

CIRSSSE - R-23

NASA-CR-192737

NAGW-1333

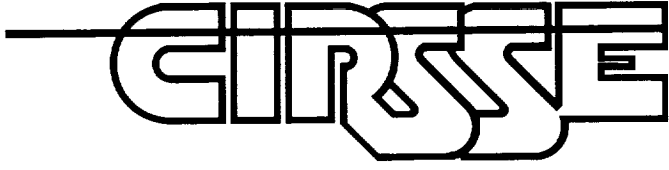
TECHNICAL REPORTS
UNIVERSITY COLLECTION

TECHNICAL REPORTS

GRANT
IN 63-CR

DATE OVERRIDE
153757

P-28



N93-21390

Unclas

G3/63 0153757

(NASA-CR-192737) DISTANCE
ESTIMATION AND COLLISION PREDICTION
FOR ON-LINE ROBOTIC MOTION PLANNING
(Rensselaer polytechnic Inst.)
28 p

Center for Intelligent Robotic Systems for Space Exploration

Rensselaer Polytechnic Institute
Troy, New York 12180-3590

Technical Reports
Engineering and Physical Sciences Library
University of Maryland
College Park, Maryland 20742

**DISTANCE ESTIMATION AND
COLLISION PREDICTION FOR
ON-LINE ROBOTIC MOTION
PLANNING**

by

K.J. Kyriakopoulos and G.N. Saridis

**Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering
Troy, New York 12180-3590**

January 1991

CIRSSE REPORT #25

02/91

DISTANCE ESTIMATION AND COLLISION PREDICTION FOR ON-LINE ROBOTIC MOTION PLANNING

K.J.Kyriakopoulos and G.N.Saridis

NASA CIRSSE
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, USA

Abstract

An efficient method for computing the minimum distance and predicting collisions between moving objects is presented. This problem has been incorporated [Kyriakopoulos 90b, 91] in the framework of an in-line motion planning algorithm to satisfy collision avoidance between a robot and moving objects modeled 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 minimum 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 possible collisions with the moving obstacles and estimate the collision time.

Keywords: Collision Avoidance, Collision Prediction, Distance Functions, Random Search.

1 INTRODUCTION

The trajectory planning problem for navigation of mobile robots or manipulation of robotic arms has been traditionally treated for environments of stationary or deterministically moving objects [Brooks 86, Chen 88, Kant 88, Khatib 85, Kyriakopoulos 88, Lozano-Perez 87, Lumelsky 87, Wu 88]. 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 menu of 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 at 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 first on the already processed information which has obtained the form of an off-line plan, and second on some additional local measurements and estimates of measures related to the obstructing moving objects. Such measures have been proposed [Kyriakopoulos 90, 90b, 91] 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 86, Gilbert 87, Gilbert 89, Kyriakopoulos 89] 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 hands, 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 constitute topics of future research.

2 DISTANCE FUNCTIONS - THE DETERMINISTIC CASE

2.1 Definitions

The distance between two objects at the time instant t is defined [Gilbert 85] as

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

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

$$d^o - d(t) \leq 0 \quad \forall t$$

where $d^o > 0$ is a safety constant.

Sets $K_i(t), K_j(t)$ are described by an equation of the form

$$K(t) = R(t) \cdot C + \{T(t)\} \quad (2.2)$$

where

$C \subset \mathfrak{R}^3$, is a compact set and describes the shape of the object

$R(t) \in \mathfrak{R}^{3 \times 3}$, is a rotation matrix

$T(t) \in \mathfrak{R}^3$ is a translation vector.

The basic assumption adopted in this work [Kyriakopoulos 89] is that set C representing the shape of the object is a convex polyhedron described by

$$C = \{x/x \in \mathfrak{R}^3 \ni A \cdot x \leq b, \quad A \in \mathfrak{R}^{m \times 3}, b \in \mathfrak{R}^m\}. \quad (2.3)$$

A point $x \in \mathfrak{R}^3$ 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

$$K(t) = \{x/x \in \mathbb{R}^3 \ni A \cdot R^{-1}(t) \cdot x \leq b - A \cdot R^{-1}(t) \cdot T(t)\}.$$

2.2 Representations of distance

Based on definitions (2.1) and (2.3) and on equation (2.2), the problem to find the minimum distance between two solids

$$C_r = \{x/x \in \mathbb{R}^3 \ni A_r \cdot x \leq b_r, \quad A_r \in \mathbb{R}^{m \times 3}, b_r \in \mathbb{R}^m\}$$

$$C_o = \{y/y \in \mathbb{R}^3 \ni A_o \cdot y \leq b_o, \quad A_o \in \mathbb{R}^{l \times 3}, b_o \in \mathbb{R}^l\}.$$

representing the robot and an obstacle respectively, is recasted to a mathematical programming problem of the form:

$$\begin{aligned} d(t) = \min_{x,y} \quad & \|x - y\| \\ \text{s.t.} \quad & A_r(t) \cdot x(t) \leq b_r(t) \\ & A_o(t) \cdot y(t) \leq b_o(t) \end{aligned} \quad (2.4)$$

where

$$\begin{aligned} A_r(t) &= A_r \cdot R_r^{-1}(t) \\ b_r(t) &= b_r - A_r \cdot R_r^{-1}(t) \cdot T_r(t) \\ A_o(t) &= A_o \cdot R_o^{-1}(t) \\ b_o(t) &= b_o - A_o \cdot R_o^{-1}(t) \cdot T_o(t) \end{aligned} \quad (2.5)$$

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 (2.4) and (2.5), the distance between the solids C_r and C_o can be viewed as a function of their configuration [Gilbert 85] described as a quadruple $P = (R_o(t), T_o(t), R_r(t), T_r(t))$:

$$d = d(P) : \mathcal{P} \equiv \mathbb{R}^{3 \times 3} \times \mathbb{R}^3 \times \mathbb{R}^{3 \times 3} \times \mathbb{R}^3 \rightarrow \mathbb{R}^+ \quad (2.6)$$

and therefore a domain \mathcal{P}^+ of d can be defined as:

$$\mathcal{P}^+ = \{P \in \mathcal{P} : d(P) > 0\} \quad (2.7)$$

The norm used in mathematical program (2.4) was not explicitly defined here. It can be any norm because norms in \mathbb{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 appendix A [Kyriakopoulos 89]. 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 (2.4) and (2.5) 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 3-D case which adds complexity just to the formulas. On the other hand, the 2-D case perfectly serves the navigation problem considered here.

