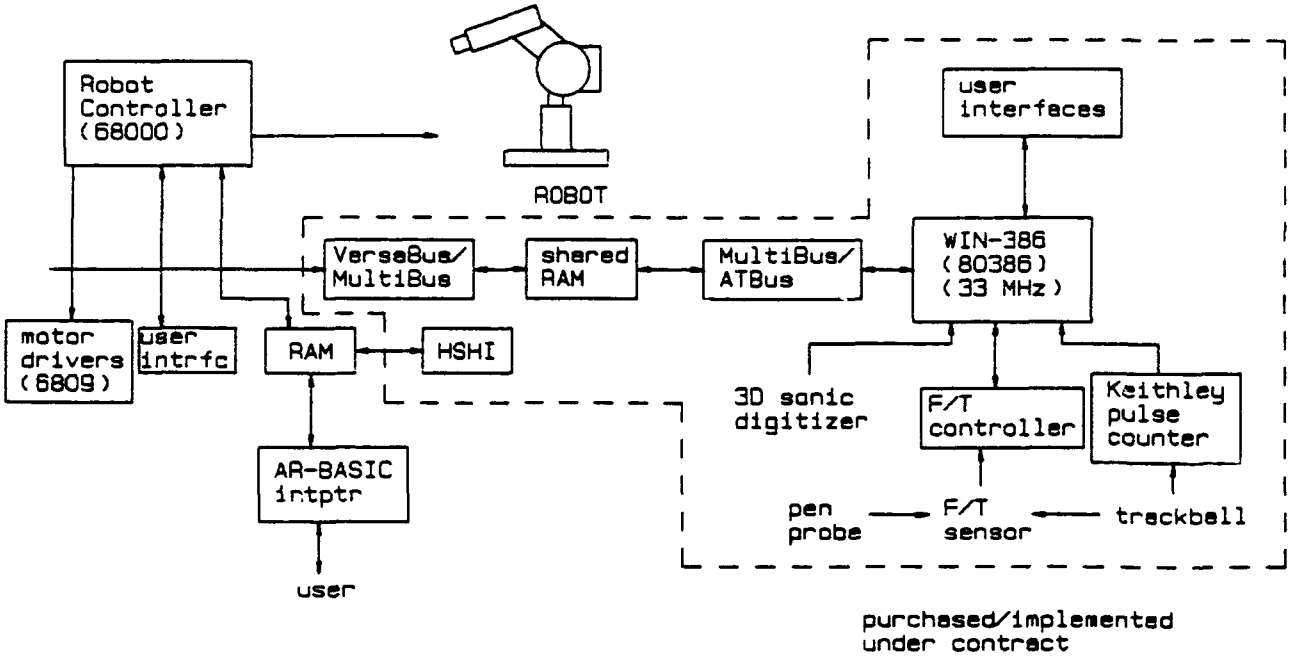


Figure 2. Block diagram of the hardware configuration assembled and used. The equipment and assemblies indicated within the dotted lines are the devices bought and/or installed with grant funds.

Results

EQUIPMENT and OPERATION

The apparatus to be installed on the project was tested with mostly minor difficulties. The only mechanical and computational difficulty in the operation of the hardware affected the force-controlled tracking procedures. However, since these procedures were critical to the data acquisition required by the algorithm, they represented significant obstacles to experimentation.

Force-controlled tracking requires continual adjustments to the robot motors in response to the force sensor output. In theoretical and applied force control, the commanded signal for a robot is conventionally the *torque* provided by the joint motors. The Merlin robot uses stepper motors, which make position servo loops particularly simple, but which *cannot* be torque-controlled. The result is the feedback of a force error signal to the position servo control loop. Such unavoidable strategy is not mathematically advised, but has been known to work with sufficient compliance at the contact.

Our contacts, though, being specifically constructed for low friction, also have very low compliance. That is, the probe tips and the surface of contact are both

relatively hard materials. Modeling contact force interactions as spring-like, we would have to model our contact as a *very* stiff spring. Very small changes in displacement react with very large changes in force. Because any mechanically controlled system is subject to positional inaccuracies and vibration, small displacement errors result in high frequency, high amplitude force measurements. These, in turn, enter the positional servo loops and result in even further displacement disturbances. The cycle of positional error/force feedback is unstable.

Our approach to the problem is to smooth the force measurements with a saturating gain to limit their effect on the probe position without hindering their effect on force control. We must maintain a minimum threshold of force to ensure contact with the surface. The result, however, is a trade-off with tracking performance. We found that we had to not only scale the force measurements, but also program a tracking behavior which was relatively slow. With reasonable tracking speed, the instability in the force loop produced a "bouncing" behavior. With this bouncing behavior, the contact is periodically lost. Such motion can still provide sufficient point-probe data, but the track-ball could not roll continuously.

AUTOMATIC LEARNING, PERCEPTION, and ESTIMATION

Subject to the limitations imposed by the above mechanical behavior, the intended experiments were conducted. Automatically acquired sensory data was sought from various shapes, including a plane, spheres of different radii, and a small sector of a spherical reflector with a large focal length and hence, small curvature.

The principal results are summarized in the series of Figures 3–7 below.

Figure 3 shows the worst performance of the algorithm among all trials. It shows the convergence of the geometric parameter estimate for the automatically chosen exploration path against that of a manually programmed path (a spiral). These are for exploration on a flat plane. By mathematical definition [1], as these cost functions approach *zero*, the expected value of the parameter estimates also approaches zero. Of course, the faster the curves approach zero, the faster the computer is "learning" the shape. From Figure 3 it is clear that the algorithm's choice of path does not "teach" the estimator as fast as that of a randomly chosen spiral path. Both estimates do eventually converge (giving the orientation of the plane in space).

The poor behavior of the learning algorithm is yet unexplained, but may be attributable to the flat geometry of the plane. Since the algorithm works by computing derivatives and searching for extrema, it may not find useful information on a plane, which has a constant spatial derivative. This may also explain the straight-line-segment nature of the automatically chosen path as seen in Figure 4.

Figure 5 shows similar cost function data for exploration of a surface which is known to be a sphere of diameter 7 inches. In this case, the cost function produced from automatic exploration indeed converges to zero well before the cost due to a manually programmed path. Repeated trials produced similar results.

