Automatica, Vol. 28, No. 2, pp. 389–394, 1992
Printed in Great Britain.

0005-1098/92 \$5.00 + 0.00
Pergamon Press plc

Brief Paper

Distance Estimation and Collision Prediction for On-line Robotic Motion Planning*

K. J. KYRIAKOPOULOS† and G. N. SARIDIS‡

Key Words—Collision avoidance; collision prediction; distance functions; random search.

Abstract—An efficient method for computing the minimum distance and predicting collisions between moving objects is presented. This problem has been incorporated in the framework of an in-line motion planning algorithm to satisfy collision avoidance between a robot and moving objects modelled as convex polyhedra. In the beginning, the deterministic problem where the information about the objects is assumed to be certain is examined. If instead of the Euclidean norm, L_1 or L_∞ norms are used to represent distance, the problem becomes a linear programming problem. The stochastic problem is formulated, where the uncertainty is induced by sensing and the unknown dynamics of the moving obstacles. Two problems are considered: First, filtering of the distance between the robot and the moving object, at the present time. Second, prediction of the minimum distance in the future, in order to predict the collision time.

1. *Introduction*

THE TRAJECTORY planning problem for mobile robots or robotic arms has been traditionally treated for environments of stationary or deterministically moving objects (Brooks, 1986; Chen, 1988; Kant and Zucker, 1988; Khatib, 1985; Lozano-Perez, 1987; Lumelsky, 1987). This is done because in order to satisfy task constraints and optimize over certain criteria, all the available *a priori* information has to be utilized, and therefore most of the planning has to have a global character. However, robots should be able to perform their tasks efficiently, not obstructed by and not obstructing other moving robots, vehicles or mechanisms with which they have loose communication. Therefore, local decision making strategies should be included in the robotic motion planning algorithms.

The difference between the global and local motion planning strategies is mainly found at the amount of processed information about the task and the environment and the considered time length of the plan. Both of these measures are significantly larger for the global case which is bound to be off-line under the state of the art of hardware.

An on-line strategy should then decrease the amount of information processing. Therefore it should be based firstly on the already processed information which has obtained the form of an off-line plan, and secondly, on some additional local measurements and estimates of measures related to the obstructing moving objects. Such measures have been proposed (Kyriakopoulos and Saridis, 1990, 1991; Kyriakopoulos, 1991) to be the minimum distance between the robot and the moving obstacles, and the expected collision time under the current plan.

The calculation of both the minimum distance and the expected collision time has been investigated in the past (Canny, 1986; Gilbert *et al.*, 1988; Kyriakopoulos and Saridis, 1989a) but in the case where full information about the description of the objects and their motion is available. Unfortunately, this is not the case in realistic environments where the information is based on measurements by a sensing system and the motion of the objects is not known but just observed. The uncertainty introduced by both factors should be included in the calculation procedures.

In this paper, the major contribution is the treatment of uncertainty introduced by both the sensing process and the unknown attitude of the moving obstacles. The distance estimation problem is posed as a filtering problem. Finally, the issue of collision time prediction is covered in depth. Such a process is computationally expensive. If the L_1 or L_∞ are used, then the distance problem is linear programming as opposed to quadratic for the case of the Euclidean norm. Linear programming for problems of moderate complexity, such as the one at hand, performs considerably better than quadratic.

A statement of the problem in the deterministic case is given in Section 2 in order to introduce the framework under which analysis is going to be performed. Section 3 contains the problem formulation of the stochastic case and a proposed solution for the minimum distance estimation. The collision time prediction is given in Section 4, while some example case with simulation results is presented in Section 5. Finally, Section 6 includes comments on the obtained results, and some discussion on issues that could consitute topics of future research.

2. *Distance functions—the deterministic case*

2.1. *Definitions.* The distance between the objects at the time instant t is defined (Gilbert, 1985) as

$$d(t) = \min_{i,j} \{\|z_i - z_j\| / z_i \in K_i(t),\ z_j \in K_j(t)\} \tag{2.1}$$

where $K_i(t)$, $K_j(t)$ are compact sets of Cartesian points representing the two objects.

Based on the above definition the necessary and sufficient condition so that no collision occurs between the two objects would be $d^o - d(t) \le 0\ \forall t$ where $d^o > 0$ is a safety constant.

Sets $K_i(t)$, $K_j(t)$ are described by an equation of the form $K(t) = R(t) \cdot C + \{T(t)\}$ where, $C \subset \mathscr{R}^3$ is a compact set and describes the shape of the object, $R(t) \in \mathscr{R}^{3\times 3}$ is a rotation matrix, $T(t) \in \mathscr{R}^3$ is a translation vector.

An assumption adopted in this work (Kyriakopoulos and Saridis, 1989) is that set C representing the shape of the

* Received 30 July 1989; revised 8 February 1991; revised 6 August 1991; received in final form 6 September 1991. The original version of this paper was presented at the 11th IFAC World Congress on Automatic Control at the Service of Mankind which was held in Tallinn, Estonia during August 1990. The published proceedings of this IFAC Meeting may be ordered from Pergamon Press plc, Headington Hill Hall, Oxford OX3 0BW, U.K. This paper was recommended for publication in revised form by Editor A. P. Sage.

† NASA Center of Intelligent Robotic Systems for Space Exploration (CIRSSE), Electrical, Computer and Systems Engineering, Dept. (ECSE), Rensselaer Polytechnic Institute (RPI); presently at the New York State Center for Advanced Technology in Robotics and Automation, RPI, Troy, NY 12180, U.S.A.