The equations of motion of the moving obstacle are :

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

where

p_x, p_y, p_θ define the cartesian position and orientation in 2-D
 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:

$$\dot{X} = A \cdot X \quad (3.2)$$

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

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

$$z = h(X) + v \quad (3.3)$$

where $z \in \mathbb{R}^m$ 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 88]. 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 proposed [Athans et.al 68] may well be used to get a close to optimal estimate \hat{X} of state 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 :

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

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

$$\begin{aligned}\dot{\hat{X}}(t/t_i) &= e^{A \cdot (t-t_i)} \cdot \hat{X}(t_i/t_i) & t \geq t_i \\ \dot{\hat{P}}(t/t_i) &= A \cdot \hat{P}(t_i/t_i) + \hat{P}(t_i/t_i) \cdot A^T\end{aligned}\quad (3.4)$$

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 functions

From (2.6), 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 2-D case:

$$R_o(t) = R_o(\theta(t)) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\quad (3.5)$$

$$T_o(t) = T_o(p_x(t), p_y(t)) = \begin{bmatrix} p_x \\ p_y \end{bmatrix}\quad (3.6)$$

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 83, Gilbert et.al 85] has shown that d although uniformly Lipschitz continuous, is not differentiable everywhere on its domain \mathcal{P}^+ defined by (2.7). However, \mathcal{P}^+ can be viewed as a subset of \mathfrak{R}^n and by Rademachers theorem [Clarke 83], function

$d(X)$ is differentiable almost every where (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 neighborhoods 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 :

$$d(X) \approx d(\hat{X}) + (X - \hat{X})^T \cdot \frac{\partial d(\hat{X})}{\partial X} + \frac{1}{2}(X - \hat{X})^T \cdot \frac{\partial^2 d(\hat{X})}{\partial X^2} \cdot (X - \hat{X}) \quad (3.7)$$

Taking expected values in both sides of (3.7),

$$E(d(X)) \geq d(\hat{X}) + \frac{1}{2} \lambda_{\min} \left(\frac{\partial^2 d(\hat{X})}{\partial X^2} \right) \cdot \text{tr}(P(t)) \quad (3.8)$$

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

In order to make possible the utilization of (3.8), 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 2-D case and is consistent with the development of [Gilbert et.al 85]. The 3-D extension is straightforward.

Define $\eta_a : \mathcal{P}^+ \times C_o \times C_r \rightarrow R^+$ by:

$$\eta_a(P, w_o, w_r) = \|R_o \cdot w_o + p_o - R_r \cdot w_r - p_r\|_a \quad (3.9)$$

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

$$\nabla_X \eta_2(P, w_o, w_r) = \left(\frac{L_x}{\eta_2}, 0, 0, \frac{L_y}{\eta_2}, 0, 0, \frac{-M_x \cdot L_x + M_y \cdot L_y}{\eta_2}, 0, 0 \right)^T \quad (3.10)$$

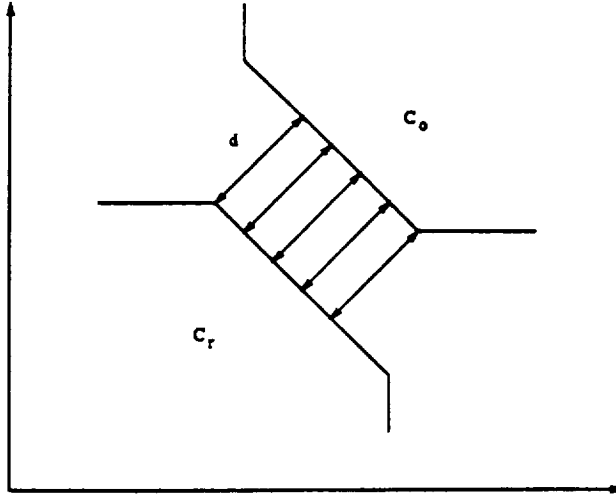


Figure 1: Singular Configuration

where

$$M_x = w_o^x \cdot \sin\theta + w_o^y \cdot \cos\theta \quad (3.11)$$

$$M_x = w_o^x \cdot \cos\theta - w_o^y \cdot \sin\theta \quad (3.12)$$

$$w_o = [w_o^x \quad w_o^y]^T$$

the Hessian can be found by further differentiation and is given in appendix B with its eigenvalues.

Based on [Clarke 75, Gilbert et.al 85] 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 on figure 1.

Case $a = 1$: In this case

$$\eta_1(P, w_o, w_r) = |L_x| + |L_y| \quad (3.13)$$

and

$$\nabla_X \eta_1(P, w_o, w_r) = (\text{sgn}(L_x), 0, 0, \text{sgn}(L_y), 0, 0, \frac{\partial \eta_1}{\partial \theta}, 0, 0)^T \quad (3.14)$$

where

$$\frac{\partial \eta_1}{\partial \theta} = -(w_o^x \text{sgn}(L_x) + w_o^y \text{sgn}(L_y)) \cdot \sin \theta + (w_o^x \text{sgn}(L_y) - w_o^y \text{sgn}(L_x)) \cdot \cos \theta \quad (3.15)$$

Similarly, the Hessian is very easy to obtain and is given in appendix B with its eigenvalues.

In appendix C the following theorem is proved:

Theorem 3.1 : 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 $a = \infty$: In this case

$$\eta_\infty(P, w_o, w_r) = \max\{|L_x|, |L_y|\} \quad (3.16)$$

and

$$\nabla_X \eta_\infty(P, w_o, w_r) = \begin{cases} [\text{sgn}(L_x), 0, 0, 0, 0, 0, \frac{\partial \eta_\infty}{\partial \theta}, 0, 0]^T & \text{if } \eta_\infty(P, w_o, w_r) = |L_x| \\ [0, 0, 0, \text{sgn}(L_y), 0, 0, \frac{\partial \eta_\infty}{\partial \theta}, 0, 0]^T & \text{if } \eta_\infty(P, w_o, w_r) = |L_y| \end{cases} \quad (3.17)$$

where

$$\frac{\partial \eta_\infty}{\partial \theta} = \begin{cases} \text{sgn}(L_x) \cdot \{-w_o^x \cos \theta - w_o^y \sin \theta\} & \text{if } \eta_\infty(P, w_o, w_r) = |L_x| \\ \text{sgn}(L_y) \cdot \{w_o^x \cos \theta - w_o^y \sin \theta\} & \text{if } \eta_\infty(P, w_o, w_r) = |L_y| \end{cases} \quad (3.18)$$

The Hessian is given in appendix B with its eigenvalues.