Figure 6 shows another convergence comparison, but for an 18 inch diameter sphere. It can be seen here that automatic exploration produces high-confidence estimates in approximately half the number of steps taken in exploration. The paths taken over the experiment are depicted in Figure 7 as a contour plot. This plot shows the spherical surface from above, with contour lines and the two paths.

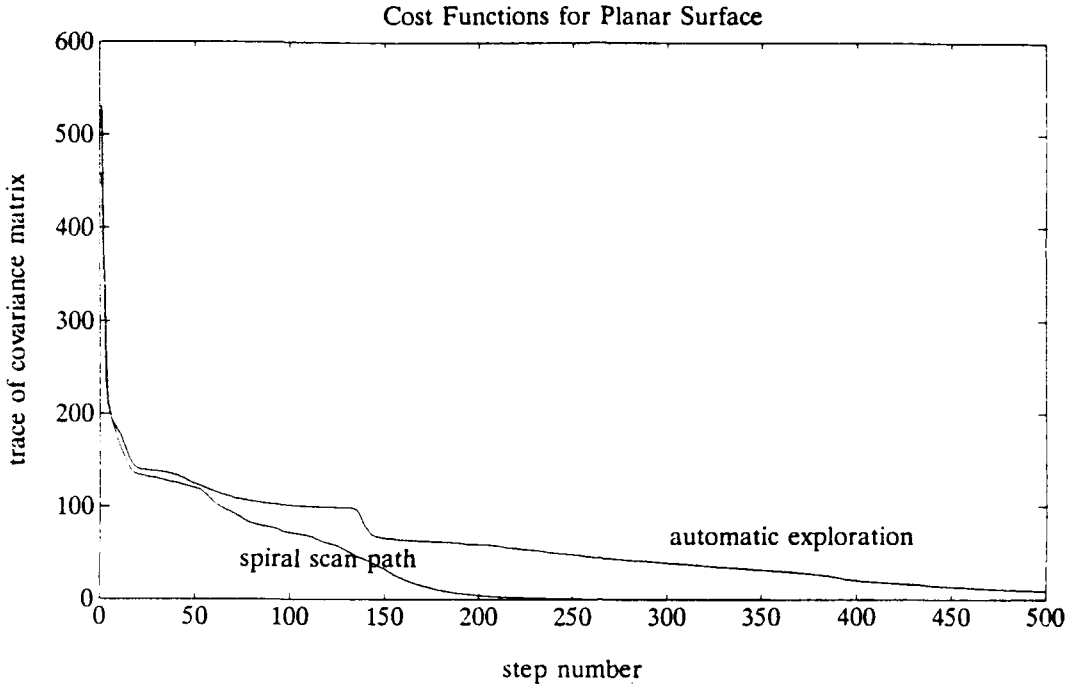


Figure 3. Convergence of cost to zero for exploration of a plane. As the plotted curves go to zero, so does the expected value of the error in the parameter estimate. The spiral scan path was arbitrarily chosen. The automatic exploration path was generated by the robot itself.

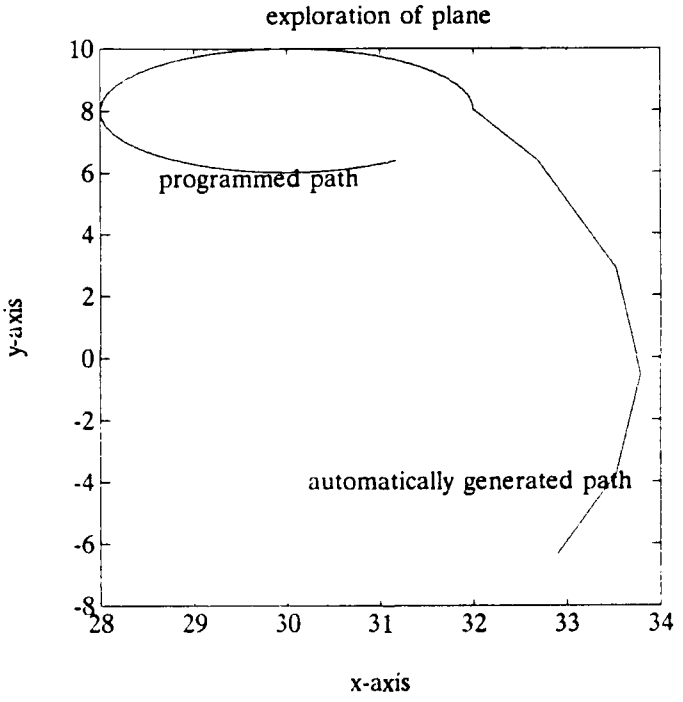


Figure 4. Paths taken on the $x-y$ plane by the pre-programmed exploration and by the automatic exploration. For exploration of a plane, the path taken actually is irrelevant. Identical surface features extend in all directions.

