REAL-TIME OBSTACLE AVOIDANCE FOR MANIPULATORS AND MOBILE ROBOTS

O. Khatib

Artificial Intelligence Laboratory
Stanford University, Stanford, CA 94305

Abstarct

This paper presents a unique real-time obstacle avoidance approach for manipulators and mobile robots based on the "artificial potential field" concept. In this approach, collision avoidance, traditionally considered a high level planning problem, can be effectively distributed between different levels of control, allowing real-time robot operations in a complex environment. We have applied this obstacle avoidance scheme to robot arm using a new approach to the general problem of real-time manipulator control. We reformulated the manipulator control problem as direct control of manipulator motion in operational space--the space in which the task is originally described--rather than as control of the task's corresponding joint space motion obtained only after geometric and kinematic transformation. This method has been implemented in the COSMOS system for a PUMA 560 robot. Using visual sensing, real-time collision avoidance demonstrations on moving obstacles have been performed.

Introduction

In previous research, robot collision avoidance has been a component of higher levels of control in hierarchical robot control systems. It has been treated as a planning problem, and research in this area has focused on the development of collision-free path planning algorithms[1,5,11,12]. These algorithms aim at providing the low level control with a path that will enable the robot to accomplish its assigned task free from any risk of collision.

From this perspective, the function of low level control is limited to the execution of elementary operations for which the paths have been precisely specified. The robot's interaction with its environment is then paced by the time-cycle of high level control, which is generally several orders of magnitude slower than the response time of a typical robot. This places limits on the robot's real-time capabilities for precise, fast, and highly interactive operations in a cluttered and evolving environment. We will show, however, that it is possible to greatly extend the function of low level control and to carry out more complex operations by coupling environment sensing feedback with the lowest level of control.

Increasing the capability of low level control has been the impetus for the work on real-time obstacle avoidance that we discuss here. Collision avoidance at the low level of control is not intended to replace high level functions or to solve planning problems. The purpose here is to make better use of low level control capabilities in performing real-time operations. At this low level of control, the degree or *level of competence*[2] will remain less than that of higher level control.

The *operational space formulation* is the basis for the application of the potential field approach to robot manipulators. This formulation has its roots in the work on end-effector motion control and obstacle avoidance[6,7] that we implemented for an MA23 manipulator at the Laboratoire d'Automatique de Montpellier in 1978. The operational space approach has been formalized by constructing its basic tool, the equations of motion in the operational space of the manipulator end-effector. Details of this work have been published elsewhere[8,9]; we will briefly review the fundamentals of the operational space formulation.

Operational Space Formulation

An *operational coordinate system* is a set x of m_0 independent parameters describing the manipulator end-effector position and orientation in a frame of reference R_0. For a non-redundant manipulator, these parameters form a set of configuration parameters in a domain of the operational space[9] and constitute, therefore, a system of generalized coordinates. The kinetic energy of the holonomic articulated mechanism is a quadratic form of the generalized velocities:

$$T(x, \dot{x}) = \frac{1}{2}\dot{x}^T \Lambda(x)\dot{x}; \tag{1}$$

where $\Lambda(x)$ designates the symmetric matrix of the quadratic form, *i.e.* the kinetic energy matrix. Using the Lagrangian formalism, the end-effector equations of motion are given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = \boldsymbol{F}; \tag{2}$$

where the *Lagrangian* $L(x, \dot{x})$ is:

$$L(x, \dot{x}) = T(x, \dot{x}) - U(x); \tag{3}$$

and $U(x)$ represents the potential energy of the gravity. $\boldsymbol{F}$ is the operational force vector. These equations can be developed[8,9] and written in the form:

$$\Lambda(x)\ddot{x} + \mu(x, \dot{x}) + p(x) = \boldsymbol{F}; \tag{4}$$

where $\mu(x, \dot{x})$ represents the centrifugal and Coriolis forces, $p(x)$ the gravity forces.

Control of the manipulator in operational space is based on the selection of $\boldsymbol{F}$ as a command vector. In order to produce this command vector, specific forces Γ must be applied with joint-based actuators. The relationship between $\boldsymbol{F}$ and the joint forces Γ is given by:

$$\Gamma = J^T(q)\,\boldsymbol{F}; \tag{5}$$

where $\boldsymbol{q}$ is the vector of the n joint coordinates, and $J(\boldsymbol{q})$ the Jacobian matrix.