‡ NASA-CIRSSE, ECSE, RPI, Troy, NY 12180, U.S.A. Author to whom all correspondence should be addressed.

object is a convex polyhedron described by $C = \{x/x \in \mathcal{R}^3 \ni A \cdot x \leq b,\ A \in \mathcal{R}^{m\times 3},\ b \in \mathcal{R}^m\}$.

A point $x \in \mathcal{R}^3$ after it is rotated and translated by $R(t)$ and $T(t)$, respectively, it comes to a new point x' where $x' = R(t) \cdot x + T(t)$. Therefore $K(t)$ since it comes from rotation and translation of points $x \in C$ can be described as $K(t) = \{x'/x' \in \mathcal{R}^3 \ni A \cdot R^{-1}(t) \cdot x' \leq b + A \cdot R^{-1}(t) \cdot T(t)\}$.

2.2. *Representations of distance.* Based on the definitions of the previous section, the problem to find the minimum distance between two solids $C_r = \{x/x \in \mathcal{R}^3 \ni A_r \cdot x \leq b_r,\ A_r \in \mathcal{R}^{m\times 3},\ b_r \in \mathcal{R}^m\}$, $C_o = \{y/y \in \mathcal{R}^3 \ni A_o \cdot y \leq b_o,\ A_o \in \mathcal{R}^{l\times 3},\ b_o \in \mathcal{R}^l\}$ representing the robot and an obstacle, respectively, is recasted to a mathematical programming problem of the form:

$$\begin{aligned} d(t) = \min_{x,y} \quad & \|x - y\| \\ s.t. \quad & A_r(t) \cdot x(t) \leq b_r(t) \\ & A_o(t) \cdot y(t) \leq b_o(t) \end{aligned} \tag{2.2}$$

where $A_r(t) = A_r \cdot R_r^{-1}(t)$, $b_r(t) = b_r + A_r \cdot R_r^{-1}(t) \cdot T_r(t)$, $A_o(t) = A_o \cdot R_o^{-1}(t)$, $b_o(t) = b_o + A_o \cdot R_o^{-1}(t) \cdot T_o(t)$ and $R_r(t)$, $T_r(t)$, $R_o(t)$, $T_o(t)$ are the rotation and translation matrices for the robot and the obstacle describing their current configuration. In view of (2.2), the distance between the solids C_r and C_o can be viewed as a function of their configuration (Gilbert and Johnson, 1985) described as a quadruple $P = (R_o(t), T_o(t), R_r(t), T_r(t))$:

$$d = d(P): \mathcal{P} \equiv \mathcal{R}^{3\times 3} \times \mathcal{R}^3 \times \mathcal{R}^{3\times 3} \times \mathcal{R}^3 \rightarrow \mathcal{R}^+ \tag{2.4}$$

and therefore a domain $\mathcal{P}^+$ of d can be defined as: $\mathcal{P}^+ = \{P \in \mathcal{P}/d(P) > 0\}$.

The norm used in mathematical program (2.2) was not explicitly defined here. It can be any norm because norms in $\mathcal{R}^n$ are equivalent. Although the Euclidean norm is the most natural to our intuition, other norms such as $\|\cdot\|_1$ and $\|\cdot\|_\infty$ can also be used because of their computational efficiency. This comes from the fact that the computation of the Euclidean norm requires the solution of a quadratic programming problem, while this of $\|\cdot\|_1$ and $\|\cdot\|_\infty$ requires solution of linear programming problems. An analysis of the above issues is presented in (Kyriakopoulos and Saridis, 1989b). In the analysis to follow, the minimum distance is going to be associated with no specific norm. In case that the kind of norm is significant then it will be indicated with the form of a subscript, e.g. d_1, d_2, d_∞.

3. *Distance functions—the stochastic case*

3.1. *Sources of uncertainty.* From (2.2) it becomes clear that the computation of the distance between two objects requires accurate knowledge of the parameters describing the shape of the objects C_o and C_r and these describing the configuration of them as represented by the quadruple P. Accurate knowledge of the solid models C_o and C_r, and of the configuration parameters $R_r(t)$, $T_r(t)$ of the robot is assumed here. The uncertain information considered here is the information about the configuration $R_o(t)$, $T_o(t)$ of the moving obstacle C_o. Future work is going to incorporate uncertainties in a more general fashion.

The uncertainty of the kinematic information of the moving obstacle C_o is partially due to the way that it is extracted. For example, in a measurement process based on a vision system the uncertainty is mainly introduced by two kinds of noises: first, the image intensity noise from the environment, and second, the quantization noise of the image array of the camera.

The most eminent source of kinematic uncertainty is the lack of knowledge about the "attitude" of the moving obstacle. In other words the lack of knowledge about the time evolution of its Cartesian position and orientation and their first and second order time derivatives. The assumption made here is continuity of position and orientation and their velocities. The accelerations were assumed to be constant in time during every sampling interval and their magnitude is going to be estimated.