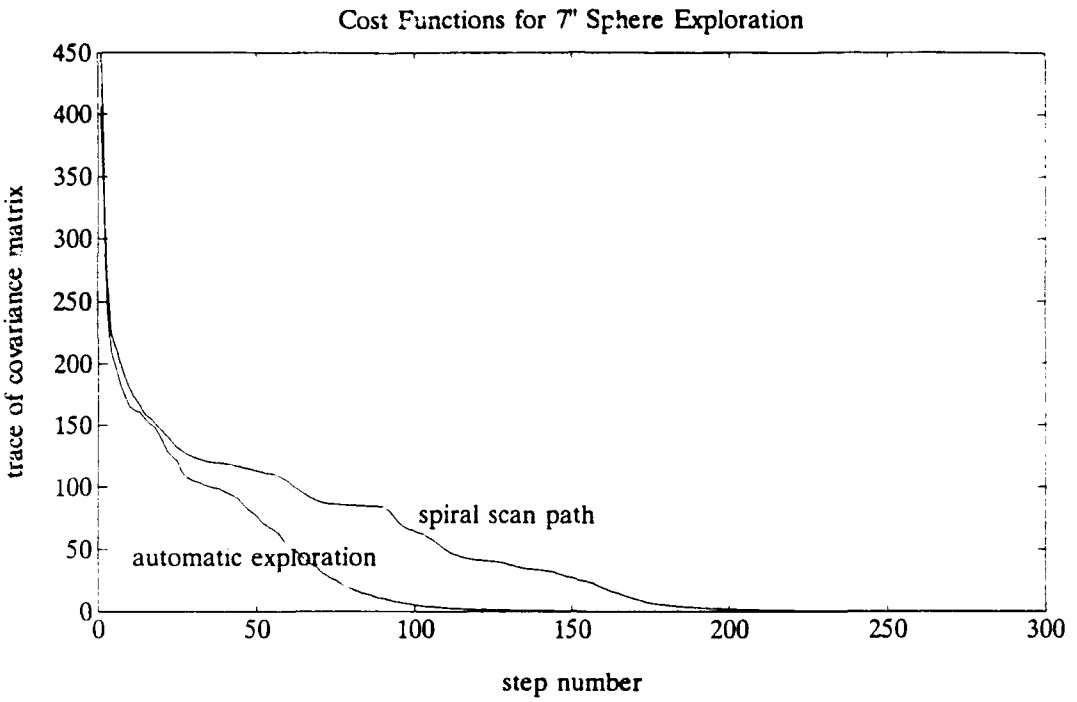


Figure 5. Convergence of the parameter estimate for exploration of the 7 inch (diameter) sphere. The automatic exploration algorithm performs significantly better than the programmed path.

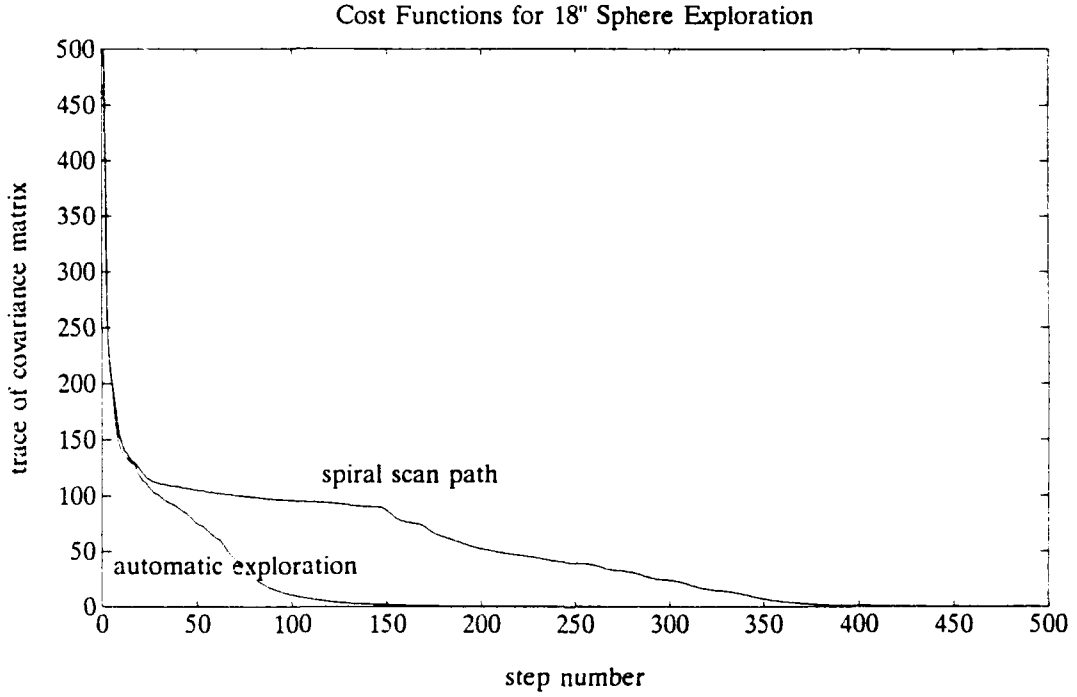


Figure 6. Convergence of the parameter estimate for exploration of the 18 inch (diameter) sphere. Again, the automatic exploration algorithm performs significantly better than the programmed path. Parameters are estimated accurately in approximately half the number of required steps.

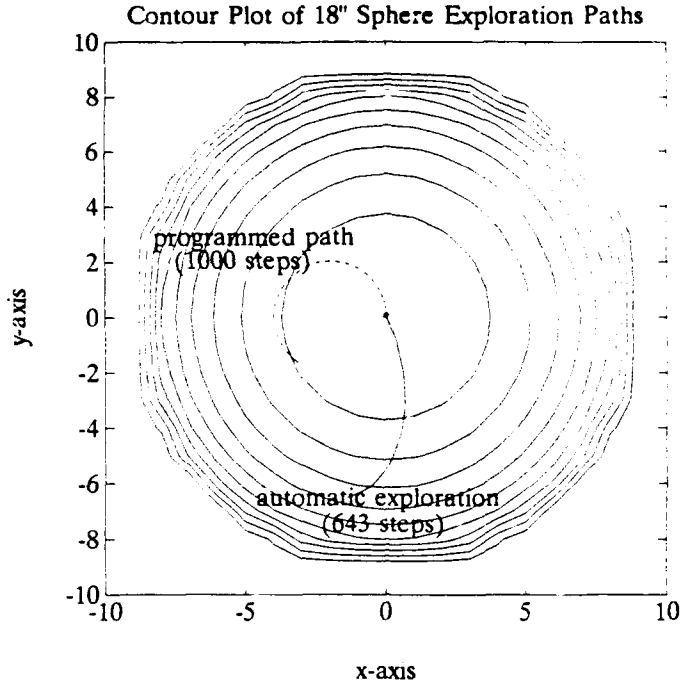


Figure 7. Contour plot showing the top view of the exploration paths on the 18 inch sphere. Because the automatic exploration path reached the surface of the table on which the half-sphere was mounted and had to be halted, the experiment resulted in fewer data points than the pre-programmed path.

In all the experiments, the cost function is shown to approach zero. As we have indicated, the Kalman filter algorithm shows that if this is the case, the expected value of the estimate *error* is zero. However, if we look at the convergence of the individual parameters, we get misleading results.