The decoupling of the end-effector motion in operational space is achieved by using the following structure of control:

$$\boldsymbol{F} = \Lambda(\boldsymbol{x})\boldsymbol{F}^* + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{p}(\boldsymbol{x}); \tag{6}$$

where $\boldsymbol{F}^*$ represents the command vector of the decoupled end-effector which becomes equivalent to a *single unit mass*.

The extension of the operational space approach to redundant manipulator control is discussed in [8,9].

The Artificial Potential Field Approach

We present this method in the context of manipulator collision avoidance. Its application to mobile robots is straightforward. The philosophy of the artificial potential field approach can be schematically described as follows.

The manipulator moves in a field of forces. The position to be reached is an attractive pole for the end-effector, and obstacles are repulsive surfaces for the manipulator parts.

Let us first consider the collision avoidance problem of a manipulator end-effector with a single obstacle O. If $\boldsymbol{x}_d$ designates the goal position, the control of the manipulator end-effector with respect to the obstacle O can be achieved by subjecting it to the artificial potential field:

$$U_{art}(\boldsymbol{x}) = U_{\boldsymbol{x}_d}(\boldsymbol{x}) + U_O(\boldsymbol{x}). \tag{7}$$

This leads to the following expression of the potential energy in the Lagrangian (3):

$$U(\boldsymbol{x}) = U_{art}(\boldsymbol{x}) + U_g(\boldsymbol{x}); \tag{8}$$

where $U_g(\boldsymbol{x})$ represents the gravity potential energy. Using Lagrange's equations (2), and taking into account the end-effector dynamic decoupling (6), the command vector $\boldsymbol{F}^*$ of the decoupled end-effector that corresponds to applying the artificial potential field U_{art} (7) can be written as:

$$\boldsymbol{F}^* = \boldsymbol{F}^*_{\boldsymbol{x}_d} + \boldsymbol{F}^*_O; \tag{9}$$

with:

$$\begin{aligned} \boldsymbol{F}^*_{\boldsymbol{x}_d} &= -\boldsymbol{grad}[U_{\boldsymbol{x}_d}(\boldsymbol{x})]; \\ \boldsymbol{F}^*_O &= -\boldsymbol{grad}[U_O(\boldsymbol{x})]; \end{aligned} \tag{10}$$

$\boldsymbol{F}^*_{\boldsymbol{x}_d}$ is an attractive force allowing the point $\boldsymbol{x}$ of the end-effector to reach the goal position $\boldsymbol{x}_d$, and $\boldsymbol{F}^*_O$ represents a *Force Inducing an Artificial Repulsion from the Surface* of the obstacle (FIRAS, from the French), created by the potential field $U_O(\boldsymbol{x})$. $\boldsymbol{F}^*_{\boldsymbol{x}_d}$ corresponds to the proportional term, *i.e.* $-k(\boldsymbol{x} - \boldsymbol{x}_d)$, in a conventional PD servo, where k is the position gain. The attractive potential field $U_{\boldsymbol{x}_d}(\boldsymbol{x})$ is simply:

$$U_{\boldsymbol{x}_d}(\boldsymbol{x}) = \frac{1}{2}k(\boldsymbol{x} - \boldsymbol{x}_d)^2. \tag{11}$$

$U_O(\boldsymbol{x})$ is selected such that the artificial potential field $U_{art}(\boldsymbol{x})$ is a positive continuous and differentiable function which attains its zero minimum when $\boldsymbol{x} = \boldsymbol{x}_d$. The articulated mechanical system subjected to $U_{art}(\boldsymbol{x})$ is stable. Asymptotic stabilization of the system is achieved by adding dissipative forces proportional to $\dot{\boldsymbol{x}}$. Let ξ be the velocity gain; the forces contributing to the end-effector motion and stabilization are of the form:

$$\boldsymbol{F}^*_{\boldsymbol{x}_d} = -k(\boldsymbol{x} - \boldsymbol{x}_d) - \xi\dot{\boldsymbol{x}}. \tag{12}$$

This command vector is inadequate to control the manipulator for tasks that involve large end-effector motion toward a goal position without path specification. For such a task, it is better for the end-effector to move in a straight line, with an upper speed limit.