3.2. *Estimation of kinematic parameters.* Naturally, the assumptions made about the models of uncertainty are necessary in order to enable the creation of models of motion of the moving object, and the sensing process. The models presented here are for the two dimensional case. Future work is going to incorporate the full three-dimensional case which just adds complexity to the formulas. On the other hand, the two-dimensional case perfectly serves the navigation problem considered here.

$$\begin{aligned} \dot{p}_x &= v_x & \dot{p}_y &= v_y & \dot{p}_\theta &= v_\theta \\ \dot{v}_x &= \alpha_x & \dot{v}_y &= \alpha_y & \dot{v}_\theta &= \alpha_\theta \\ \dot{\alpha}_x &= 0 & \dot{\alpha}_y &= 0 & \dot{\alpha}_\theta &= 0 \end{aligned} \tag{3.1}$$

where (p_x, p_y, p_θ) define the Cartesian position and orientation in two dimensions, (v_x, v_y, v_θ) are the translational and rotational velocities and $(\alpha_x, \alpha_y, \alpha_\theta)$ are the translational and rotational accelerations. The above equations constitute a set of state equations, and can be written in a matrix form: $\dot{X} = A \cdot X$ where

$$X = [p_x \quad v_x \quad \alpha_y \quad p_y \quad v_y \quad \alpha_y \quad p_\theta \quad v_\theta \quad \alpha_\theta],\ A \in \mathcal{R}^{9\times 9}.$$

The measurement model is described by an equation of the form:

$$z = h(X) + v \tag{3.2}$$

where $z \in \mathcal{R}^m$ is a vector of parameters of several geometric features of the object. These parameters are extracted by the sensing process (e.g. vision). The dependency of z on X is given by the, nonlinear in general, function $h(X)$. The corrupting noise v is assumed to be white and Gaussian, and independent of X. The assumption about the Gaussian property of the noise although not accurate, is attributed to the Central Limit Theorem, because the noise under consideration is as an aggregation of many independent sources of noise (Durrant, 1988). The mean and the covariance matrix of noise are, respectively, $E\{v\} = 0$, $C(v_i, v_j) = E(v_i \cdot v_j) = R_i \cdot \delta_{ij}$.

A variation of the Kalman filter (Athans *et al.*, 1968) may well be used to get a close to optimal estimate X of state (motion) vector X. The selection of this specific filter was done because it is a second order filter which normally gives better results in terms of bias errors. The utilization of Kalman filter provides, at every instant t_i:

- $\hat{X}(t_i/t_i)$, a suboptimal estimate of X based on measurements up to t_i, and
- $\hat{P}(t_i/t_i)$ the error covariance matrix.

The prediction equations for the future, based on information up to t_i, are

$$\begin{aligned} \hat{X}(t/t_i) &= e^{A\cdot(t-t_i)} \cdot \hat{X}(t_i/t_i) \quad t \geq t_i \\ \dot{\hat{P}}(t/t_i) &= A \cdot \hat{P}(t_i/t_i) + \hat{P}(t_i/t_i) \cdot A^T. \end{aligned} \tag{3.3}$$

The above Ricatti equation does not induce any kind of computational difficulty because it is homogeneous and can be solved analytically.

3.3. *Bounds of distance function.* From (2.4), it becomes evident the fact that the dependency of d on quadruple $P = (R_o(t), T_o(t), R_r(t), T_r(t))$ implies its dependence on the motion vector X. This comes from the fact that $R_o(t)$ and $T_o(t)$ explicitly depend on X. In fact, in the two-dimensional case:

$$R_o(t) = R_0(\theta(t)) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$T_o(t) = T_o(p_x(t), p_y(t)) = \begin{bmatrix} p_x \\ p_y \end{bmatrix}.$$

Assuming that $R_r(t)$ and $T_r(t)$ are well-known at t, the distance d becomes a function of the random variable X, or $d = d(X)$; and as a matter of fact d is a random variable, too. From the discussion at the end of Section 2.2 it becomes obvious that $d(X)$ is not given in a closed form, but it is an outcome of a mathematical programming stage. Additional investigation (Clarke, 1983; Gilbert and Johnson, 1985) has shown that d although uniformly Lipschitz continuous, is not differentiable everywhere on its domain $\mathcal{P}^+$. However, $\mathcal{P}^+$ can be viewed as a subset of $\mathcal{R}^n$ and by Rademachers theorem (Clarke, 1983), function $d(X)$ is differentiable almost everywhere (a.e), i.e. except on a subset of points with Lebesgue measure zero. From the geometry of the problem, we can easily see that the set of points of nondifferentiability is not dense on $\mathfrak{R}^n$ and therefore there exist open neighbourhoods where $d(X)$ is differentiable everywhere and analytic. In the discussion to follow, this is going to be mathematically more evident.

As a matter of fact, function $d(X)$ can be approximated by a Taylor expansion up to the second order (due to the Gaussian noise) around the expected value $\hat{X}$ of X:

$$d(X) \approx d(\hat{X}) + (X - \hat{X})^T \cdot \frac{\partial d(\hat{X})}{\partial X} + \tfrac{1}{2}(X - \hat{X})^T \cdot \frac{\partial^2 d(\hat{X})}{\partial X^2} \cdot (X - \hat{X}).$$

Taking expected values in both sides,

$$E(d(X)) \geq d(\hat{X}) + \tfrac{1}{2}\lambda_{\min}\left(\frac{\partial^2 d(\hat{X})}{\partial X^2}\right) \cdot tr(P(t)) \quad (3.4)$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a matrix.

In order to make possible the utilization of (3.4), the Hessian of $d(X)$ has to be explicitly found, in order to get the eigenvalues in closed form. This would be more efficient for an on-line distance estimation scheme.

3.4. *Derivatives of distance functions.* The analysis that follows considers the two-dimensional case and is consistent with the development of Gilbert and Johnson (1985). The three-dimensional extension is straightforward.