The filter is programmed to determine the nine parameters which define a general quadric surface (plane, cone, sphere, ellipsoid, etc.). For example, if we are estimating the shape of an ellipsoid, we need three parameters to specify the axis lengths, three to determine an offset from the coordinate origin, and three to determine a possible rotation in space about a fixed coordinate system.

We found by examining the parameter estimates that the location of the centers of the spheres were estimated quite well, while the radii were in error. Although at the close of the funding period the analysis of this error was incomplete, it is suspected that the symmetric nature of the spheres was a factor. In particular, because the sphere is symmetric about all three axes, the rotation parameters are *undefined.* It is not clear whether undefined parameters are consistently estimated. However, it is difficult to find known ellipsoidal objects on which to perform a quick test, so the question remains unresolved.

AN UNANTICIPATED RESULT

During the course of initial installation and programming, it was discovered that our information of the robot link lengths and sensor dimensions may have been inaccurate. If so, we would require re-calibration and identification of the true lengths to close tolerances.

The standard procedure for this type of calibration is to place the robot joints in a number of known angles and measure the coordinates of the end effector. From these coordinates and the kinematic equations of the arm, we can recursively identify the lengths with the Kalman filter. Since the filter and the autonomous exploration routine were already programmed on the PC, this approach was a likely choice. Mathematical analysis was conducted to derive the covariance cost function and its time- and spatial rates of change, as discussed above. Simulations showed that the algorithm under investigation enabled the robot to *put itself* in the best sequence of configurations such that the link length estimates converged as fast as possible.

It was found that a 3D ultrasonic coordinate digitizing table (property of the Spatial Data Analysis laboratory at Virginia Tech) could measure end effector position to within .01 in. by sensing a high frequency "ping" emitted by a wand. The wand could be held by the robot and moved throughout the sensing region of the sensors.

After receiving more accurate information from the robot manufacturer's documentation, the autonomous calibration experiment was deferred until completion of the shape-sensing experiments. Calibration experiments are planned for later in the Fall (1991).

Problems and Unresolved Issues

Some of the major difficulties encountered in the experimentation were discussed above, especially the contour tracking difficulties. This particular difficulty prevented us from resolving two questions from the originally proposed work: *i)* to what tolerance can the geometric parameters be measured?; and *ii)* can the track-ball sensor be used to measure curvature in different directions on a surface?

Tolerance of the fit for the estimated parameters is important for industrial application of robotic metrology systems. In manufacturing, model-makers and designers often need to know precisely the the dimensions of pre-existing parts. These measurements can be acquired with a device known as a component measuring machine (CMM). These CMM's have touch-probes which make contact with the objects at isolated points and provide precise dimensions. Many CMM's are very expensive, sometimes costing over $250,000. Part of the motivation for robotic shape sensing comes from the desire to reduce this cost.

One benefit of investigating CMM technology has suggested a possible solution to the force-tracking problem. Each CMM "gaging probe" is available for purchase separately, sometimes at low cost. One type of probe is based on the *linear variable displacement transducer* (LVDT), which has, in principle, infinite resolution. Furthermore, some of these devices can be fitted with a spring-return mechanism. The spring-return could be useful in alleviating the force-controlled tracking problem, since the spring itself would absorb some contact energy, while transmitting force unaltered. An LVDT was bought under this project for exactly this application at a cost of $235, but preliminary experiments showed that it was vulnerable to transverse (bending) damage by contact forces. More robust sensors at similarly low cost are being sought.

The advantage of curvature sensing would have applications in both metrology of curved surfaces and robotic manipulation. Since the curvature attributes of an object are useful for describing that object both qualitatively and quantitatively, measures of curvature would have multiple applications. High-tolerance machining and inspection and robotic grasping are obvious examples.

One original goal of the project was to use the track-ball for curvature estimation because it, together with the force sensor, could provide the ratio of normal direction to arclength change which defines curvature in a plane. Experiments were planned to determine the upper and lower bounds of object curvature which could be sensed with the project apparatus. As mentioned, though, the force control algorithm could not adequately prevent sensor bouncing. Some improvement was obtained from using very small applied forces, but this too was insufficient. The track-ball has a polished surface and required considerable contact force to prevent slippage on the object. At these higher forces, stability problems were exacerbated. This was particularly problematic for small radii spheres, since tracking on their surfaces would require quickly adjusting normal (probe axis) directions and rejection of the consequent force disturbances.

Future work will focus on curvature sensing with the help of re-designed end effectors. For example, if a spring-loaded linear displacement sensor could be incorporated in the track-ball mechanism, compliant contact could be achieved without sacrificing resolution. Perhaps with the subsequent measures of high and low surface curvatures, more complex object shapes, such as those with sharp edges and vertices, could be accommodated.

Publications

One publication has already resulted from this project. The computing facilities and sensor models obtained from the project were used in the following conference paper (copy attached):

> J. S. Bay, "An Estimator Based Exploration Engine for Autonomous Learning Systems," *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Charlottesville, VA, October 1991.

At least two other publications are anticipated. First, the investigator is the organizer and chair of a proposed invited session at the International Symposium on Robotics and Manufacturing (Santa Fe, NM, November 1992), at which the experimental results of the project will be presented. In addition, the application of the autonomous learning algorithm in arm calibration will be experimentally investigated and submitted to the 1992 IEEE/Robotics Society of Japan International Conference on Intelligent Robots and Systems, provided that experiments are concluded by December 1, 1991. That work, with its theoretical background, may also be submitted for publication in a refereed journal. These papers (yet to be assembled) will be co-authored with James G. Hollinger, the graduate research assistant who performed much of the equipment installation and most of the programming for the project.

All published work has or will be properly cited with ONR grant acknowledgements.

References

[1] Bay, J. S., "A Fully Autonomous Active Sensor-Based Exploration Concept for Shape-Sensing Robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 4, July/August 1991.

[2] Bay, J. S. and H. Hemami, "Dynamics of a Learning Controller for Surface Tracking Robots on Unknown Surfaces," *IEEE Transactions on Automatic Control*, vol. 35, no. 9, September 1990, pp. 1051-1054.

An Estimator Based Exploration Engine for Autonomous Learning Systems

John S. Bay
Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0111

Abstract – We examine the problem of an intelligent system which acquires knowledge through active sensing. Such a system can exert some control over the sensing apparatus in response to past sensor data and intelligent decision-making.