In appendix C the following corollary is proved using the theorem 3.1 :

Corollary 3.2 : 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

$$\mathcal{W}(L) = \arg \min\{|L_x|, |L_y|\}$$

$$W(P) \triangleq \{(w_o, w_r) \in C_o \times C_r : \eta_\infty(P, w_o, w_r) = d_\infty(P)\}$$

are singleton, with $\mathcal{W}(L) = \{\ell^*\}$ and $W(P) = \{(w_o^*, w_r^*)\}$ then

$$\nabla_X d(X) = \nabla_X \ell^*(P, w_o^*, w_r^*).$$

3.5 Distance Estimation Algorithm

In summary, the steps of the distance estimation (prediction) algorithm are given:

- KALMAN FILTER provides $\hat{X}(t/t_i)$, $\hat{P}(t/t_i)$ $t \geq t_i$ (equation (3.4))
- from a mathematical programming stage $d_2(\hat{X})$ (or $d_1(\hat{X})$) is obtained (equation (2.4))
- from one of (B.1),(B.2),(B.3) $\lambda_{\min} \left(\frac{\partial^2 d(\hat{X})}{\partial X^2} \right)$ is obtained, and
- from (3.8) 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. First, the future motion of the robot which is preplanned and therefore accurately known. Second, the motion of the moving obstacle, as can be “optimally” predicted by (3.4). 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,

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

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. Additionally some methods that have recently appeared, [Gilbert 89] although they 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 77] was utilized with modifications to accommodate the problem. The convergence properties of this method have well been investigated, and its performance experimentally verified. The basic idea of AESRS is the following:

Obtain the **lower bound** $d(t_i)$ from (3.8) and set $d_{min} = d(t_i)$. Split the interval $\Omega = [t_i, t_i + T]$ of concern in N subintervals $\Omega_k \subset \Omega$ such that $\Omega_i \cap \Omega_j = \emptyset$, $i \neq j$. Define a probability density function (pdf) $p(\varrho)$ over Ω by:

$$p(\varrho) = \sum_{k=1}^N \beta_k p_k(\varrho_k), \quad \varrho_k \in \Omega_k, \quad \sum_{k=1}^N \beta_k = 1 \quad (4.2)$$

where each $p_k(\varrho_k)$ is defined only on Ω_k . The pdf's $p_k(\varrho_k)$ can be chosen to be any distribution, but in this work were chosen to have **uniform distribution** for practical reasons. Initially $\beta(k) = \beta_0(k) = \frac{1}{N}$.

From (4.2) a random time instant $t_r \in \Omega$ is obtained, and from (3.8) 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 k -th bin,

$$d(t_r) \begin{cases} \leq d_{min} \Rightarrow & \begin{cases} \beta_i(k) = \beta_{i-1}(k) + \gamma_i [1 - \beta_{i-1}(k)] \\ \beta_i(j) = \beta_{i-1}(j) + \gamma_i \beta_{i-1}(j) & j \neq k \end{cases} \\ > d_{min} \Rightarrow & \beta_i(k) = \beta_{i-1}(k) \quad k = 1, \dots, N \end{cases} \quad (4.3)$$

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

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

therefore $\gamma_i = \frac{1}{i}$ and $\beta_0(k) = \frac{1}{N}$.

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 figure 2. The present time is $t_i = 2sec$ and the time horizon of interest is $T_h = 6sec$. A histogram of $N = 12$ bins represents $p_k(\varrho_k)$, while the dashed

