

An Integrated Architecture for Robot Motion Planning and Control in the Presence of Obstacles With Unknown Trajectories

Rob Spence and Seth Hutchinson

Abstract—We present an integrated architecture for real-time planning and control of robot motions, for a robot operating in the presence of moving obstacles whose trajectories are not known a priori. The architecture comprises three control loops: an inner loop to linearize the robot dynamics, and two outer loops to implement the attractive and repulsive forces used by an artificial potential field motion planning algorithm. From a control theory perspective, our approach is unique in that the outer control loops are used to effect both desirable transient response and collision avoidance. From a motion planning perspective, our approach is unique in that the dynamic characteristics of both the robot and the moving obstacles are considered. Several simulations are presented that demonstrate the effectiveness of the planner/controller combination.

I. INTRODUCTION

AN autonomous robot must be able to maneuver effectively in its environment, achieving its goals while avoiding collisions with obstacles and/or other robots. This problem has been addressed both by researchers in motion planning, who typically focus on the trajectory *generation* problem, and by researchers in control theory, who typically focus on the trajectory *tracking* problem. In this paper, we present an integrated architecture for motion planning and control in the presence of moving obstacles with unknown trajectories. Our system uses a variation of the well-known potential fields planning method (first introduced by Khatib [13], [12]), combined with a nonlinear control structure. The result is a combination planner/controller that simultaneously satisfies motion planning goals (such as avoiding collisions with obstacles), and control goals (such as effecting a desired transient response).

Our planner/controller architecture, shown in Fig. 1, comprises three control loops. An inner-loop is used to linearize the robot dynamics, and two outer control loops are used to implement the artificial potential field algorithm. The two outer control loops correspond to an attractive force (which attracts the robot to its goal configuration), and to repulsive forces (which repel the robot from the moving obstacles).

The inner-loop/outer-loop control structure has been used by a number of researchers in the past (see, for example, [1], [4], [17], [31], [32]). Likewise, many researchers have used

Manuscript received September 12, 1992; revised February 12, 1994.

The authors are with The Beckman Institute for Advanced Science and Technology Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

IEEE Log Number 9404977.

the artificial potential field method for motion planning (see, for example, [3], [12], [14], [15], [23]). From a control theory perspective, our approach is unique in that the outer control loops are used to effect both desirable transient response and collision avoidance. From a motion planning perspective, our approach is unique in that the dynamic performance of both the robot and the moving obstacles are considered, unlike the artificial potential field planners cited above, which take into account only the instantaneous relative positions of the robot and the obstacles, and not relative velocities of the robot and the moving obstacles (e.g., [12]).

The remainder of the paper is organized as follows. In Section II, we provide an overview of research related to the problem of motion planning in the presence of obstacles moving with unknown trajectories. In Section III, we present a brief review of the artificial potential field method of motion planning. Section IV contains a detailed description of our system architecture, including design criteria used to construct each of the outer control loops. In Section V, we address the stability of our system and provide a proof that our system is asymptotically stable in the absence of obstacles. This stability proof considers the case when the model used by the inner-loop controller differs from the nominal model. In Section VI, system performance is evaluated in three scenarios: polygons moving in the plane; a planar articulated robot; and a Stanford arm and a planar robot arm sharing a workspace. Finally, in Section VII we summarize the contributions of the paper.

II. RELATED RESEARCH

The amount of research dedicated to on-line robot navigation in the presence of moving obstacles with unknown trajectories has been surprisingly scarce. To date, the applicable research has focused almost entirely on the problem of estimating or predicting the trajectories of the obstacles. These estimated/predicted obstacle trajectories are used by a motion planner to modify the robot trajectory to avoid collisions with the obstacles. Since it is assumed that the planner has no knowledge of the obstacle trajectories before the motion begins, it is necessary for both the analysis of obstacle trajectories and the revision of the motion plan to occur in real time.