Rewriting equation (12) leads to the following expression, which can be interpreted as speciling a desired velocity vector in a pure velocity servo-control.

$$\dot{\boldsymbol{x}}_d = \frac{k}{\xi}(\boldsymbol{x}_d - \boldsymbol{x}). \tag{13}$$

Let V_{max} designate the assigned speed limit. The limitation of the end-effector velocity magnitude can then be obtained[6] by:

$$\boldsymbol{F}^*_{\boldsymbol{x}_d} = -\xi(\dot{\boldsymbol{x}} - \nu\dot{\boldsymbol{x}}_d); \tag{14}$$

where:

$$\nu = min\left(1, \frac{V_{max}}{\sqrt{\dot{\boldsymbol{x}}_d^T\dot{\boldsymbol{x}}_d}}\right). \tag{15}$$

With this scheme, shown in Fiqure 1, the velocity vector $\dot{\boldsymbol{x}}$ is controlled to be pointed toward the goal position while its magnitude is limited to V_{max}. The end-effector will then travel at that speed, in a straight line, except during the acceleration and deceleration segments or when it is inside the repulsive potential field regions of influence.

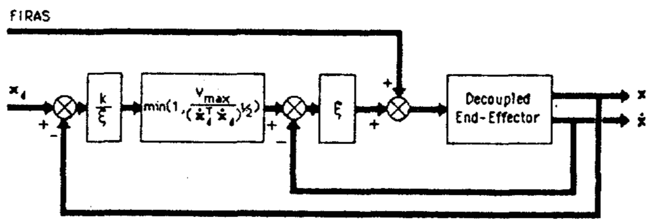

Figure 1. *End-effector Control for a Goal Position*

FIRAS Function

The artificial potential field $U_O(\boldsymbol{x})$ should be designed to meet the manipulator stability condition and to create at each point on the obstacle's surface a potential barrier which becomes negligible beyond that surface. Specifically, $U_O(\boldsymbol{x})$ should be a non-negative continuous and differentiable function whose value tends to infinity as the end-effector approaches the obstacle's surface. In order to avoid undesirable perturbing forces beyond the obstacle's vicinity, the influence of this potential field must be limited to a given region surrounding the obstacle.

Using analytic equations $f(x) = 0$ for obstacle description, the first artificial potential field function we used[7] was based on the values of the function $f(x)$:

$$U_O(x) = \begin{cases} \frac{1}{2}\eta(\frac{1}{f(x)} - \frac{1}{f(x_0)})^2, & \text{if } f(x) \leq f(x_0); \\ 0, & \text{if } f(x) > f(x_0). \end{cases} \tag{16}$$

The region of influence of this potential field is bounded by the surfaces $f(x) = 0$ and $f(x) = f(x_0)$, where x_0 is a given point in the vicinity of the obstacle and η a constant gain. This potential function can be obtained very simply in real-time since it does not require any distance calculations. However, this potential is difficult to use for asymmetric obstacles, where the separation between an obstacle's surface and equipotential surfaces can vary widely.

Using the shortest distance to an obstacle $\mathcal{O}$, we have proposed[8] the following artificial potential field:

$$U_O(x) = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho} - \frac{1}{\rho_0})^2, & \text{if } \rho \leq \rho_0; \\ 0, & \text{if } \rho > \rho_0; \end{cases} \tag{17}$$

where ρ_0 represents the limit distance of the potential field influence and ρ, the shortest distance to the obstacle $\mathcal{O}$.
Any point of the robot can be subjected to the artificial potential field. A *Point Subjected to the Potential* is called a PSP. The control of a PSP with respect to an obstacle $\mathcal{O}$ is achieved using the FIRAS function:

$$F^*_{(O,psp)} = \begin{cases} \eta(\frac{1}{\rho} - \frac{1}{\rho_0})\frac{1}{\rho^2}\frac{\partial \rho}{\partial x}, & \text{if } \rho \leq \rho_0; \\ 0, & \text{if } \rho > \rho_0; \end{cases} \tag{18}$$

where $\frac{\partial \rho}{\partial x}$ denotes the partial derivative vector of the distance from the PSP to the obstacle:

$$\frac{\partial \rho}{\partial x} = [\frac{\partial \rho}{\partial x}\ \frac{\partial \rho}{\partial y}\ \frac{\partial \rho}{\partial z}]^T. \tag{19}$$