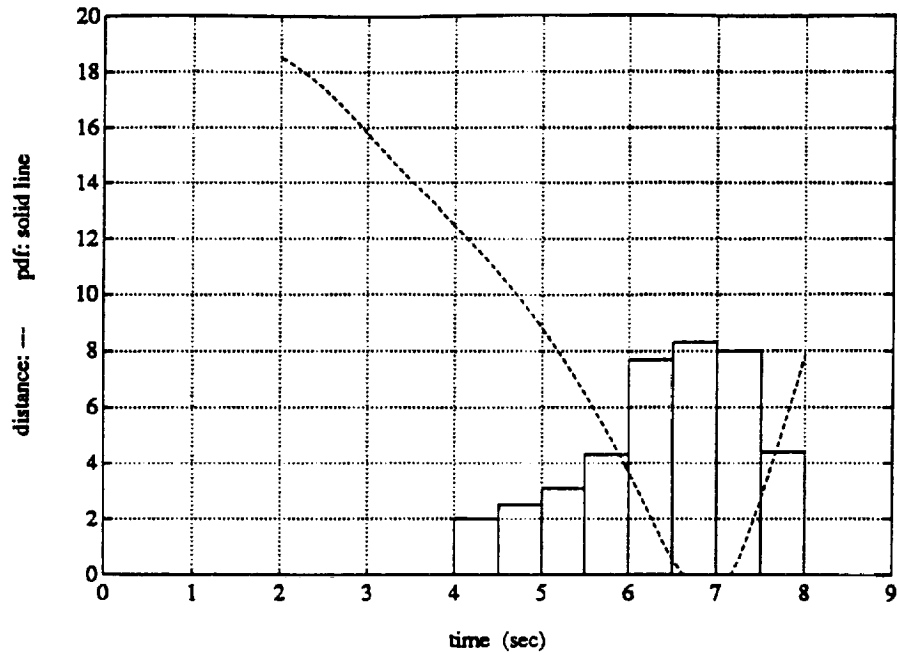


Figure 2: $d(t/t_i = 2)$ and probability histogram for $\Omega = [2, 8]$

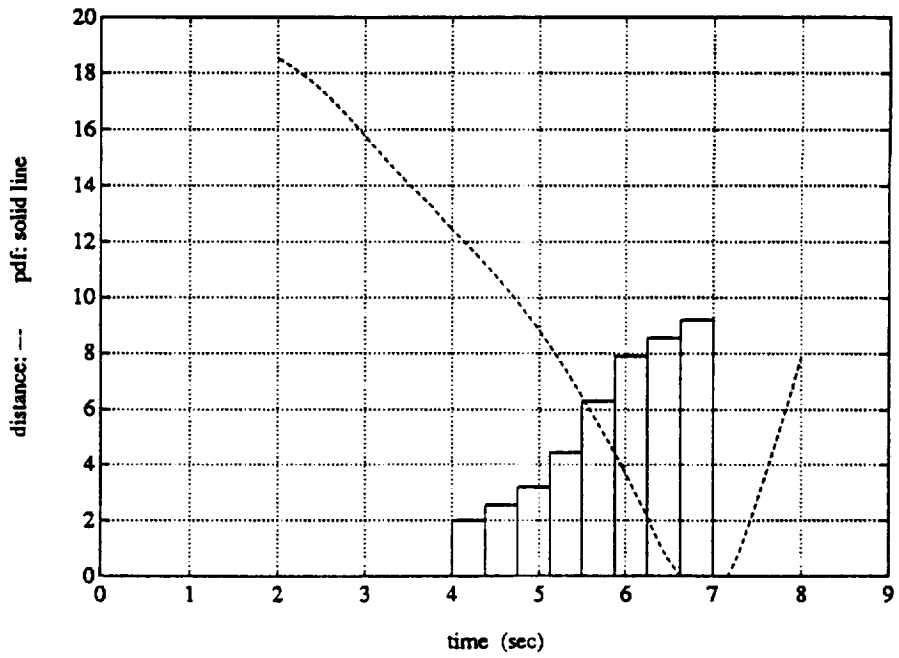


Figure 3: $d(t/t_i = 2)$ and probability histogram for $\Omega = [2, 7]$

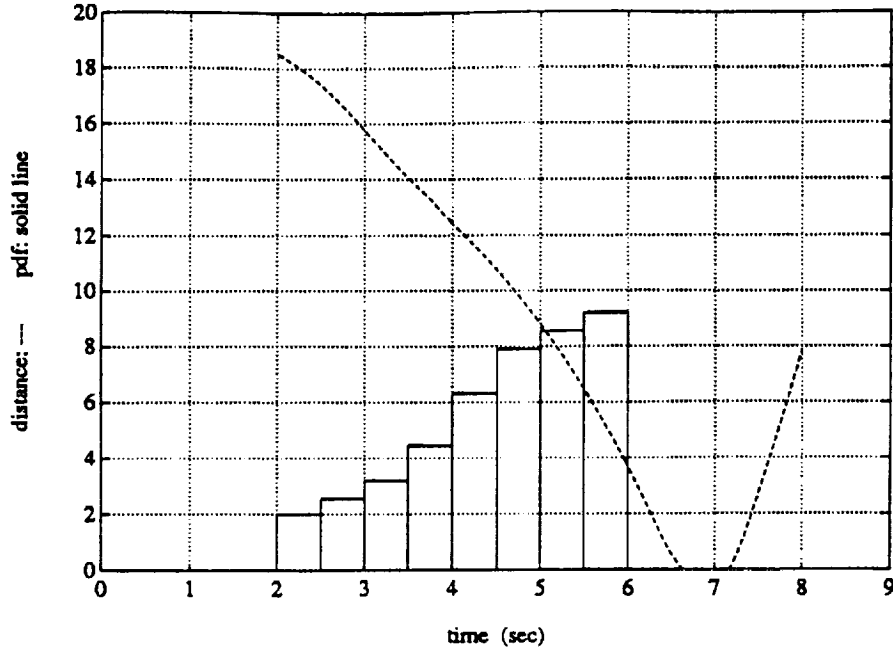


Figure 4: $d(t/t_i = 2)$ and probability histogram for $\Omega = [2, 6]$ at time $t = 3^-$.

line is $d(t/t_i = 2)$ $t \in [2, t_i + T_h]$. The lower bound of the predicted distance $d(t/t_i)$ is computed using (3.8) not everywhere but only at random instants t_r produced by $p_k(\rho_k)$.

Assume that a trial time instant $t_r = 7 \text{ sec}$ 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 (figure 3). Since the grid becomes finer and finer the uniform distribution for $p_k(\rho_k)$ is satisfactory. The shape of the distribution of $\beta(k)$ $k = 1, \dots, N$ in the new interval, is similar to this of the previous distribution providing the search with "memory".

B) Consider figure 4. The present time is $t = 3^-$, all the information processing has been done based on sensing info at $t_i = 2 \text{ sec}$ and the time horizon of interest is $T_h = 4 \text{ sec}$. 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 (figure 5), the distribution of $\beta(k)$ for the bins that available from the previous search (i.e $[3, 6]$ in figure 4), is 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

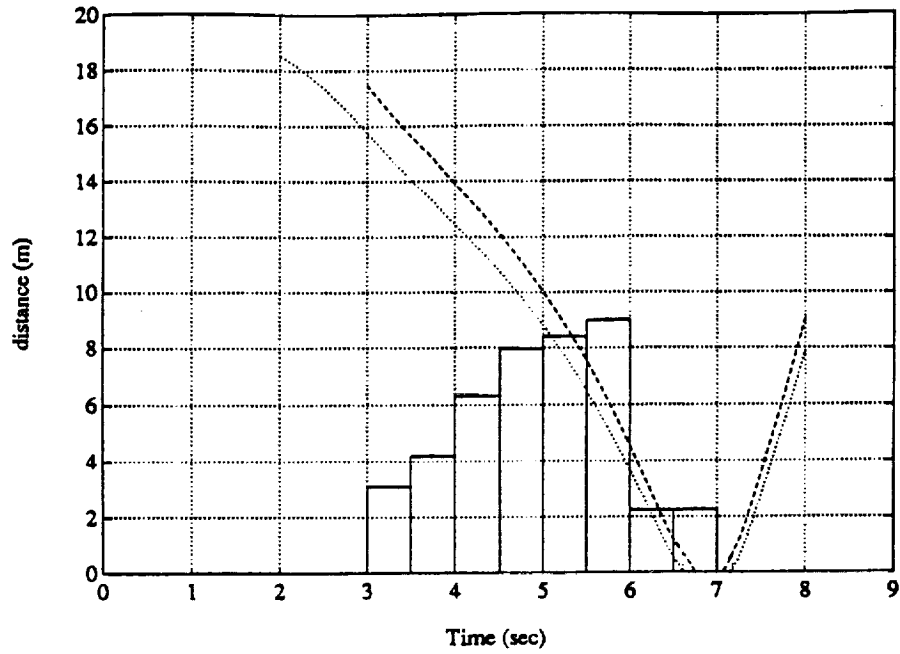


Figure 5: $d(t/t_i = 2)$ (\cdots), $d(t/t_i = 3)$ ($---$) and probability histogram for $\Omega = [3, 7]$ at time $t = 3^+$.

time interval $[2,3]$ in figure 4) is uniformly distributed along the new extension (i.e. time interval $[6,7]$ in figure 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. In this work a reasonable and safe selection was considered:

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

where n is a positive integer.