Wang and Tsai [37] use a least-mean-square-error (LMSE) classification scheme to determine the trajectories of moving obstacles. The robot is modelled as moving in hallway

that contains both moving and stationary obstacles, with the trajectories of the moving obstacles constrained to be along linear paths. After the LMSE algorithm determines the obstacle trajectories, a method introduced in [36] is used to adjust the robot velocity to create a collision-free path. This plan is modified every cycle as new information is detected.

Similar methods are proposed by Zhu [38] and Kehtarnavaz and Li [11], each of which use statistical methods to estimate obstacle locations. Zhu applies a hidden Markov stochastic model that is updated periodically throughout the motion. Kehtarnavaz and Li fit the obstacle motion to an auto-regressive model. These estimation techniques are reasonably complex (from a computational point of view), and to date have been implemented only for 2-dimensional polygonal environments.

De La Madrid and Gini [8] have implemented a path tracking system that adjusts a nominal robot trajectory to avoid dynamic obstacles. This adjustment is performed under the constraints that the adjusted path not deviate by more than some threshold from the nominal path, and that the arrival time of the robot at the goal position not deviate by more than some threshold from the target arrival time. In their approach, the robot is modelled as a point and the moving obstacles as discs. Once a collision is predicted, the planner determines new points where a collision will not occur, and attempts to create a path through these points. Both robot and obstacle trajectories are constrained to consist only of piecewise linear segments, and during each of these segments the robot and obstacles move at constant velocity.

The system developed by Steel, [33] creates a repulsive force based on the vector from the robot to the point of closest approach to a moving obstacle. This repulsive force is used to adjust the trajectory of the robot in both space and time. This approach assumes that the obstacle motion will be along a straight line path. This work extends earlier work by Steel, and Starr [34].

Kyriakopoulos and Saridis [18], [19] present a system that adjusts the robot velocity to avoid unexpected moving obstacles using an optimal control technique. This approach is unique in that it considers robot dynamics (for a mobile robot moving in the plane) in determining the path.

All of these methods have been implemented only for mobile robots moving in the plane, with restrictive assumptions about obstacles' shape and/or trajectory. Except for determining the point of intersection between moving obstacles and the robot, no technique has incorporated information about obstacle velocity into the planning process. Finally, with the exception of [18], [19], none of these techniques consider robot dynamics. In fact, the systems of [8], [11] permit instantaneous changes in velocity.

In addition to the research listed above, there has also been work done in the robot motion planning community for the case where the trajectories of the moving obstacles are known a priori, or where there are multiple robots (which is similar in many respects to the case of moving-obstacles). In this case, motion planning can be performed in the joint configuration space of the robot and moving obstacles (see, e.g., [2], [28]), or by decoupling the planning into a phase where each moving object is treated individually followed by a phase in which the

individual paths are coordinated (see, e.g., [24], [16], [22], [26], [29], [35]). The inherent computational complexity of the problem of motion planning in the presence of moving obstacles with known trajectories has been addressed in [27].

III. POTENTIAL FIELD PLANNING

In the potential field approach to motion planning, the robot is treated as a particle in configuration space, under the influence of an artificially created potential field. The value of the potential field at a particular configuration will depend upon the robot's goal position and its distance to obstacles located in the workspace. The potential function originally proposed by Khatib [12] is given by:

$$U_{\text{total}}(\mathbf{q}) = U_{\text{att}}(\mathbf{q}) + U_{\text{rep}}(\mathbf{q}) \quad (1)$$

where \mathbf{q} is the configuration of the robot, U_{att} is the *attractive potential* and U_{rep} is the *repulsive potential*.

A standard form for the attractive potential is:

$$U_{\text{att}}(\mathbf{q}) = \frac{1}{2} \zeta \rho_{\text{goal}}^2(\mathbf{q}) \quad (2)$$

where $\rho_{\text{goal}}(\mathbf{q})$ is the metric distance from the current robot configuration to the goal configuration, and ζ is a scaling constant that specifies the strength of the attractive potential.