Observing (6) and (9), the joint forces corresponding to $F^*_{(O,psp)}$ are obtained using the Jacobian matrix associated with this PSP. These forces are given by:

$$\Gamma_{(O,psp)} = J^T_{psp}(q)\Lambda(x)F^*_{(O,psp)}. \tag{20}$$

Obstacle Geometric Modelling

Obstacles are described by the composition of *primitives*. A typical geometric model base includes primitives such as a point, line, plane, ellipsoid, parallelepiped, cone, and cylinder. The first artificial potential field (16) requires analytic equations for the description of obstacles. For primitives such as a *parallelepiped, finite cylinder*, and *cone*, we have developed analytic equations representing envelopes which best approximate the primitives' shapes. The surface, termed an *n-ellipsoid*, is represented by the equation:

$$(\frac{x}{a})^{2n} + (\frac{y}{b})^{2n} + (\frac{z}{c})^{2n} = 1; \tag{21}$$

and tends to a parallelepiped of dimensions (a,b,c) as n tends to infinity. A good approximation is obtained with $n = 4$, as shown in Figure 2.

A cylinder of elliptical cross section (a, b) and of length $2c$ can be approximated by the so-called *n-cylinder* equation:

$$(\frac{x}{a})^2 + (\frac{y}{b})^2 + (\frac{z}{c})^{2n} = 1. \tag{22}$$

The analytic description of primitives is not necessary for the artificial potential field (17), since the continuity and differentiablity requirement is on the shortest distance to the obstacle. The primitives above, and more generally all convex

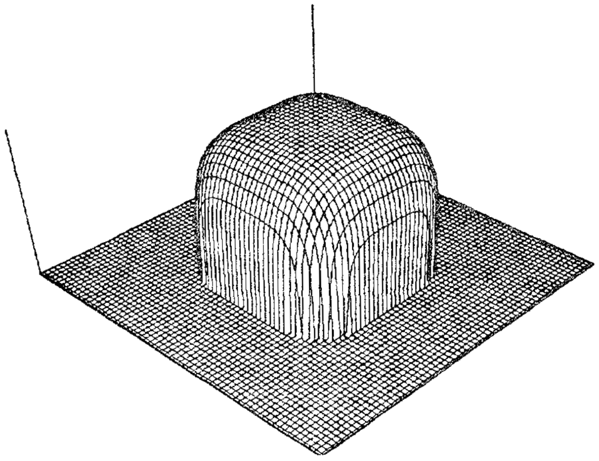

Figure 2. *An n-ellipsoid with n=4*

primitives, comply with this requirement.

Determining the orthogonal distance to an n-ellipsoid or to an n-cylinder requires the solution of a complicated system of equations. To avoid this costly computation, a variational procedure for the distance evaluation has been developed. The distance expressions for other primitives are presented in Appendices I through III.

Robot Obstacle Avoidance

An obstacle $\mathcal{O}_i$ is described by a set of primitives $\{\mathcal{P}_p\}$. The superposition property (additivity) of potential fields enables the control of a given point of the manipulator with respect to this obstacle by using the sum of the relevant gradients:

$$F^*_{\mathcal{O}_i,psp} = \sum_p F^*_{(\mathcal{P}_p,psp)}. \tag{23}$$

Control of this point for several obstacles is obtained using:

$$F^*_{psp} = \sum_i F^*_{(\mathcal{O}_i,psp)}. \tag{24}$$

It is also feasible to have different points on the manipulator controlled with respect to different obstacles. The resulting joint force vector is given by:

$$\Gamma_{obstacles} = \sum_j J^T_{psp_j}(q)\Lambda(x)F^*_{psp_j}. \tag{25}$$

Specifying an adequate number of PSPs enables the protection of all of the manipulator's parts. An example of a dynamic simulation for a redundant 4 *dof* manipulator operating in the plane[7] is shown in the display of Figure 3. The artificial potential field approach can be extended to *moving obstacles*, since stability of the mechanism persists with a continuously time-varying potential field.

The manipulator obstacle avoidance problem has been formulated in terms of *collision avoidance of links*, rather than