5 SIMULATION RESULTS

The case study considers a mobile robot endowed with a vision and motion control system and a moving obstacle (figure 6). The vision system “traces” the edges of the moving polyhedron representing the obstacle, and the parameters of their mathematical description, constituted the measurement vector z of equation (3.3). The

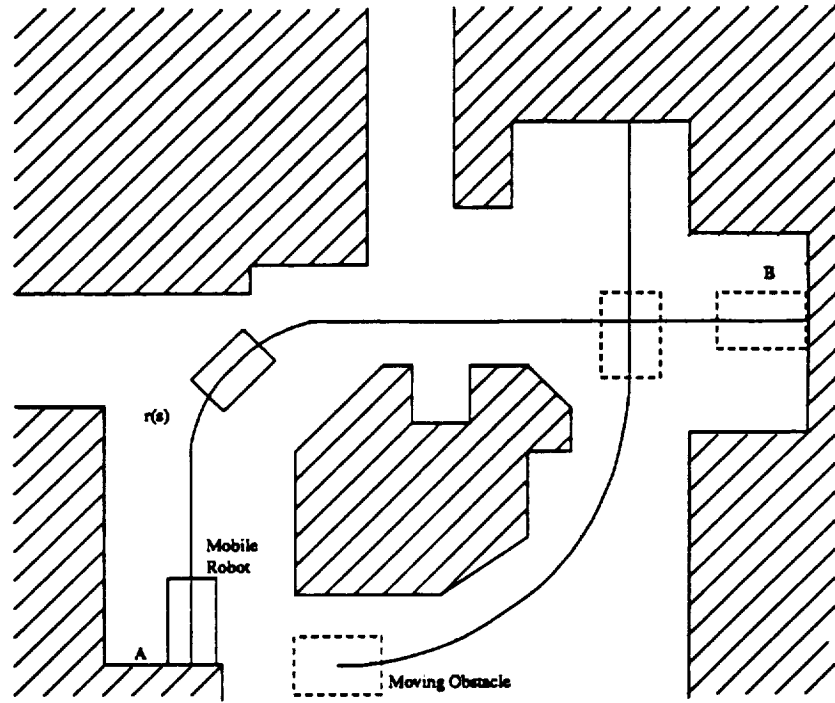


Figure 6: Environment with a mobile robot and a moving obstacle

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 (3.8), are plotted in figure 7(a), while the error is plotted in figure 7(b). As expected, the lower bound underestimates and is close enough to the actual distance. The maximum absolute error is 7cm in a range of actual distances of 20m.

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 DISCUSSION - FUTURE RESEARCH

A theoretical analysis of the distance estimation and collision time prediction problems between moving polyhedra was presented along with simulation results. The

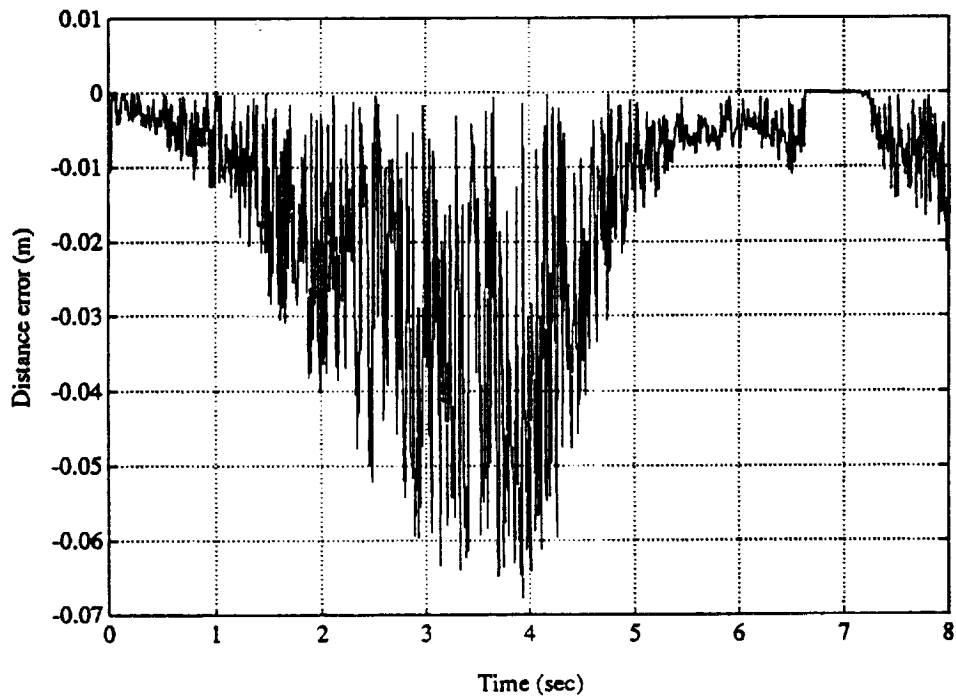
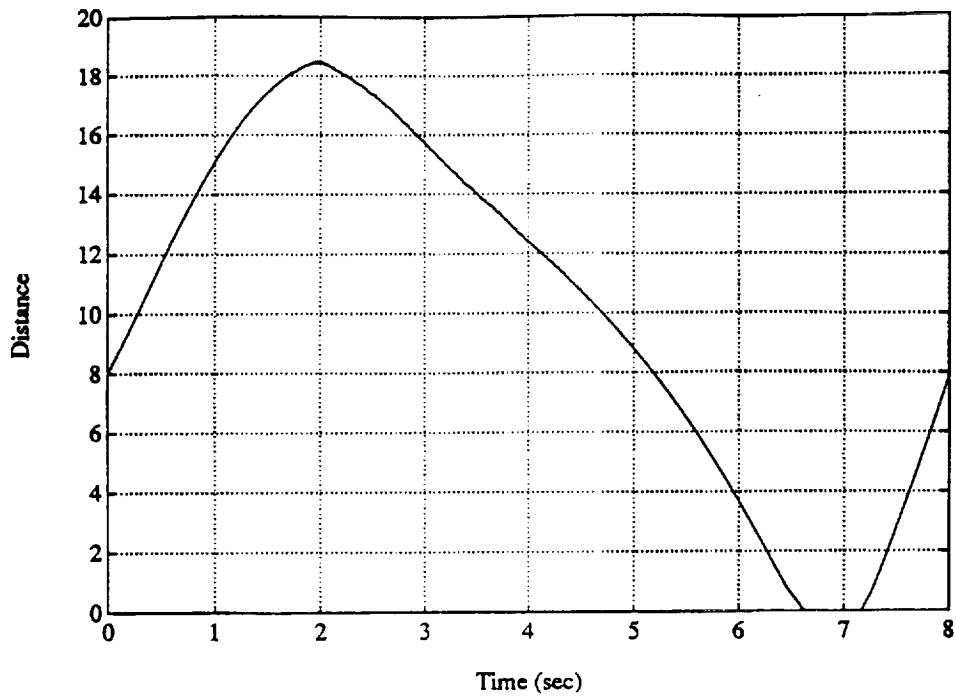


Figure 7: (a) Lower Bound of $E(d_2(X(t)))$ (b) error $E(d_2(X(t))) - d_2(X(t))$

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 to 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, either some parallelizable scheme 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 90b, 91]. Further applications are under development and results are going to be reported in the near future.