Define $\eta_a : \mathcal{P}^+ \times C_o \times C_r \to R^+$ by: $\eta_a(P, w_o, w_r) = \|R_o \cdot w_o + p_o - R_r \cdot w_r - p_r\|_a$ where "a" defines the kind of used norm. Let

$$L = \begin{bmatrix} L_x \\ L_y \end{bmatrix} = R_o \cdot w_o + p_o - R_r \cdot w_r - p_r.$$

Case a = 2. In this case $\eta_2(P, w_o, w_r) = (L_x^2 + L_y^2)^{1/2}$ and

$$\nabla_X \eta_2(P, w_o, w_r) = \left(\frac{L_x}{\eta_2}, 0, 0, \frac{L_y}{\eta_2}, 0, 0, \frac{-M_x \cdot L_x + M_y \cdot L_y}{\eta_2}, 0, 0\right)^T \quad (3.5)$$

where $M_x = w_o^x \cdot \sin\theta + w_o^y \cdot \cos\theta$, $M_y = w_o^x \cdot \cos\theta - w_o^y \cdot \sin\theta$ and $w_o = [w_o^x\ w_o^y]^T$. The Hessian can be found by further differentiation and is given in Kyriakopoulos and Saridis (1989b) with its eigenvalues.

Based on Clarke (1975), and Gilbert and Johnson (1985) it can be proved that in the case that P is nonsingular, i.e. when $W(P) = \{(w_o, w_r) \in C_o \times C_r / \eta_2(P, w_o, w_r) = d(P)\}$ is singleton, then $\nabla_X d(X) = \nabla_X \eta_2(P, w_o^*, w_r^*)$ where $(w_o^*, w_r^*) \in W(P)$. A typical case of singular configuration is demonstrated in Fig. 1.

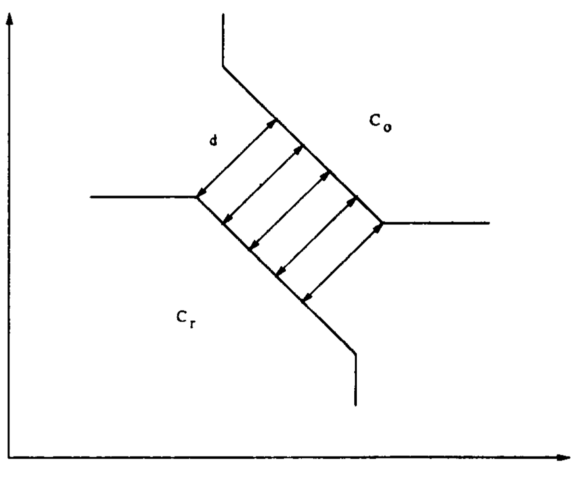

FIG. 1. Singular configuration.

Case a = 1. In this case $\eta_1(P, w_o, w_r) = |L_x| + |L_y|$ and

$$\nabla_X \eta_1(P, w_o, w_r) = \left(\operatorname{sgn}(L_x), 0, 0, \operatorname{sgn}(L_y), 0, 0, \frac{\partial \eta_1}{\partial \theta}, 0, 0\right)^T \quad (3.6)$$

$$\frac{\partial \eta_\infty}{\partial \theta} = \begin{cases} \operatorname{sgn}(L_x) \cdot \{-w_o^x \cos\theta - w_o^y \sin\theta\} \\ \quad \text{if } \eta_\infty(P, w_o, w_r) = |L_x| \\ \operatorname{sgn}(L_y) \cdot \{w_o^x \cos\theta - w_o^y \sin\theta\} \\ \quad \text{if } \eta_\infty(P, w_o, w_r) = |L_x|. \end{cases}$$

Similarly, the Hessian is very easy to obtain and is given in Kyriakopoulos and Saridis (1989b) with its eigenvalues.

In Kyriakopoulos and Saridis (1989b), the following theorem is proven.

Theorem 1. If

$$d_1(P) = \min_{i,j} \{\|z_i - z_j\|_1 / z_i \in K_i(P),\ z_j \in K_j(P)\},$$

$\eta_1(P, w_o, w_r) = \|R_o \cdot w_o + p_o - R_r \cdot w_r - p_r\|_1$ and $W(P) \triangleq \{(w_o, w_r) \in C_o \times C_r / \eta_1(P, w_o, w_r) = d_1(P)\}$ is singleton, then $\nabla_X d(X) = \nabla_X \eta_1(P, w_o^*, w_r^*)$ where $(w_o^*, w_r^*) \in W(P)$.

Case a = ∞. In this case $\eta_\infty(P, w_o, w_r) = \max\{|L_x|, |L_y|\}$ and

$$\nabla_X \eta_\infty(P, w_o, w_r) = \begin{cases} \left[\operatorname{sgn}(L_x), 0, 0, 0, 0, 0, \dfrac{\partial \eta_1}{\partial \theta}, 0, 0\right]^T \\ \quad \text{if } \eta_\infty(P, w_o, w_r) = |L_x| \\ \left[0, 0, 0, \operatorname{sgn}(L_y), 0, 0, \dfrac{\partial \eta_1}{\partial \theta}, 0, 0\right]^T \\ \quad \text{if } \eta_\infty(P, w_o, w_r) = |L_y| \end{cases} \quad (3.7)$$

where

$$\frac{\partial \eta_\infty}{\partial \theta} = \begin{cases} \operatorname{sgn}(L_x) \cdot \{-w_o^x \cos\theta - w_o^y \sin\theta\} \\ \quad \text{if } \eta_\infty(P, w_o, w_r) = |L_x| \\ \operatorname{sgn}(L_y) \cdot \{w_o^x \cos\theta - w_o^y \sin\theta\} \\ \quad \text{if } \eta_\infty(P, w_o, w_r) = |L_x|. \end{cases}$$

The Hessian is given in Kyriakopoulos and Saridis (1989b) with its eigenvalues.