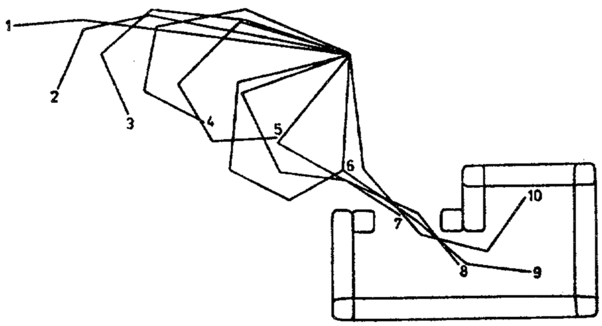

Figure 3. *Displacement of a 4 dof Manipulator Inside an Enclosure*

points. Link collision avoidance is achieved by continuously controlling the link's closest point to the obstacle. At most, n PSP's then have to be considered. Additional links can be artificially intoduced or the length of the last link can be extended to account for the manipulator tool or load. In an articulated chain, a link can be represented as the line segment defined by the Cartesian positions of its two neighboring joints. In a frame of reference R, a point $m(x,y,z)$ of the link bounded by $m_1(x_1,y_1,z_1)$ and $m_2(x_1,y_1,z_1)$ is described by the parametric equations:

$$\begin{aligned} x &= x_1 + \lambda(x_2 - x_1); \\ y &= y_1 + \lambda(y_2 - y_1); \\ z &= z_1 + \lambda(z_2 - z_1). \end{aligned} \tag{26}$$

The problem of obtaining the link's shortest distance to a parallelepiped can be reduced to that of finding the link's closest point to a vertex, edge, or face. The analytic expressions of the link's closest point, the distance, and its partial derivatives are given in Appendix I. In Appendices II and III these expressions are given for a cylinder and a cone, respectively.

Joint Limit Avoidance

The potential field approach can be used to satisfy the manipulator internal joint constraints. Let $\underline{q}_i$ and $\overline{q}_i$ be respectively the minimal and maximal bounds of the i^{th} joint coordinate q_i. q_i can be kept within these boundaries by creating barriers of potential at each of the hyperplanes $(q_i = \underline{q}_i)$ and $(q_i = \overline{q}_i)$. The corresponding joint forces are:

$$\Gamma_{\underline{q}_i} = \begin{cases} \eta(\frac{1}{\underline{\rho}_i} - \frac{1}{\underline{\rho}_{i(0)}})\frac{1}{\underline{\rho}_i^2}, & \text{if } \underline{\rho}_i \leq \underline{\rho}_{i(0)}; \\ 0, & \text{if } \underline{\rho}_i > \underline{\rho}_{i(0)}; \end{cases} \tag{27}$$

and:

$$\Gamma_{\overline{q}_i} = \begin{cases} -\eta(\frac{1}{\overline{\rho}_i} - \frac{1}{\overline{\rho}_{i(0)}})\frac{1}{\overline{\rho}_i^2}, & \text{if } \overline{\rho}_i \leq \overline{\rho}_{i(0)}; \\ 0, & \text{if } \overline{\rho}_i > \overline{\rho}_{i(0)}; \end{cases} \tag{28}$$

where $\underline{\rho}_{i(0)}$ and $\overline{\rho}_{i(0)}$ represent the distance limit of the potential field influence. The distances $\underline{\rho}_i$ and $\overline{\rho}_i$ are defined by:

$$\begin{aligned} \underline{\rho}_i &= q_i - \underline{q}_i; \\ \overline{\rho}_i &= \overline{q}_i - q_i \end{aligned} \tag{29}$$

Level of Competence

The complexity of tasks that can be achieved with this collision avoidance approach is limited. In a cluttered environment, local minima can occur in the resultant potential field. This can lead to a stable positioning of the robot before reaching its goal. While local procedures can be designed to exit from such configurations, limitations for complex tasks will remain. This is because the approach has a *local* perspective of the robot environment.

Nevertheless, the resulting potential field does provide the global information necessary, and a collision-free path, if attainable, can be found by linking the absolute minima of the potential. Linking these minima requires, however, a computationally expensive exploration of the potential field. This goes beyond the real-time control we are concerned with here, but can be considered as an integrated part of higher level control. Work on high level collision-free path planning based on the potential field concept has been investigated by C. Buckley[4].

Real-Time Implementation

Finally, the global control system integrating the potential field concept with the operational space approach has the following structure:

$$\Gamma = \Gamma_{motion} + \Gamma_{obstacles} + \Gamma_{joint-limit}; \tag{30}$$

where Γ_{motion} can be developed [Khatib 1983] in the form:

$$\Gamma_{motion} = J^T(q)\Lambda(q)F^*_{x_d} + \tilde{B}(q)[\dot{q}\dot{q}] + \tilde{C}(q)[\dot{q}^2] - g(q); \tag{31}$$

the matrices $\tilde{B}(q)$, $\tilde{C}(q)$, and $g(q)$ of the Coriolis, centrifugal and gravity forces have the dimensions $n \times n(n-1)/2$, $n \times n$, and $n \times 1$, respectively. $[\dot{q}\dot{q}]$ and $[\dot{q}^2]$ are defined by:

$$\begin{aligned} [\dot{q}\dot{q}] &= [\dot{q}_1\dot{q}_2 \ \dot{q}_1\dot{q}_3 \ldots \dot{q}_{n-1} \ \dot{q}_n]^T; \\ [\dot{q}^2] &= [\dot{q}_1^2 \ \dot{q}_2^2 \ldots \dot{q}_n^2]^T. \end{aligned} \tag{32}$$