A standard form for the repulsive potential is:

$$U_{\text{rep}}(\mathbf{q}) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0} \right)^2 & \text{if } \rho(\mathbf{q}) \leq \rho_0 \\ 0 & \text{if } \rho(\mathbf{q}) > \rho_0 \end{cases} \quad (3)$$

where $\rho(\mathbf{q})$ is the minimum distance from the current robot configuration to a configuration in which the robot contacts an obstacle, and η is a scaling constant that specifies the strength of the repulsive field. The constant ρ_0 represents a "boundary distance," beyond which the repulsive potential field does not affect the robot.

In order to use gradient descent techniques for motion planning, the potential functions are converted into corresponding force vectors. The attractive force is given by:

$$\begin{aligned} \vec{F}_{\text{att}}(\mathbf{q}) &= -\vec{\nabla} U_{\text{att}}(\mathbf{q}) \\ &= -\zeta(\mathbf{q} - \mathbf{q}_{\text{goal}}) \end{aligned} \quad (4)$$

The repulsive force is given by:

$$\begin{aligned} \vec{F}_{\text{rep}}(\mathbf{q}) &= -\vec{\nabla} U_{\text{rep}}(\mathbf{q}) \\ &= \begin{cases} \eta \left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(\mathbf{q})} \vec{\nabla} \rho(\mathbf{q}) & \text{if } \rho(\mathbf{q}) \leq \rho_0 \\ \mathbf{0} & \text{if } \rho(\mathbf{q}) > \rho_0 \end{cases} \end{aligned} \quad (5)$$

where $\vec{\nabla} \rho(\mathbf{q})$ is a unit vector in the direction of the gradient of U_{rep} .

The primary problem facing a potential field planner is the presence of local minima in the potential field. These local minima result from the interaction between the repulsive forces due to obstacles and the attractive force due to the goal. A number of approaches have been proposed to deal with this problem, including using randomized motions [3], and carefully constructing the potential field to minimize the number of local minima that occur [14].

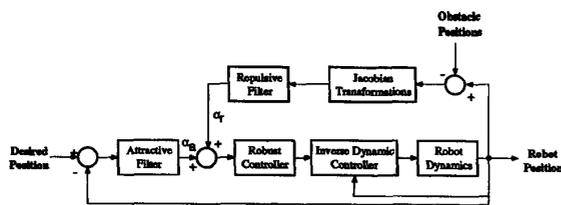


Fig. 1. Planner/controller architecture.

A second difficulty that confronts potential field planners is the calculation of distances between the robot and obstacles, particularly when the robot and/or obstacles are modelled by curved surfaces. This difficulty can be reduced by restricting the planner to compute potentials only at a fixed number of *control points* [20]. The primary disadvantage to this approach is that, while the control points themselves are guaranteed to avoid collisions, no such guarantee can be made for the other points on the robot. However, the possibility of such collisions can be reduced by increasing the number of control points on the robot, or by allowing “floating control points” (i.e. control points that are allowed to move to the point on the robot boundary that is nearest an obstacle). Furthermore, the control point approach can be implemented in real-time using special purpose distance sensors mounted on the robot, for example the sensitive skin described in [5], [6].

The real-time capabilities of the potential field planning method make it particularly attractive for our problem: avoiding moving obstacles with unknown trajectories. However, the standard potential field algorithm is limited in that it does not take advantage of any dynamic information (such as obstacle velocity), nor does it explicitly consider the dynamic response of the robot. In fact, at each iteration, the standard potential field planning algorithm considers only the instantaneous positions of the robot and of obstacles. In the next section, we describe a new approach, in which a potential field planner is directly integrated into a robot control architecture. The resulting system is able to exploit the capabilities of inverse dynamics control, and to use digital filtering techniques to improve the robot’s response to moving obstacles.