In Kyriakopoulos and Saridis (1989b) the following corollary is proven.

Corollary 1. If

$$d_\infty(P) = \min_{i,j} \{\|z_i - z_j\|_\infty / z_i \in K_i(P),\ z_j \in K_j(P)\},$$

$$L = \begin{bmatrix} L_x \\ L_y \end{bmatrix} = R_o \cdot w_o + p_o - R_r \cdot w_r - p_r,$$

$$\eta_\infty(P, w_o, w_r) = \|L\|_\infty = \max\{|L_x|, |L_y|\}$$

and the sets $\mathcal{W}(L) = \arg\min\{|L_x|, |L_y|\}$, and $W(P) \triangleq \{(w_o, w_r) \in C_o \times C_r / \eta_\infty(P, w_o, w_r) = d_\infty(P)\}$ are singleton, with $\mathcal{W}(L) = \{l^*\}$ and $W(P) = \{(w_o^*, w_r^*)\}$ then $\nabla_X d(X) = \nabla_X l^*(P, w_o^*, w_r^*)$.

3.5. *Distance prediction algorithm.* In summary, the steps of the distance prediction algorithm are given.

- Kalman filter provides $\hat{X}(t/t_i)$, $\hat{P}(t/t_i)$ $t \geq t_i$ (3.3)
- A mathematical programming stage (2.2) gives $d_2(\hat{X})$ (or $d_1(\hat{X})$ or $d_\infty(\hat{X})$)
- $\lambda_{\min}\left(\dfrac{\partial^2 d(\hat{X})}{\partial \hat{X}^2}\right)$ is obtained (Kyriakopoulos and Saridis, 1989b)
- From (3.4) a lower bound for $E(d(X))$ is obtained.

4. *Prediction of collision time*

The prediction of collision time is going to be based on two kinds of information: firstly, the future motion of the robot which is preplanned and therefore accurately known, and secondly, the motion of the moving obstacle, as can be "optimally" predicted by (3.3). It is worthwhile noticing the fact that distance, as a function of time, is not necessarily differentiable. Additionally, the future time-evolution of $d(t)$

is not convex. Finally, we are not interested just in a local minimum of $d(t)$, but in the first time instant t_c that $d(t)$ becomes zero, or mathematically,

$$t_c = \inf_{t \in [t_i, t_i + T]} \{d(t) = 0\} \tag{4.1}$$

where t_i is the present time, and T is the time horizon of interest.

The characteristics of the above problem do not permit the use of stochastic approximation methods. Although some methods that have recently appeared give accurately the collision time based on the considered information, they seem to have computational problems that do not enable their use in on-line schemes. For those reasons, another method, the accelerated expanding subinterval random search (AESRS) (Saridis, 1977) was utilized with modifications to accommodate the problem. The convergence properties of this method have been well investigated, and its performance experimentally verified. The basic idea of AESRS is the following.

Obtain the lower bound $d(t_i)$ from (3.4) and set $d_{min} = d(t_i)$. Split the interval $\Omega = [t_i, t_i + T]$ of concern into N subintervals $\Omega_i \subset \Omega$ such that $\Omega_i \cap \Omega_j = \emptyset$, $i \neq j$. Define a probability density function (pdf) $p(\rho)$ over Ω by: $p(\rho) = \sum_{k=1}^{N} = \beta(k)p_k(\rho_k)$, $\rho_k \in \Omega_k$, $\sum_{k=1}^{N} \beta_k = 1$ where each $p_k(\rho_k)$ is defined only on Ω_k. The pdfs $p_k(\rho_k)$ can be chosen to be any distribution, but in this work were chosen to have uniform distribution for practical reasons. Initially $\beta(k) = \beta_0(k) = 1/N$.

Based on $p(\rho)$ a random time instant $t_r \in \Omega$ is obtained, and from (3.4) a lower bound $d(t_r)$ is obtained for the distance. If $d(t_r) < d_{min}$ then it is assumed to be a "successful" iteration. The weights $\beta_i(k)$ are updated by a stochastic approximation reinforcement algorithm, based on the frequency of "successful" iterations in the particular kth bin,

$$d(t_r)\begin{cases} \leq d_{min} \Rightarrow \begin{cases} \beta_i(k) = \beta_{i-1}(k) + \gamma_i[1 - \beta_{i-1}(k)] \\ \beta_i(j) = \beta_{i-1}(j) - \gamma_i\beta_{i-1}(j) \qquad , \quad j \neq k \end{cases} \\ > d_{min} \Rightarrow \beta_i(k) = \beta_{i-1}(k) \quad k = 1, \ldots, N. \end{cases} \tag{4.2}$$

In the above equations γ_i and $\beta_0(k)$ must satisfy Dvoretzky's conditions for convergence

$$\lim_{i\to\infty} \gamma_i = 0, \ \lim_{i\to\infty} \sum_{j=1}^{i} \gamma_j = \infty, \ \lim_{i\to\infty} \sum_{j=1}^{i} \gamma_j^2 < \infty, \ \|\beta_0(k)\| < \infty$$

therefore $\gamma_i = 1/i$ and $\beta_0(k) = 1/N$.