In this control structure, dynamic decoupling of the end-effector is obtained using the end-effector dynamic parameters (EEDP) $\Lambda(q)$, $\tilde{B}(q)$, $\tilde{C}(q)$ and $g(q)$, which are configuration dependent. In real time, these parameters can be computed at a lower rate than that of the servo control. In addition, the integration of an operational position and velocity estimator allows a reduction in the rate of end-effector position computation, which involves evaluations of the manipulator geometric model. This leads to a two-level control system architecture[10]:

- a low rate *parameter evaluation level*: updating the EEDP, the Jacobian matrix and the geometric model;
- a high rate *servo control level*: computing the command vector using the estimator and the updated dynamic parameters.

The control system architecture is shown in Figure 4 where np represents the number of PSP's. The Jacobian matrices $J^T_{psp_j}$ have common factors with the end-effector Jacobian matrix J^T. Thus, their evaluation does not require significant additional

Applications

An experimental manipulator programming system COSMOS (Control in Operational Space of a Manipulator-with-Obstacles System), has been designed at the Stanford Artificial Intelligence

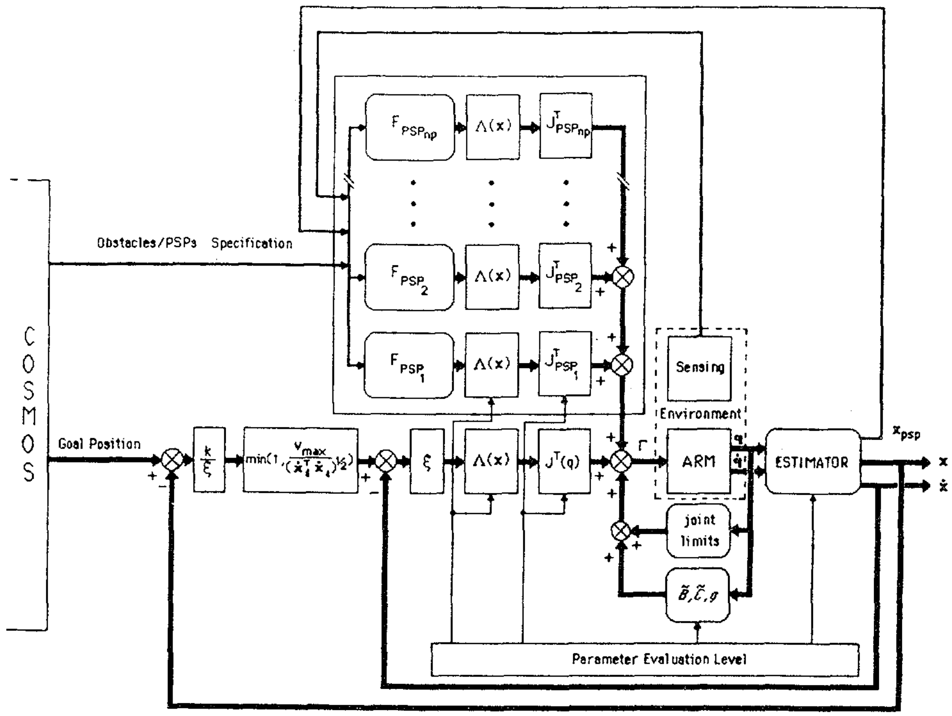

Figure 4. *Operational Space Control System Architecture*

Laboratory for implementation of the operational space control approach for the Unimation PUMA 560 arms. For these manipulators, the ability to control joint torque is considerably restricted by the nonlinearities and friction inherent in their joint actuator/transmission systems. Therefore, the centrifugal and Coriolis forces have been ignored in the PUMA end-effector dynamic model.