APPENDIX A

In (2.4) the distance computation problem between two convex polyhedra was stated as the mathematical programming problem:

$$\begin{aligned} d &= \min_{x,y} \|x - y\|_a \\ \text{s.t. } & A_r \cdot x \leq b_r \\ & A_o \cdot y \leq b_o \end{aligned}$$

where integer a defines the kind of sought distance, $x, y \in \mathfrak{R}^3$, $A_r \in \mathfrak{R}^{m \times 3}$, $b_r \in \mathfrak{R}^m$, $A_o \in \mathfrak{R}^{l \times 3}$, $b_o \in \mathfrak{R}^l$. Integers m, l are the numbers of the sides of the polyhedra representing the robot and the obstacle respectively.

If we set $z = [x^T \ y^T]^T$, $L = [b_r^T \ b_o^T]^T$, $D = [I_{3 \times 3} \ -I_{3 \times 3}]$ (I is the identity matrix),

$$M = \begin{bmatrix} A_r & 0_{m \times 3} \\ 0_{l \times 3} & A_o \end{bmatrix}$$

then the above problem is written in the standard mathematical programming form

$$\begin{aligned} d &= \min_z \|D \cdot z\|_a \\ \text{s.t. } & M \cdot z \leq L \end{aligned} \tag{A.1}$$

$a = 2$

Mathematical programming problem (A.1) is equivalent to

$$\begin{aligned} d &= \min_z z^T \cdot Q \cdot z \\ \text{s.t. } & M \cdot z \leq L \end{aligned}$$

with $Q = D^T \cdot D \in \mathbb{R}^{6 \times 6}$, and becomes a quadratic programming problem. Q is positive semidefinite. Therefore more than one minimizing vectors may exist, with all of them corresponding to the same minimum.

$$\underline{a = 1}$$

If a new vector of unknowns $w \in \mathbb{R}^3$ is defined

$$w_i = |D_i \cdot z| = |x_i - y_i|$$

with D_i the i -th row of D , then the mathematical programming problem (A.1) is equivalent to

$$\begin{aligned} d &= \min_{z,w} \sum_{i=1}^3 w_i \\ \text{s.t. } w_i &\geq D_i \cdot z \quad i = 1, 2, 3 \\ w_i &\geq -D_i \cdot z \quad i = 1, 2, 3 \\ M \cdot z &\leq L \end{aligned}$$

and becomes a linear programming problem of 9 variables and $m + l + 6$ constraints. More than one minimizing vectors may exist, with all of them corresponding to the same minimum.

$$\underline{a = \infty}$$

If a new unknown $w \in \mathbb{R}$ is introduced, such that

$$w = \max_{i=1,2,3} |D_i \cdot z| = \max_{i=1,2,3} |x_i - y_i|$$

with D_i the i -th row of D , then the mathematical programming problem (A.1) is equivalent to

$$\begin{aligned} d &= \min_{z,w} w \\ \text{s.t. } w &\geq D_i \cdot z \quad i = 1, 2, 3 \\ w &\geq -D_i \cdot z \quad i = 1, 2, 3 \\ M \cdot z &\leq L \end{aligned}$$

and becomes a linear programming problem of 7 variables and $m + l + 6$ constraints. More than one minimizing vectors may exist, with all of them corresponding to the same minimum.

APPENDIX B

In this appendix, the derivation of the eigenvalues of the Hessian of the distance function is presented for two cases of norms.

$a = 2$: If $G = M_x L_y + M_y L_x$ then direct differentiation of (3.10) gives:

$$\frac{\partial^2 \eta_2(X)}{\partial X^2} = \frac{1}{\eta_2^3} \cdot \begin{bmatrix} L_y^2 & -L_x L_y & -L_y G \\ -L_x L_y & L_x^2 & L_x G \\ -L_y G & L_x G & G(G - \eta_2^2) \end{bmatrix}$$

Setting $\Gamma = -G\eta_2^4$ and $B = \eta_2^2(G - 1) - G^2$,

$$\lambda_{\min}\left(\frac{\partial^2 \eta_2(X)}{\partial X^2}\right) = \frac{-B - (B^2 - 4 * \Gamma)^{\frac{1}{2}}}{2\eta_2^3} < 0 \quad (B.1)$$

$a = 1$: Direct differentiation of (3.13) in conjunction with (3.15) gives:

$$\frac{\partial^2 \eta_1(X)}{\partial X^2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{\partial^2 \eta_1}{\partial \theta^2} \end{bmatrix}$$

where

$$\frac{\partial^2 \eta_1}{\partial \theta^2} = -(w_o^x \text{sgn}(L_x) + w_o^y \text{sgn}(L_y)) \cdot \cos \theta - (w_o^x \text{sgn}(L_y) - w_o^y \text{sgn}(L_x)) \cdot \sin \theta$$

and therefore

$$\lambda_{\min}\left(\frac{\partial^2 \eta_1(X)}{\partial X^2}\right) = \min\left\{0, \frac{\partial^2 \eta_1}{\partial \theta^2}\right\}. \quad (B.2)$$

$a = \infty$: Direct differentiation of (3.16) in conjunction with (3.18) gives:

$$\frac{\partial^2 \eta_\infty(X)}{\partial X^2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{\partial^2 \eta_\infty}{\partial \theta^2} \end{bmatrix}$$

where

$$\frac{\partial^2 \eta_\infty}{\partial \theta^2} = \begin{cases} \text{sgn}(L_x) \cdot \{w_o^x \sin \theta - w_o^y \cos \theta\} & \eta_\infty(P, w_o, w_r) = |L_x| \\ \text{sgn}(L_y) \cdot \{-w_o^x \sin \theta - w_o^y \cos \theta\} & \eta_\infty(P, w_o, w_r) = |L_x| \end{cases}$$

and therefore

$$\lambda_{\min}\left(\frac{\partial^2 \eta_\infty(X)}{\partial X^2}\right) = \min\left\{0, \frac{\partial^2 \eta_\infty}{\partial \theta^2}\right\}. \quad (B.3)$$

APPENDIX C

Some definitions, theorems and lemmas that are going to be utilized in the proof of the theorem are presented here. Most of those results are based on [Clarke75] [Clarke83] [Gilb85].