The definition of t_c in (4.1) suggests that the following amendments to the above algorithm should be done to achieve better performance.

(A) Consider Fig. 2. The present time is $t_i = 2$ sec and the time horizon of interest is $T_h = 6$ sec. A histogram of $N = 12$ bins represents $p_k(\rho_k)$, while the dashed line is $d(t/t_i = 2)t \in [2, t_i + T_h]$. The lower bound of the predicted distance

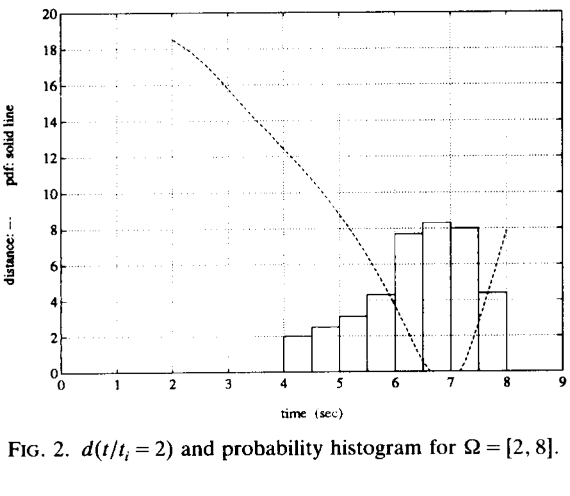

FIG. 2. $d(t/t_i = 2)$ and probability histogram for $\Omega = [2, 8]$.

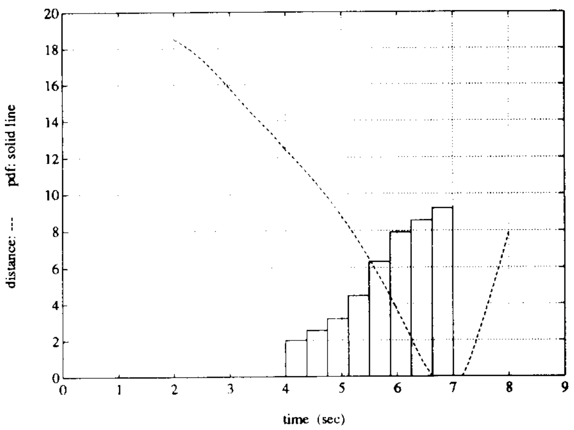

FIG. 3. $d(t/t_i = 2)$ and probability histogram for $\Omega = [2, 7]$.

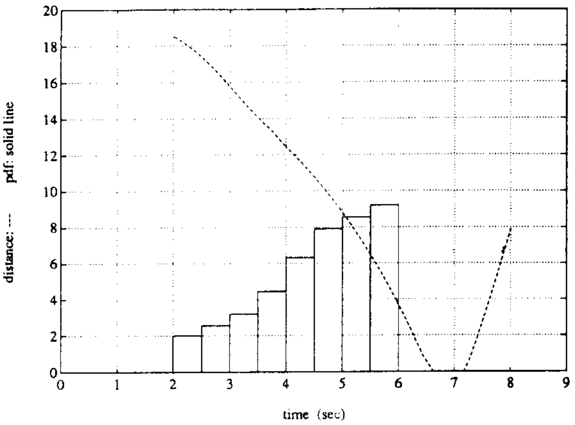

FIG. 4. $d(t/t_i = 2)$ and probability histogram for $\Omega = [2, 6]$ at time $t = 3^-$.

$d(t/t_i)$ is computed using (3.4) not everywhere but only at random instants t_r produced by $p_k(\rho_k)$.

Assume that a trial time instant $t_r = 7$ sec is randomly produced by the probability of (4.2). Then $d(t_r/t_i) = 0$. Then a candidate collision time $t_c = t_r = 7$ has been found and the search interval Ω is modified to $\Omega = [t_i = 2, t_c = 7]$. Then the new search interval is being divided in N subintervals and therefore a finer grid is obtained (Fig. 3). Since the grid becomes finer and finer the uniform distribution for $p_k(\rho_k)$ is satisfactory. The shape of the distribution of $\beta(k)$ $k = 1, \ldots, N$ in the new interval, is similar to the previous distribution, thus providing the search with "memory".

(B) Consider Fig. 4. The present time is $t = 3^-$, all the

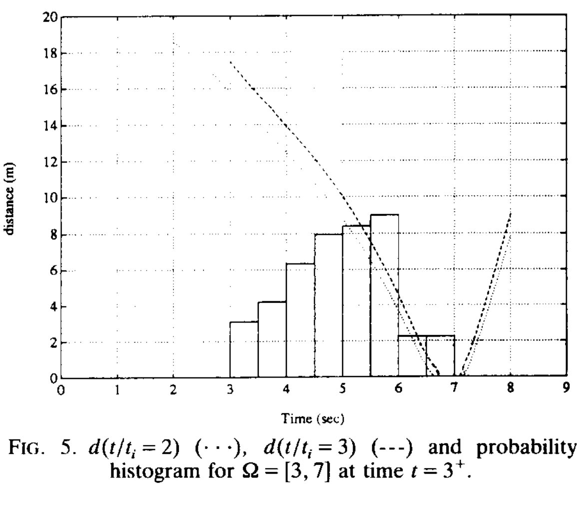

FIG. 5. $d(t/t_i = 2)$ $(\cdots)$, $d(t/t_i = 3)$ (---) and probability histogram for $\Omega = [3, 7]$ at time $t = 3^+$.