We consider a system consisting of a robotic arm, a fingertip sensor, and an intelligent controller. When presented with an unknown object to be identified, the system intelligently explores the object so that convergence to an accurate shape representation is as fast as possible. The system thus "seeks" out the features which it finds most useful in its sensing task.

The exploration technique estimates system unknowns with a Kalman filter. The covariance provided by the filter provides a measure of uncertainty which is processed to indicate the spatial direction in which it is predicted that the uncertainty will decrease at a maximal rate.

The technique is formulated in terms as general as possible, so that applications may be found for many forms of active sensing, and for learning problems not necessarily restricted to unknown shape representation, such as efficient problem solving or assembly.

I. INTRODUCTION: MACHINE PERCEPTION

It is widely known that in human perception experiments, certain features are more valuable, or "salient" than others. For example, tests which track eye gaze in face recognition experiments show that a person's eyes, mouth, and hairline are more important features than ears or cheeks. For the representation of 3-D objects with line drawings, it has been found that planar lines of curvature are most important [1]. When a person feels an object to determine its shape, the edges or other contours tend to be tracked by the fingertips [5].

This work was sponsored in part by the Office of Naval Research under grant no. N00014-91-J-1621.

It is intuitively clear that the salient features of objects depend on the nature of the sensing apparatus. For example, Braille characters are entirely dissimilar to written characters, and experiments to determine visual recognition rates of Braille characters and tactile recognition rates of Roman characters generally show poor performance. Furthermore, the psychophysical make-up of the test subject plays a significant role, since two subjects may undertake significantly different strategies when presented with the same complex recognition problem.

The study of artificially intelligent *machine* perception must include the same factors. A robot equipped with only tactile sensors is apt to "prefer" different features of an object than would a robot with only visual sensors. Whereas individual sensory information preferences result from complex psychological factors in humans, software and controller design might affect the value of sensor information in a robot. Certain medical imaging equipment, for example, uses reconstruction techniques based on Fourier descriptors. It is therefore programmed to gather its data in spiral scan paths. Different shape descriptors such as bicubic splines or Bezier patches generally operate on relatively regular rectangular grid points. The *motivation* for sensing therefore affects the *process* of sensing; in humans and in machines.

Curiously, a popular approach to machine perception design is the anthropomorphic approach. Although we have a long way to go before robots are either structurally or functionally equivalent to humans, human *behavior* has often been the programming goal.

We suggest in this paper a departure from this philosophy. We propose a technique by which a machine can use its own sensor apparatus and programming to acquire information automatically, in a manner to which it is uniquely suited. It can then "seek" the information it desires by influencing the control of its sensors. Just as a human will "look for" the critical feature of a problem, the machine will "explore" an object to expedite learning.

ISSN# 0-7803-0233-8/91 $1.00

When the system's hardware or software motivation changes, then so too will the path it chooses for its search. For example, replacing a tactile array sensor with a single force/torque probe should automatically result in a different path, since the information assimilated by each is different. Likewise, instructing the system to simply "find all the edges and vertices" will produce a different behavior than the command, "find the largest flat spot."

II. THE KALMAN FILTER ESTIMATOR

We begin by describing the parameterization of the unknowns and the structure of the learning mechanism. We restrict ourselves to unknowns which can be represented as constant parameters (time-varying parameters present only computational complications). Let the unknown parameter be represented as a vector $P \in \mathcal{P} \subset \Re^n$. Let sensor i be capable of measuring a relationship $g_i(P, x)$ between the parameter vector and the set of physical Cartesian coordinates $x \in X \subset \Re^3$ given by the sensor's effective location in the workspace (the effective location of a camera is the image coordinates, not the camera coordinates). If we allow some uncertainty and sensor error in the measurement to be modeled as (Gaussian) noise vector ν, the actual sensor outputs are denoted by:

$$Z(P, x) = \begin{bmatrix} g_1(P, x) \\ \vdots \\ g_m(P, x) \end{bmatrix} + \nu \qquad (1)$$

Also let the parameter vector be constrained by the relationship $S(P, x) = 0$.

We can often model measurements with sensor models which are linear in parameter vector P, so that

$$g_i(P, x) = H_i(x)\, P \qquad (2)$$

or with *linearizable* sensor models so that (2) can be used, where in that case,

$$H_i(x) = \left.\frac{\partial g_i}{\partial P}\right|_{P = \hat{P}} \qquad (3)$$

Here $\hat{P}$ is the most recent estimate of the true unknown parameter P. The concatenated matrix of measurements from the m sensors appearing in (1) is then is simply written as $H(x)$.

With the notation ${}^kP$ denoting a value received or computed at time k, we can now write the standard equations for the measurement update of the Kalman filter estimator for vector P (*extended* Kalman filter if (3) is used):

$${}^k\hat{P} = {}^{k-1}\hat{P} + {}^{k-1}Q\,{}^{k-1}H^{\mathrm{T}}\, R^{-1}({}^kZ_{act} - {}^{k-1}Z)$$

$${}^kQ = \left[{}^{k-1}Q^{-1} + {}^{k-1}H^{\mathrm{T}}\, R^{-1}\,{}^{k-1}H\right]^{-1} \qquad (4)$$

where $R = E\{\nu\nu^{\mathrm{T}}\}$ is the measurement noise covariance (assumed constant here), and

$${}^kQ = E\{(P - {}^k\hat{P})(P - {}^k\hat{P})^{\mathrm{T}}\} \qquad (5)$$

is the parameter estimate covariance. Actual received measurements are denoted ${}^kZ_{act}$, and ${}^{k-1}Z$ is a computed approximate measurement given by (2).

Since we have assumed that the unknown parameters are constant, we need not consider the time update equations usually associated with Kalman filters.

III. MINIMIZING A COST CRITERION

The choice of the Kalman filter is motivated by the structure of the covariance update equation (4) and the fortuitous functional dependence on the physical coordinate variable x.

The error covariance is measure of *uncertainty* in the parameter estimate, so it would be reasonable to try to minimize a norm of this matrix as estimation proceeds. Let such a scalar function be denoted $J(Q)$. We will show that the function

$$J(Q) = -\,trace({}^kQ^{-1} - {}^{k-1}Q^{-1}) \qquad (6)$$

is a convenient cost criterion.