The COSMOS system is implemented on a PDP 11/45 interfaced to a PUMA 560. The PDP 11/23 and VAL are disconnected, and only the joint microprocessors in the PUMA controller are used for motor current control. The PUMA is equipped with a six degree of freedom force wrist that is interfaced to the PDP 11/45 via an A/D converter. The PUMA is also interfaced to a Machine Intelligence Corporation vision module.

In the current COSMOS implementation, the rate of the servo control level is 125 Hz while the parameter evaluation level runs at 40 Hz. With the new multiprocessor implementation (PDP 11/45 and PDP 11/60), COSMOS is expected to achieve a dynamic and kinematic update rate of 100 Hz and a servo control rate of 300 Hz.

We have demonstrated real-time end-effector motions both free and constrained, with the COSMOS system. These include contact, slide, insertion, and compliance operations, as well as real-time collision avoidance with links and moving obstacles[3].

Summary and Discussion

We have described the formulation and the implementation of a real-time obstacle avoidance approach based on the artificial potential field concept, using analytic primitives for obstacle geometric modelling. In this approach, collision avoidance, generally treated as high level planning, has been demonstrated to be an effective component of low level real-time control. Further, we have briefly presented our operational space formulation of manipulator control which provides the basis for this obstacle avoidance approach, and have described the two-level architecture designed to increase the real-time performance of the control system.

The integration of this low level control approach with a high level planning system seems to be one of the more promising solutions to the obstacle avoidance problem in robot control. With this approach, the problem may be treated in two stages:

- at high level control, generating a global strategy for the manipulator's path in terms of intermediate goals (rather than finding an accurate collision-free path);
- at the low level, producing the appropriate commands to attain each of these goals, taking into account the detailed geometry and motion of manipulator and obstacle, and making use of real-time obstacle sensing (low level vision and proximity sensors).

By extending low level control capabilities and reducing the high level path planning burden, the integration of this collision avoidance approach into a multi-level robot control structure will improve the real-time performance of the overall robot control system. Potential applications of this control approach include moving obstacle avoidance, grasping collision avoidance, and obstacle avoidance problems involving multi-manipulators with multi-fingered hands.

Appendix I: Link Distance to a Parallelepiped

The axes of the frame of reference R are chosen to be the parallelepiped axes of symmetry. l is the link's length and $(\cdot)$ designates the dot product.

Distance to a Vertex

The closest point m of the line (26) to the vertex v is such that:

$$\lambda = \frac{(vm_1)\cdot(m_1m_2)}{l^2}. \qquad (A1-1)$$

The link's closest point m is identical to m_1 if $\lambda \leq 0$; it is identical to m_2 if $\lambda \geq 1$ and it is given by (26) otherwise. The shortest distance is therfore:

$$\rho = \begin{cases} [\rho_1^2 - \lambda^2 l^2]^{1/2}, & \text{if } 0 \leq \lambda \leq 1; \\ \rho_1, & \text{if } \lambda < 0; \\ \rho_2, & \text{if } \lambda > 1; \end{cases} \qquad (A1-2)$$

where ρ_1 and ρ_2 are the distance to the vertex from m_1 and m_2, respectively. The distance partial derivatives are:

$$\frac{\partial\rho}{\partial\boldsymbol{x}} = [\frac{x}{\rho}\ \frac{y}{\rho}\ \frac{z}{\rho}]^T. \qquad (A1-3)$$

Distance to an Edge

By a projection in the plane perpendicular to the considered edge (xoy, yoz, or zox), this problem can be reduced to that of finding the distance to a vertex in the plane. This leads to expressions similar to those of (A1-1)-(A1-3) with a zero partial derivative of the distance *w.r.t.* the axis parallel to the edge.

Distance to a Face

In this case, the distance can be directly obtained by comparing the absolute values of the coordinates of m_1 and m_2 along the axis perpendicular to the face. The partial derivative vector is identical to the unit normal vector of this face.

Appendix II: Link Distance to a Cylinder

The frame of reference R is chosen such that its z-axis is the cylinder axis of symmetry and its origin is the cylinder center of mass. r and h designate, respectively, the cylinder radius and height.

Distance to the Circular Surface

The closest point of the link (27) to the circular surface of the cylinder can be deduced from the distance to a vertex considered in the xoy plane and by allowing for the radius r.