IV. THE INTEGRATED PLANNER/CONTROLLER ARCHITECTURE

As described above, our planner/controller comprises three control loops: an inner loop (to linearize the robot dynamics), and two outer loops (one loop to implement the attractive forces, and one to implement the repulsive forces). In this section, we describe each of these three control loops. Because much work has been done in the area of inverse dynamics control, our description of the inner-loop controller is fairly brief compared to the description of the outer control loops. A block diagram of our planner/controller system is shown in Fig. 1.

A. Inverse Dynamics Control

Inverse dynamics control is achieved by exploiting a model of the robot dynamics to linearize the robot system. This is typically accomplished by using an inner-loop/outer-loop

control architecture, in which the inner loop is used to linearize the system, and the outer loop is used to control the resulting linearized system [30]. An introduction to inverse dynamics can be found in [17] and a more mathematical development in [31]. Spong and Vidyasagar [32] added robustness to this scheme by compensating for uncertainties in the dynamic model using a factorization approach. Other researchers have investigated the effects of various types of controllers upon the linearized robot, including PI [4] and PID [1] controllers.

The particular controller that we use is a variation of the robust controller described by Spong and Vidyasagar [30]. The main distinction of our controller is the outer-loop control law. While [30] uses state-feedback for outer-loop control, our outer-loop control is comprised of the attractive and repulsive control loops, each of which will be discussed in detail in the following sections.

The stability of the resulting controller depends largely on the stability of the attractive control loop. In Section V, we address the stability of the attractive loop with the repulsive loop disconnected (this corresponds to a situation in which no obstacles are present). We present a proof that this system is asymptotically stable along solution trajectories of the system, even when the robot dynamic model is not exactly known. With this in mind, in the remainder of the paper we will assume that the robot model is exact, and therefore the robot dynamics for any particular joint can be treated as a double-integrator system, with a transfer function given by $G_p(s) = \frac{1}{s^2}$.

B. The Attractive Loop

The attractive force, $\vec{F}_{att}(\mathbf{q})$ is implemented by the attractive control loop, the main functional component of which is the *attractive filter*. In a traditional potential field planner, the attractive force, given by (4), is the distance from the current configuration to the goal, multiplied by a constant gain. The primary disadvantage to this formulation is that it does not include any of the dynamic effects of the robot. To remedy this, we have included a digital filter in the forward path. This filter allows the system response to be a function, not only of instantaneous distance to the goal, but also of the past trajectory of the robot (and its time derivatives).

The attractive filter must be designed to be stable and proper, but this restriction does not significantly reduce the design flexibility. In particular, improper compensators, such as PD controllers, are typically implemented using either digital computers or analog opamp networks. Each of these has specific bandwidth limitations, which implies that such a controller will have high-order dynamics that are usually ignored in the control design (since they have a negligible effect on the performance). For example, a PD controller will have a pole in the left half-plane. Typically, this pole is over 10 times the magnitude of the zero of the controller, and therefore has little effect on the performance of the closed-loop system. However, the pole will serve to make the PD controller proper and stable [9].

We have chosen to implement the attractive control loop in joint space rather than in the Cartesian, or task, space. The main reason for this choice is that a task space formulation

requires the inclusion of the non-linear forward kinematic equations of the robot in the feedback path. While it is possible to stabilize the resulting closed-loop system, the robot will occasionally undergo severe oscillations in approaching the goal [7].

We use three design criteria for the attractive filter:

1. The attractive loop must be asymptotically stable, with poles distant from the $j\omega$ -axis. The latter condition is necessary since uncertainties can drive the poles in any direction, and if they are not far into the left half-plane, the uncertainties could drive the system unstable.
2. As is typical for robotics systems, the transient response should be either critically damped or overdamped.
3. The signals throughout the forward path must not exceed a maximum bound. For example, there will be a maximum possible voltage that the joint actuators will be able to source before they saturate. If we exceed this voltage and the actuators saturate, the nominally linear system becomes nonlinear and we can no longer guarantee that it will remain stable.