First, we note from (4) and (6) that

$$J(Q) = -\,trace({}^{k-1}H^{\mathrm{T}}\, R^{-1}\,{}^{k-1}H) \qquad (7)$$

so that matrix inverses are avoided. Next, we motivate the form of (6) by noting that as we seek out information, we would like to maximize the rate of *descent* of $\|Q\|$, which is equivalent to maximizing the rate of *ascent* of $\|Q^{-1}\|$. We can approximate the *rate* of *ascent* as

$$\Delta Q^{-1} = \left\|{}^kQ^{-1} - {}^{k-1}Q^{-1}\right\|, \qquad (8)$$

and since the covariance matrix is constrained to be positive definite, we can use $\|\cdot\| = trace(\cdot)$. Finally,

since we like to think of "cost" as something to be minimized rather than maximized, we negate the whole expression, resulting in (6).

In order to be an efficient criterion, the cost (6) must be analytically minimized in an efficient way. This is where the functional dependence is critical. We know that from (4), the covariance is computed as a function of the parameter estimate $\widehat{P}$. Considering previous values of Q to be constant, we also see that $\widehat{P}$ is expressed as a function of the system measurement model matrix H. Finally, we know that H is a function of the physical coordinates x. Therefore, by function composition, it is possible to express the cost J as a function of the sensor's physical coordinates x. Therefore, if it is computationally feasible, it is possible to extremize the cost over the set of physical variables. Since we are concerned with active sensing, we pursue the physical variables corresponding to optimal cost.

Finally, we must constrain the search to a particular submanifold within X; in particular, to the tangent space of the constraint surface S. This ensures that the constraint will remain enforced and effectively implies that the optimization actually provides a *direction* $\dot{x}$ in which the sensor should "look" in order to obtain the "most valuable" information about the unknown.

The structure of the exploration and learning mechanism takes the form of the block diagram in Fig. 1. While the unknown and the active sensor are independent of the learning engine, the information they provide and the way they constrain the data are modeled. Together with this model, the estimator provides the uncertainty measure and a prediction of future parameter values, which is passed to the "prospector." The prospector encompasses the cost computations and its minimization routine. The result of prospecting for "valuable territory" is available in the form of direction $\dot{x}$. This direction will always point to the most promising region of the constrained unknown parameter space, but of course, it need not always be followed.

Provided that through function composition the cost J is analytic in the search domain, one might envision other types of explorations as well. The cost given in (6) simply expedites convergence of the parameter estimate P. One might instead prefer to "seek out" a stability extremum, a particular geometric feature on an object (an umbilic, *e.g.*), or a different set of parameters by redefining the parameter vector. That is, if instead of simply describing the identity of the unknown, vector P might be written so that it describes a desired function or other attribute required by a task specification.

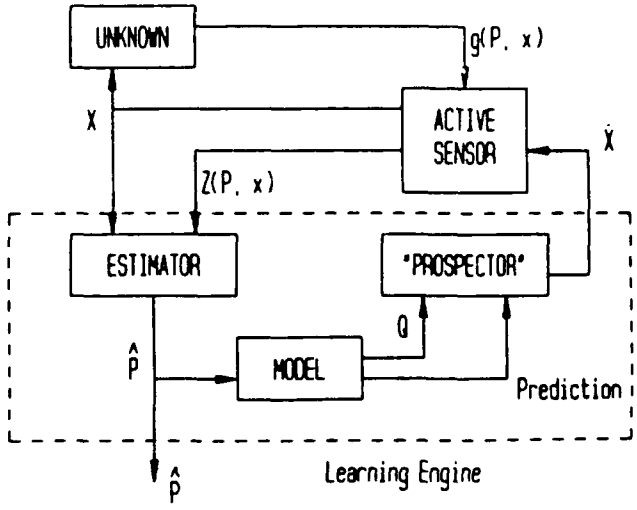


Fig. 1 Block diagram of learning engine.

IV. COMPUTATIONAL ISSUES

The requirement of computational feasibility is further satisfied by the *trace* operator, which facilitates analytic differentiation of J. In order to extremize J over the physical variables x, we set (at each time k):

$$\begin{aligned}\frac{\partial J}{\partial x} &= \frac{\partial}{\partial x} -trace(H^{\mathrm{T}} R^{-1} H) \\ &= -\frac{\partial}{\partial H}(trace(H^{\mathrm{T}} R^{-1} H))\frac{\partial H^{\mathrm{T}}}{\partial x} \\ &= -2 H^{\mathrm{T}} R^{-1}\frac{\partial H^{\mathrm{T}}}{\partial x} \\ &= 0. \end{aligned} \tag{9}$$

This form is sufficient for systems with a single sensor, wherein H is simply a row matrix. If we extend the study to exploit the natural sensor fusion properties of the Kalman filter, we know from (2) that the matrix H can consist of several rows, grouped into sets defined by g_i as in (1). It can be easily shown [2] that if each row of H^{T} is denoted h_j, $j = 1,\ldots,n$, then (9) can be more conveniently written

$$0 = 2\sum_{j=1}^{n} h_j R^{-1}\frac{\partial h_j^{\mathrm{T}}}{\partial x}. \tag{10}$$

The final step is the solution of (9) (or (10)) over all x such that an optimal search direction can be chosen. This procedure is dependent on the actual structure of the sensor models, so we will proceed with a concrete example: shape estimation from a tactile probe.

V. EXAMPLE AND SIMULATIONS

In order to demonstrate the effectiveness of the strategy, the example problem is taken to be the estimation of the shape of a geometric object from information gathered by a fingertip probe. The constraint for this case becomes the surface model, which we choose as quadrics and superquadrics in the following examples, respectively. Both shape models can be written in the (parametric) surface form:

$$S(x,P)=\begin{bmatrix} \frac{1}{\sqrt{a}}\cos^{\varepsilon_1}u_1 \;\; \cos^{\varepsilon_2}u_2 \\ \frac{1}{\sqrt{b}}\cos^{\varepsilon_1}u_1 \;\; \sin^{\varepsilon_2}u_2 \\ \frac{1}{\sqrt{c}}\sin^{\varepsilon_1}u_1 \end{bmatrix} \tag{11}$$

where $\varepsilon_1=\varepsilon_2=1$ for standard quadrics [6]. Transformations between this parametric form and the implicit form of (11) are straightforward [7]. Also, one can easily add six more parameters to describe 3-D translation and rotation from a fixed reference.