Distance to the Circular Edges

The closest distance to the cylinder circular edge can be obtained from that of the circular surface by taking into account the relative z-coordinate of m to the circular edge *i.e.* $(z + h/2)$ for the base and $(z - h/2)$ for the top. The distance partial derivative vector results from the torus equation:

$$[x^2 + y^2 + (z \pm h/2)^2 - r^2 - \rho^2]^2 = 4r^2[\rho^2 - (z \pm h/2)^2]. \qquad (A2-1)$$

This vector is:

$$\frac{\partial\rho}{\partial\boldsymbol{x}} = [\varsigma\frac{x}{\rho}\ \varsigma\frac{y}{\rho}\ \frac{z \pm h/2}{\rho}]^T; \qquad (A2-2)$$

with:

$$\varsigma = \frac{x^2 + y^2 + (z \pm h/2)^2 - r^2 - \rho^2}{x^2 + y^2 + (z \pm h/2)^2 + r^2 - \rho^2}. \qquad (A2-3)$$

The distance to the planar surfaces is straightforward and can be simply obtained as in Appendix I.

Appendix III: Link Distance to a Cone

In this case, the frame of reference R is chosen such that its z-axis is the cone axis of symmetry and its origin is the center of the cone circular base. r, h, and β represent, respectively, the cone base radius, height and half angle.

Distance to the Cone-Shaped Surface

The problem of locating $m(x, y, z)$ is identical to that for the cylinder case. The distance can be written as:

$$\rho = z\, sin(\beta) + (\sqrt{x^2 + y^2} - r)cos(\beta). \qquad (A3-1)$$

The partial derivatives come from the equation:

$$x^2 + y^2 = r_z^2; \qquad (A3-2)$$

where:

$$r_z = tan(\beta)[h + \rho\, sin(\beta) - z]. \qquad (A3-3)$$

They are:

$$\frac{\partial\rho}{\partial\boldsymbol{x}} = [\frac{x}{r_z\, tan(\beta)}\ \frac{y}{r_z\, tan(\beta)}\ \frac{1}{sin(\beta)}]^T. \qquad (A3-4)$$

The problem of the distance to the cone circular edge is identical to that of the cylinder circular edge in Appendix II. The distance to the cone vertex is solved as in Appendix I.

Acknowledgments

Tom Binford and Bernie Roth have encouraged and given support for the continuation of this research at Stanford University. Many thanks to Harlyn Baker, Peter Blicher, and Jeff Kerr for their help in the preparation of this manuscript.

References

[1] Brooks, R. 1983. Solving the Find-Path Problem by Good Representation of Free Space. IEEE Systems, Man and Cybernetics. SMC-13:190-197.

[2] Brooks, R. 1984 (Aug. 20-23). Aspects of Mobile Robot Visual Map Making. Second International Symposium of Robotics Research. Kyoto, Japan.

[3] Brooks, K. and Khatib, O. 1984 (June). Robotics in Three Acts (Film). Stanford University.

[4] Buckley, C. 1985. The Application of Continuum Methods to Path Planning. Ph.D. Thesis (in progress). Stanford University. Department of Mechanical Engineering.

[5] Chatila, R. 1981. Système de Navigation pour un Robot Mobile Autonome: Modélisation et Processus Décisionnels. Thése de Docteur-Ingénieur. Université Paul Sabatier. Toulouse, France.

[6] Khatib, O., Llibre, M. and Mampey, R. 1978. Fonction Decision-Commande d'un Robot Manipulateur. Rapport No. 2/7156. Toulouse, France. DERA/CERT.

[7] Khatib, O. and Le Maitre, J.F. 1978 (Sep. 12-15). Dynamic Control of Manipulators Operating in a Complex Environment. 3rd CISM-IFToMM. Udine, Italy.

[8] Khatib, O. 1980. Commande Dynamique dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Obstacles. Thése de Docteur-Ingénieur. École Nationale Supérieure de l'Aéronautique et de l'Espace (ENSAE). Toulouse, France.

[9] Khatib, O. 1983 (Sep 15-20). Dynamic Control of Manipulators in Operational Space. Sixth CISM-IFToMM Congress on Theory of Machines and Mechanisms. New Delhi, India.

[10] Khatib, O. 1984(Oct. 13). Real-Time Control of Manipulators in Operational Space. 28th Annual Stanford Conference of the American Society for Quality Control. Stanford.

[11] Lozano-Perez, T. 1980 Spatial Planning: A Configuration Space Approach. AI Memo 605. Cambridge, Mass. MIT Artificial Intelligence Laboratory.

[12] Moravec, H.P. 1980. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Ph.D. Thesis. Stanford University. Artificial Intelligence Laboratory.