Consider $f : O \rightarrow \mathfrak{R}$, be locally Lipschitz, where $O \subset \mathfrak{R}^n$. We say that f admits a **Frechet derivative** ∇f at $x \in O$ if

$$\lim_{\delta \rightarrow 0} \frac{f(x + \delta x) - f(x)}{\|\delta x\|} = \langle \nabla f, \delta x \rangle$$

where \langle, \rangle denotes the inner product. The usual (one-sided) **Directional derivative** of f at x in the direction v is

$$f'(x; v) = \lim_{t \downarrow 0} \frac{f(x + tv) - f(x)}{t}$$

The **Generalized gradient** of f at x is

$$\partial f(x) = \text{co}\{w : \nabla f(x_i) \rightarrow w, \nabla f(x_i) \text{ exists}, x_i \rightarrow x\}$$

where co denotes the convex hull. Finally the **Generalized directional derivative** $f^\circ(x; v)$ is defined as:

$$f^\circ(x; v) = \lim_{\substack{y \rightarrow x \\ t \downarrow 0}} \sup \frac{f(x + tv) - f(x)}{t}$$

Definition C-1: A function $f(x)$ is said to be **regular** at x provided:

- (i) $f'(x; v)$ exists $\forall v$.
- (ii) $f'(x; v) = f^\circ(x; v) \forall v$.

Proposition C-2: Let f be Lipschitz near x then f is regular at x .

Proof: The proof is given in [Clarke83](proposition 2.3.6).■

Proposition C-3: Let $f : X \rightarrow \mathfrak{R}$, with X finite dimensional. Then ∂f is upper semicontinuous $\forall x \in X$.

Proof: The proof is given in [Clarke83](proposition 2.1.5).■

The following theorem is key to our development

Theorem C-4: Let U be a sequentially compact space, and let $g : \mathfrak{R}^n \times U \rightarrow \mathfrak{R}$ have the following properties:

- (a) $g(x, u)$ is upper semicontinuous in (x, u) .
- (b) g is locally Lipschitz in x , uniformly for u in U .
- (c) $g_x^\circ(x, u; \cdot) = g'_x(x, u; \cdot)$, the derivatives being w.r.t x .
- (d) $\partial_x g(x, u)$ is upper semicontinuous in (x, u) .

If we let $f(x) = \max_{u \in U} \{g(x, u)\}$ then :

- (1) f is locally Lipschitz.
- (2) $f'(x; v)$ exists.
- (3) $f'(x; v) = f^\circ(x; v) = \max\{\zeta \cdot v : \zeta \in \partial_x g(x, u), u \in M(x)\}$, where $M(x) = \{u \in U : g(x, u) = f(x)\}$.
- (4) $\partial f(x) = \text{co}\{\partial_x g(x, u) : u \in M(x)\}$. ($\text{co}\{\cdot\}$ denotes the convex hull of a set).

Proof: The proof is given in [Clarke75](Theorem 2.1).■

Finally, the following proposition is the last piece of the proof under development.

Lemma C-5: The following are equivalent:

- (a) $\partial f(x) = \{\zeta\}$, is a singleton. (b) $\nabla f(x)$ exists, $\nabla f(x) = \zeta$, and ∇f is continuous at x relative to the set upon it exists.

Proof: The proof is given in [Clarke75] (Proposition 1.13). ■

Lemma C-6: The following results are true for

$$\eta_1(P, w_o, w_r) = \|R_o \cdot w_o + p_o - R_r \cdot w_r - p_r\|_1$$

- a) $\eta_1(P, w_o, w_r)$ is continuous and convex on its domain.
b) η_1 is Locally Lipschitz.
c) $\eta_{1P}^o(P, w_o, w_r; \cdot) = \eta_{1P}'(P, w_o, w_r; \cdot)$
d) $\partial_P \eta_1(P, w_o, w_r)$ is upper-semi-continuous.

Proof: (a) and (b) can be easily derived from the definition of η_1 .

(c) From C-2 η_1 is regular, and from definition C-1 QED.

(d) The domain of η_1 can be viewed as a subset of \mathfrak{R}^n . From proposition C-3 the sought result is obvious. ■

Proof of Theorem 3.1 : Lemma C-6 in conjunction to theorem C-4 gives the following results for function

$$d_1(P) = \min\{\eta_1(P, w_o, w_r) : w_o \in C_o, w_r \in C_r\}$$

- 1) d_1 is locally Lipschitz.
2) $d_1'(P; P')$ exists.
3) $d_1'(P; P') = \min\{\langle \nabla_P \eta_1(P, w_o, w_r), P' \rangle : (w_o, w_r) \in W(P)\}$ where
 $W(P) = \{(w_o, w_r) \in C_o \times C_r : \eta_1(P, w_o, w_r) = d_1(P)\}$
4) $\partial d_1(P) = \text{co}\{\langle \nabla_P \eta_1(P, w_o, w_r), P' \rangle : (w_o, w_r) \in W(P)\}$.

From lemma C-5 the following statements are shown to be equivalent:

$$\partial d_1(P) = \{\xi\}, \text{ singleton; } \nabla d_1(P) \text{ exists; } \nabla d_1(P) = \xi.$$

Result

When the configuration of C_o and C_r is such that there is only one pair of points (w_o^*, w_r^*) minimizing $d_1(P)$ (i.e when $W(P)$ is singleton) then according to the above results $\nabla_P d_1(P) = \nabla_P \eta_1(P, w_o^*, w_r^*)$. Q.E.D. ■

Before continuing with the proof of corollary 3.2 the following lemma is proved:

Lemma C-7: Let $f_1, f_2 : X \rightarrow \mathfrak{R}$. with X finite dimensional. If f_1, f_2 are continuous, convex and Lipschitz continuous then the following results are true for

function $f : X \rightarrow \mathfrak{R}$ defined by

$$f(x) = \max\{f_1(x), f_2(x)\}$$

a) f is continuous and convex on its domain.

b) f is Locally Lipschitz.

c) $f_x^o(x; \cdot) = f_x'(x; \cdot)$

d) $\partial_x f(x)$ is upper-semi-continuous.

Proof:

(a) The continuity of f is obvious. Consider $x \in X$ such that $x = \lambda \cdot x_1 + (1 - \lambda) \cdot x_2$ where $x_1, x_2 \in X$. Then $f(x) = f_i(x) \quad i \in \{1, 2\}$, but since f_i is convex,

$$f(\lambda \cdot x_1 + (1 - \lambda) \cdot x_2) = f(x) = f_i(x) \leq \lambda \cdot f_i(x_1) + (1 - \lambda) \cdot f_i(x_2)$$

but since $f(x_j) \geq f_i(x_j) \quad i, j \in \{1, 2\}$,

$$f(\lambda \cdot x_1 + (1 - \lambda) \cdot x_2) \leq \lambda \cdot f(x_1) + (1 - \lambda) \cdot f(x_2).$$

Thus f is convex.