The measurement equations for (1) are taken to be a fused combination of the probe's contact location and the object's surface normal vector N. For the standard quadric, these equations can be formulated linearly in P, whereas for the superquadrics, they must be linearized. In the case of the quadric, the explicit equations can be found in [3] or [4].

The search procedure in this example seeks all directions on the tangent of the object surface such that if a step is taken in that direction, the estimate of the parameter vector improves more than if a step is taken in any other direction. For three dimensional object estimation, the search in the tangent plane reduces the minimization problem to a one-dimensional problem. In particular, we search around a unit disk on the tangent plane, centered at the current sensor position. We therefore can search over the angular parameter ρ depicted in Fig. 2. Around this circle, we can compute the change in cost predicted by a step in that direction. By finding the direction in which the *derivative* of this change crosses zero, we can determine the direction of maximal change. We then move in the direction of the ρ indicated by this zero-crossing.

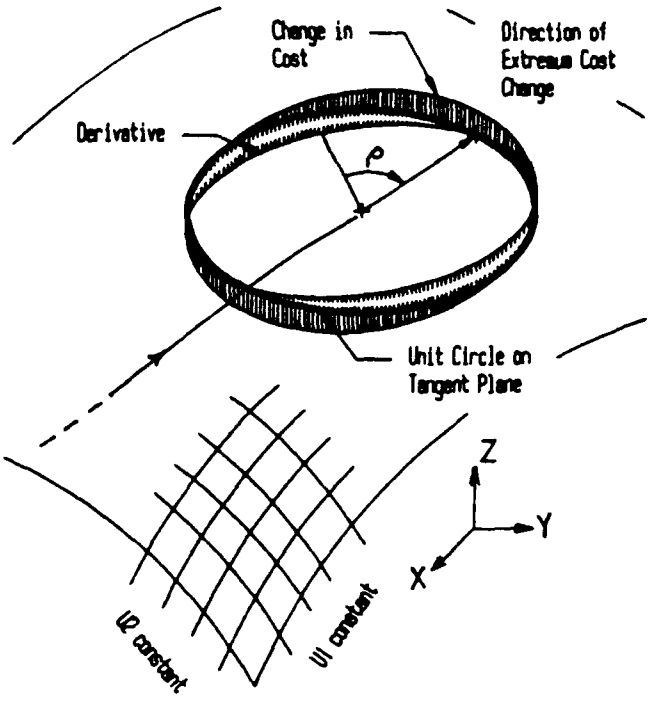


Fig. 2 Portion of constraint surface showing exploration path and unit circle centered at current operating point. We need only search the circle for the angle ρ which directs us to the direction of extremum cost change. This cost change and its derivative are represented as graphs evaluated on the circle.

For the quadric shapes chosen, the cost change will be a second-order polynomial, so that the minimization proceeds quickly. In fact, the computations required in (10) are reasonably fast and real-time experimentation is in progress, using a Merlin 6500 industrial robot with computations performed on an external host running the C language.

Shown in Fig. 3 is the standard quadric with $a=1$, $b=4$, and $c=16$. Starting from a randomly chosen initial condition on the surface and with arbitrary choice of initial covariances, seven paths are shown. Path number 4 is the path automatically chosen by searching the tangent plane of the surface for the best direction in physical coordinates. The other paths are perturbations to the optimal for the purpose of comparison.

The corresponding time histories of cost for the seven paths are shown at the bottom of the figure. Note that indeed while following path 4, the cost function decreases at a maximal rate, while costs for paths 1 and 7 show poor convergence. The dots shown on the path indicate the steps taken (ten for each path). After only these ten steps, the search algorithm can converge from an initial guess of $a_0=0$, $b_0=0$, $c_0=0$ to $a_{10}=0.80$, $b_{10}=3.78$, $c_{10}=15.23$. This

uniformly converges to the true parameters (1, 4, 16) until the cost is practically flat. At that point, the exploration path terminates, since the minimization algorithm can no longer find an extremum. The simulated probe stops at that point on the surface (see [2]).

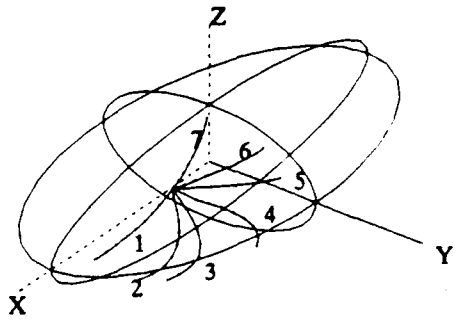
Figure 4 gives a similar plot for the exploration of the superquadric as presented in (11), in this case with parameters $a = b = c = 1$, $\varepsilon_1 = \varepsilon_2 = 1.5$. This plot demonstrates again that the "optimal" path, number 4, converges to a lower cost value faster than the others. Here, however, we see a significant portion of the cost function plot in which a *different* path appears to be out-performing the optimal. This behavior has not been fully explained but is partially due to the rather large steps ($2\pi/100$ rads) taken around the unit circle when searching for extrema of the change in cost. A step in a slightly inaccurate direction in one iteration could conceivably reduce optimality for several future steps.

Note also that path 4 seems to track a contour before entering a more flat region. This behavior is also not fully explained, but may indicate the high saliency of contours as discussed below. As contour data becomes sufficient to provide more complete information about a particular subspace of $\mathcal{P}$, the path then must diverge from the contour for information on the rest of $\mathcal{P}$. That is, the contour provides rich information when it is first reached, but after a while, information from other regions becomes relatively more valuable: the contour has provided all the information it could provide.

VI. DISCUSSION

An interesting issue raised by the plot in Figs. 3 and 4 is the behavior of the path traced on the surface, and its implication for sensor information. It is clear that the exploration path must not be parallel to any parametric (u_1-constant and u_2-constant) line on the surface. Otherwise, the information gained by the sensor would provide clues to only a subset of the parameter space.