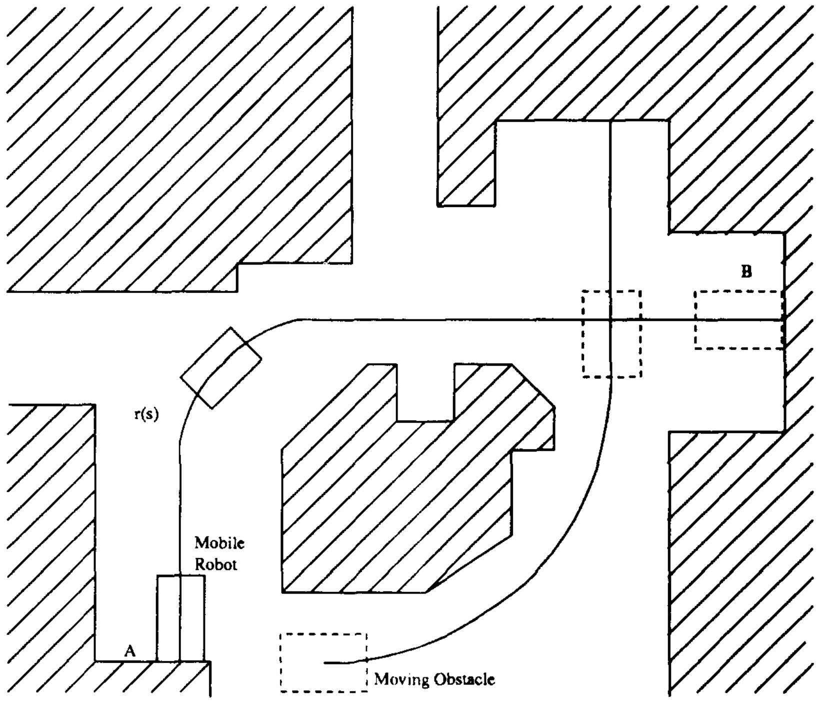

FIG. 6. Environment with a mobile robot and a moving obstacle.

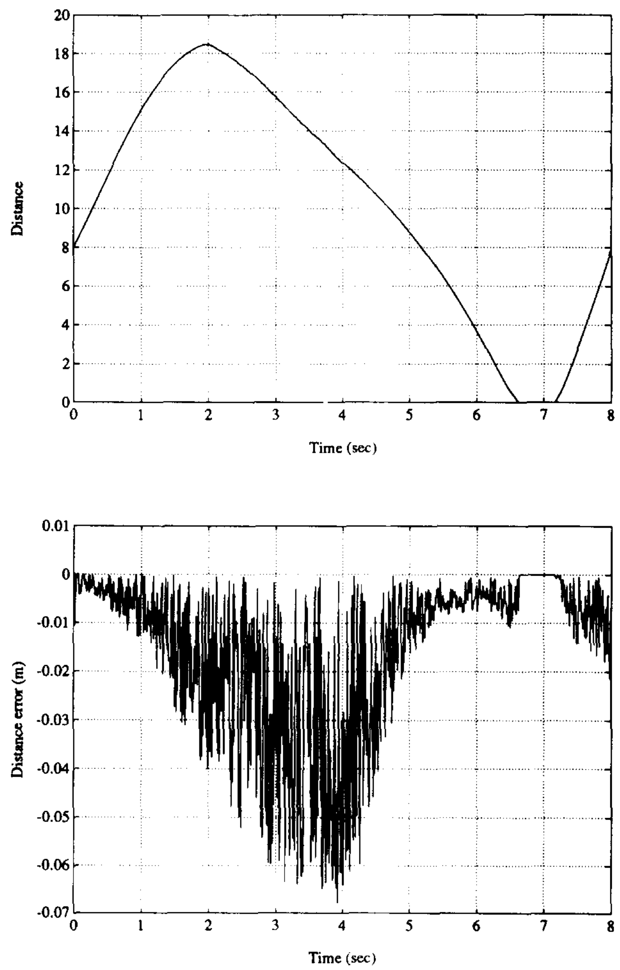

FIG. 7. (a) Lower bound of $E(d_2(X(t)))$, (b) error $E(d_2(X(t))) - d_2(X(t))$.

information processing has been done based on sensing information at $t_i = 2$ sec and the time horizon of interest is $T_h = 4$ sec. A histogram of $N = 8$ bins represents $p_k(\rho_k)$, while the dashed line is $d(t/t_i = 2)t \in [2, t_i + T_h]$.

When a new measurement $d(t_{i+1} = 3)$ arrives (Fig. 5), the values of $\beta(k)$s that are available from the previous search (i.e. [3, 6] in Fig. 4), are not changed because, due to the Lipschitz continuity of $d(\cdot)$ w.r.t. ϕ, small changes of $\hat{\phi}(t_i/t_i)$ produce, bounded changes of $d(t_r/t_i)$. The distribution over $[t_i, t_{i+1}]$ (i.e. of time interval [2, 3] in Fig. 4) is uniformly distributed along the new extension (i.e. time interval [6, 7] in Fig. 5).

In this search d_1 was used, as measure of distance, because it is computationally faster, while it gives the same collision time as d_2.