(b) Since f_i is Lipschitz then there exists constant $c_i > 0$ such that

$$|f_i(x) - f_i(y)| \leq c_i \|x - y\|_X \quad \forall x, y \in X \quad (C.1)$$

where $\|\cdot\|_X$ is the induced norm of space X . Assume that

$$f(x) = f_i(x) \quad (C.2)$$

$$f(y) = f_j(y) \quad (C.3)$$

if $i = j$ then $f_i(y) = f_j(y)$. Otherwise, $f_j(y) > f_i(x)$ (from C.2). Therefore, in general

$$f_i(y) - f_j(y) \leq 0 \quad (C.4)$$

Consider now,

$$f(x) - f(y) \stackrel{(C.2,3)}{=} f_i(x) - f_j(y) = f_i(x) - f_i(y) + f_i(y) - f_j(y) \stackrel{(C.4)}{\leq} f_i(x) - f_i(y)$$

therefore

$$|f(x) - f(y)| \leq |f_i(x) - f_i(y)| \stackrel{(C.1)}{\leq} c_i \|x - y\|_X \quad i \in \{1, 2\}$$

and

$$|f(x) - f(y)| \leq c \|x - y\|_X \quad c = \max_{i=1,2} c_i$$

therefore f is Lipschitz.

(c) & (d) are proved the same way as in lemma C-6. ■

Proof of Corollary 3.2 : We follow the same steps as in the theorem of Theorem 3.1.

From Lemma C-7, Theorem C-4 and Lemma C-5 we come to the conclusion that when the configuration of C_o and C_r is such that there is only one pair of points (w_o^*, w_r^*) minimizing $d_\infty(P)$ (i.e when $W(P)$ is singleton) then

$$\nabla_P d_\infty(P) = \nabla_P \eta_\infty(P, w_o^*, w_r^*). \quad (C.5)$$

Similarly, when $L(P, w_o^*, w_r^*) = [L_x \ L_y]^T$ is such that only one of L_x, L_y (denote it as ℓ^*) maximizes $\eta(P)$ (i.e when $W(L)$ is singleton) then

$$\nabla_P \eta_\infty(P, w_o^*, w_r^*) = \nabla_P \ell^*(P, w_o^*, w_r^*). \quad (C.6)$$

From (C.5,C.6) the following result is obtained

$$\nabla_P d_\infty(P) = \nabla_P \ell^*(P, w_o^*, w_r^*).$$

ACKNOWLEDGMENTS

This work was supported by NASA Grant NAGW-1333 1988. The first author was partially supported by a fellowship of Alexander Onassis Foundation.

REFERENCES

- M.Athans, R.Wishner, A.Bertolini,(1968) "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements", IEEE Trans. Autom. Contr., Vol. AC-13, No. 5, October 1968.
- R.A.Brooks, (1986) "A Robust Layered Control System for a Mobile Robot", IEEE Journ. of Robotics and Automation, Vol.RA 2, No. 1, March 1986
- J.Canny, (1986) "Collision Detection for Moving Polyhedra ", IEEE-PAMI, Vol. PAMI-8, 1986.
- Y.C.Chen, M.Vidyasagar, (1988) "Optimal Trajectory Planning for Planar N-Link Revolute Manipulators in the Presence of Obstacles", Proc IEEE 1988 Intern. Conf. on Robotics and Automation.
- F.H. Clarke, (1983) "Optimization and Nonsmooth Analysis". New York: Wiley,

1983.

F.H. Clarke, (1975) "Generalized gradients and applications", Trans. Amer. Math. Soc., vol. 205, 1975.

H.F. Durrant-Whyte, (1988) "Uncertain Geometry in Robotics", IEEE Journ. of Robotics and Automation, Vol. RA 4, No.1., Febr. 1988

E.G.Gilbert, D.W.Johnson, (1985) "Distance Functions and their Application to Robot Path Planning in the Presence of Obstacles", IEEE Journ. of Robotics and Automation, Vol. RA 1, No.1., March 1985

E.G.Gilbert, D.W.Johnson and S.S.Keerthi (1988), "A Fast procedure for computing the distance between complex objects in three-dimensional space", IEEE Journ. of Robotics and Automation, Vol. RA 4, p.p. 193-203, March 1988

E.G. Gilbert, S.M.Hong, (1989) "A New Algorithm for Detecting the Collision of Moving Objects", Proc. IEEE 1989 Intern. Conf. on Robotics and Automation.

K.Kant, S.Zucker, (1988) "Planning Collision-Free Trajectories in Time Varying Environments: A Two Level Hierarchy", Proc. IEEE 1988 Intern. Conf. on Robotics and Automation.

O.Khatib,(1985) "Real Time Obstacle Avoidance for Manipulators and Mobile Robots", Proc. IEEE 1985 Intern. Conf. on Robotics and Automation.

K.J.Kyriakopoulos, G.N.Saridis, (1988) "Minimum Jerk Path Generation", Proc. IEEE 1988 Intern. Conf. on Robotics and Automation.

K.J.Kyriakopoulos, G.N.Saridis, (1989) "An Efficient Minimum Distance and Collision Estimation Technique for On-Line Motion Planning of Robotic Manipulators", NASA-CIRSSE Rept. No. 19, March 1989.

K.J.Kyriakopoulos (1990) "A Supervisory Control Strategy for Navigation of Mobile Robots in Dynamic Environments", Ph.D. Thesis, February 1991, ECSE, RPI.

K.J.Kyriakopoulos, G.N.Saridis, (1990 b) "On-Line Motion Planning for Mobile Robots in Non-stationary Environments", Proc. 1990 Intern. Symp. on Intelligent Control.

K.J.Kyriakopoulos, G.N.Saridis, (1991) "Collision Avoidance of Mobile Robots in Nonstationary Environments", Proc. IEEE 1991 Intern. Conf. on Robotics and Automation.

T.Lozano-Perez, (1987) "A Simple Motion Planning Algorithm for General Robot Manipulators", IEEE Journ. of Robotics and Automation, Vol. RA 3, No.3., June 1987

V.J.Lumelsky, (1987) "Effect of Kinematics on Motion Planning for Planar Robot Arms Moving Amidst Unknown Obstacles", IEEE Journ. of Robotics and Automa-

tion, Vol. RA 3, June 1987

G.N.Saridis, (1977) "Expanding Subinterval Random Search for System Identification and Control", IEEE Trans. Autom. Contr., pp 405-412, June 1977

C.Wu, C.Jou, (1988) "Design of a Controlled Spatial Curve Trajectory for Robot Manipulators", Proc. of 27 Conf. on Decision and Control, Dec. 1988