For example, suppose we forced the sensor to travel in a path formed by the intersection of the $x-y$ plane and the quadric object. Then the parameters (a and b only) which describe that ellipse of intersection could be determined, but nothing could possibly be learned about parameter c. Instead, as we see in the figures, all paths must contain some information about the entire parameter space. (This principle can be related to the sufficient excitation problem in control theory.)



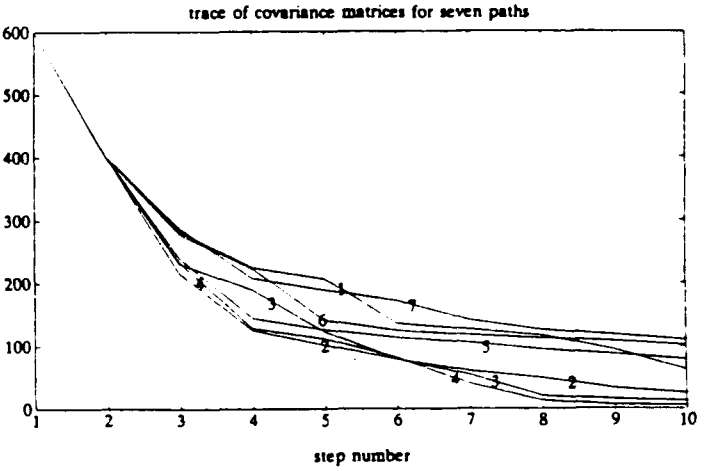


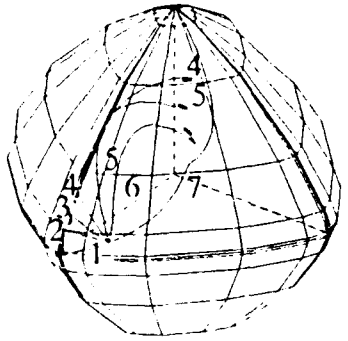
Fig. 3 Optimal path on ellipsoid (number 4), along with associated error criteria for each.

This can be a significant drawback in the case of many objects. For example, the system may be presented with an object with a relatively flat patch (such as when a superquadric has an ε-parameter very close to 0 or 2), with the initial condition in this patch. The learning system can easily interpret this patch as a plane, and estimate its orientation, but this is only a small part of the object. The covariance matrix will thereafter not change much, and the path "wanders" around, not finding an optimal direction. The flat patch on one object looks locally very much like the flat patch on any other object, so convergence will be poor unless some additional exploration criterion is provided to straighten-out the exploration path.

Now consider the possibility that some of the sensor data is weighted more heavily than other sensor data, either through the sensor noise covariance matrix R or some arbitrary weighting scale introduced into the cost criterion. One would expect that the observed behavior manifested in the exploration path would vary continuously with the weighting values. Some sensors would be more "interested" in some geometric features than in others. For example, we have assumed a sensor which provides the normal direction to the surface. If this sensor is increasingly weighted in the exploration

engine, we would expect the path to tend toward regions of higher surface *curvature*. This is because the normal direction per arclength traversed changes more quickly at such points than at any other point.

Even more interesting for this algorithm is the fact that the path traversed is generated autonomously. By observing the behavior of the system, we can see what features the learning system is "interested in;" the path leads us to them. For learning systems intended for industrial manufacturing operations, such as adaptive arc-welding robots, these "salient" features become important in part design. The robot itself, with this learning algorithm, tells us what features it needs in order to do its job, and we can build these features into the workpieces.



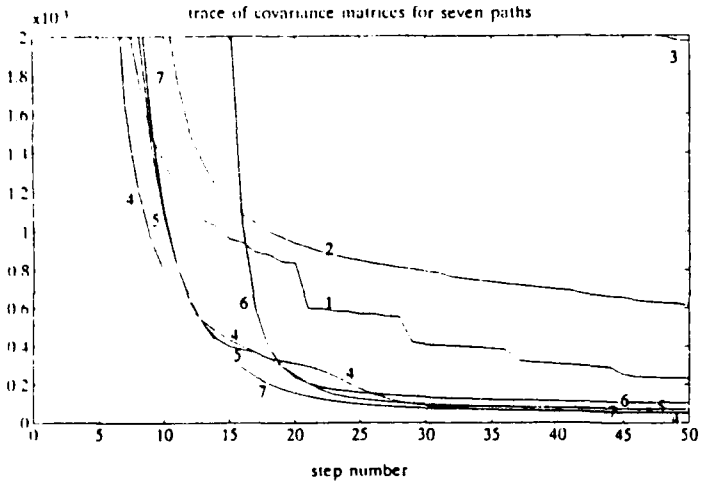


Fig. 4 Optimal path on superquadric (number 4), along with associated error criteria for each. In this simulation, translation and rotation parameters were not estimated.

ACKNOWLEDGEMENT

The author would like to thank Mr. Thomas Stockli for his work in simulating the exploration of the superquadric shape.

REFERENCES

[1] Barnard, S. T. and A. P. Pentland, "Three-dimensional shape from line drawings," *Proc. 8th International Joint Conference on Artificial Intelligence*, pp. 1062-1064, Karlsruhe, FRG, 1983.

[2] Bay, J. S., "A fully autonomous active sensor-based exploration concept for shape sensing robots," *IEEE Transactions on Systems, Man, and Cybernetics*, in press.

[3] Bay, J. S., "Tactile shape sensing via single- and multi-fingered hands," *Proc. IEEE International Conference on Robotics and Automation*, pp. 290-295, Scottsdale, AZ, 1990.

[4] Bay, J. S. and H. Hemami, "Dynamics of a learning controller for surface tracking robots on unknown surfaces," *IEEE International Conference on Robotics and Automation*, Cincinatti, pp. 1910-1915, OH, 1990.

[5] Klatzky, R. L., S. Lederman, and C. Reed, "Haptic integration of object properties: texture, hardness, and planar contour," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 1, pp. 45-57, 1989.

[6] Solina, F., "Shape recovery and segmentation with deformable part models," Ph.D. Thesis, University of Pennsylvania, December 1987.

[7] Struik, D. J., Lectures on Classical Differential Geometry, New York: Dover Publications, Inc., 1961.