Finally an issue of interest is the selection of the time horizon T of search. It should be based on a number of criteria depending on safety and performance issues. If n is a positive integer, a reasonable and safe selection is:

$$T = n \cdot \frac{d(t_{i+1})}{V}, \quad V = |d(t_{i+1}) - d(t_i)|.$$

5. *Simulation results*

The case study considers a mobile robot endowed with a vision and motion control system and a moving obstacle (Fig. 6). The vision system "traces" the edges of the moving polyhedron representing the obstacle. The measurement vector z (3.2) is composed of the parameters of the mathematical description of the edges. The measurement noise v was assumed to be white, Gaussian, with zero mean and diagonal covariance matrix with diagonal elements equal to 0.1.

The Kalman filter output behaved well especially in the straight line motions. The lower bounds of $E(d_2(X))$ calculated by (3.4), are plotted in Fig. 7(a), while the error is plotted in Fig. 7(b). As expected, the lower bound underestimates and is close enough to the actual distance. The maximum absolute error is 7 cm in a range of actual distances of 20 m.

The collision time was calculated using the modified AESRS that was outlined in the previous chapter. As a metric, $d_1(t)$ was used. Therefore using a SUN 3-150 we were able to perform 30 iterations/sec. At time instants $t = 4, 5, 6$ sec the collision time was found to be $t_c =$ 7, 6.8, 6.69 sec, respectively, while the actual collision time is $t_c = 6.63$ sec.

6. *Conclusions and future research*

A theoretical analysis of the distance estimation and collision time prediction problems between moving polyhedra was presented along with simulation results. The whole analysis was based on an uncertainty model including sensing and attitude uncertainty. The computational efficiency resulting from the use of L_1 and L_∞ metrics and the heuristic amendments to AESRS was verified with a simulation study. Future research should include application of the above results in real time systems. Such an effort is currently in progress in the NASA-CIRSSE Laboratories of RPI. The above analysis should be extended for the case of nonconvex objects that can be decomposed into a number of convex ones. Further investigation is also needed for the calculation of the collision time. Although the modified AESRS performed well, it really makes big use of CPU. Therefore a parellel version of AESRS should be developed.

Decision schemes for collision avoidance, using the information calculated here in the framework described in the introduction have been reported (Kyriakopoulos, 1991; Kyriakopoulos and Saridis, 1991). Further applications are under development and results are going to be reported in the near future.

Acknowledgements—This work was supported by NASA grant NAGW-1333 1988. K. J. K. was partially supported by a fellowship of Alexander Onassis Foundation.

References

Athans, M., R. Wishner and A. Bertolini (1968). Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Trans. Aut. Control*, **AC-13,** 504–514.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE J. Robotics Automation*, **RA-2**.

Canny, J. (1986). Collision detection for moving polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-8,** 200–209.

Chen, Y. C. and M. Vidyasagar (1988). Optimal trajectory planning for planar N-link revolute manipulators in the presence of obstacles. *Proc. IEEE* 1988 *Int. Conf. on Robotics and Automation*, Philadelphia, pp. 202–208.

Clarke, F. H. (1983). *Optimization and Nonsmooth Analysis*. Wiley, New York.

Clarke, F. H. (1975). Generalized gradients and applications. *Trans. Am. Math. Soc.*, **205**.

Durrant, H. F. (1988). Uncertain geometry in robotics. *IEEE J. Robotics Automation*, **RA-4**.

Gilbert, E. G. and D. W. Johnson (1985). Distance functions and their application to robot path planning in the presence of obstacles. *IEEE J. Robotics Automation*, **RA-1**.

Gilbert, E. G., D. W. Johnson and S. S. Keerthi (1988). A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J. Robotics Automation*, **RA-4,** 193–203.

Kant, K. and S. Zucker (1988). Planning collision-free trajectories in time varying environments: A two level hierarchy. *Proc. IEEE* 1988 *Int. Conf. on Robotics and Automation*, Philadelphia, pp. 1644–1659.

Khatib, O. (1985). Real time obstacle avoidance for manipulators and mobile robots. *Proc. IEEE* 1985 *Int. Conf. on Robotics and Automation*, St. Louis, pp. 500–505.

Kyriakopoulos, K. J. and G. N. Saridis (1989a). An efficient minimum distance and collision estimation technique for on-line motion planning of robotic manipulators. NASA-CIRSEE report no. 19.

Kyriakopoulos, K. J. and G. N. Saridis (1989b). Distance estimation and collision prediction for on-line robotic motion planning. NASA-CIRSSE report no. 25.

Kyriakopoulos, K. J. and G. N. Saridis (1990). On-line motion planning for mobile robots in nonstationary environments. *Proc.* 1990 *Int. Symp. on Intelligent Control.*, Philadelphia, pp. 694–700.

Kyriakopoulos, K. J. (1991). A supervisory control strategy for navigation of mobile robots in dynamic environments. Ph.D. Thesis, ECSE, REP, NY.

Kyriakopoulos, K. J. and G. N. Saridis (1991). Collision avoidance of mobile robots in nonstationary environments. *Proc. IEEE* 1991 *Int. Conf. on Robotics and Automation*.

Lozano-Perez, T. (1987). A simple motion planning algorithm for general robot manipulators. *IEEE J. Robotics Automation*, **RA-3**.

Lumelsky, V. J. (1987). Effect of kinematics on motion planning for planar robot arms moving amidst unknown obstacles. *IEEE J. Robotics Automation*, **RA-3**.

Saridis, G. N. (1977). Expanding subinterval random search for system identification and control. *IEEE Trans. Aut. Control*, 405–412.