These criteria represent standard problems in linear control design. In fact, the above criteria can be met with a simple first order controller, such as a lead or lag filter.

C. The Repulsive Loop

In the traditional potential field planning algorithm, repulsive forces are a function only of the instantaneous distance from the current robot configuration to the nearest configuration in which a collision would occur. The disadvantage to this approach is that the planner cannot exploit any knowledge of the dynamic characteristics of obstacle motion. In our system, dynamic properties of obstacle motion are incorporated into the planning/control process by using a digital filter to modify the computed repulsive forces. We refer to this filter as the *repulsive filter*. The frequency response of this filter determines how the dynamic characteristics of obstacle motion will affect system response.

When designing the repulsive filter, two issues arise.

- 1) If the repulsive forces are implemented in discrete time, it is possible that collisions may occur between sampling instants.
- 2) There will always be situations in which the robot will be unable to prevent a collision. Hogan [10] notes that this problem can occur with *any* control structure.

There are basically two design parameters for the repulsive filter: the repulsive force gains (as a function of frequency) and the sampling rate of the digital system. The gains of the repulsive forces should be large enough to ensure that the robot does not collide with obstacles, and the sampling rate should be chosen so that the robot will not collide with obstacles between sampling instants. We now examine the construction of the repulsive forces.

Construction of the Repulsive Forces: The signal constructed by the repulsive loop is an n -dimensional vector that represents a force (or, for revolute joints, a torque) upon the joints of the robot. As described in Section III, our algorithm computes repulsive forces only for a discrete set of control

points on the robot. We define the repulsive force on each control point by determining the nearest obstacle to that point, and constructing a force vector from the obstacle to the control point with a magnitude inversely proportional to the distance to the obstacle (the precise formulation is given below in (7)). The obstacle region can be defined using surface polynomials or by using control points. In the latter case, for each robot control point the planner determines the nearest obstacle control point. The vector from that obstacle point to the robot control point is used to determine the force magnitude and direction.

For an articulated robot, once the forces upon the robot control points have been determined, it remains to convert these forces to torques upon the joints of the robot. The relationship between the end-effector forces and the joint torques for an n -link robot is given by

$$\tau = J^T F_e \quad (6)$$

where τ is the vector of joint torques, J is the manipulator Jacobian, and F_e is a vector of forces applied to the end-effector (consisting of three forces and three torques). In general, a $6 \times i$ Jacobian matrix, J_{ip} , relates forces exerted on a point, p on link i of the robot to torques on the first i joints of the robot (since forces exerted on link i will not create a torque on any joints beyond that link, assuming friction and gravity have already been taken into account).

Given a collection of control points along the entire length of the manipulator, we create the repulsive torque vector as follows:

- F_i^p is the repulsive force exerted on control point p in link i , expressed in terms of the inertial frame. In our experiments, we have defined the repulsive forces by:

$$F_i^p = \frac{X_i^p - X_{OBi}^p}{\|X_i^p - X_{OBi}^p\|^2} \quad (7)$$

where X_i^p represents the coordinates of the p^{th} control point in link i , and X_{OBi}^p represents the coordinates of the obstacle point nearest to X_i^p , both expressed with respect to the world coordinate frame.

- Let J_{ip} be the $6 \times i$ Jacobian matrix such that

$$\tau_i = J_{ip}^T F_i^p \quad (8)$$

where τ_i is the vector of joint torques for joints $0, \dots, i-1$.

- $\bar{J}_{ip} = [\bar{J}_{ip} \quad \mathbf{0}]$ is the $6 \times n$ augmented Jacobian matrix where n is the total number of links.
- τ_{rep} , the vector of total repulsive torques on all joints of the robot, is

$$\tau_{\text{rep}} = \sum_{i=1}^n \sum_{p=1}^{P_i} \bar{J}_{ip}^T F_i^p \quad (9)$$

where P_i is the total number of repulsive control points on link i .

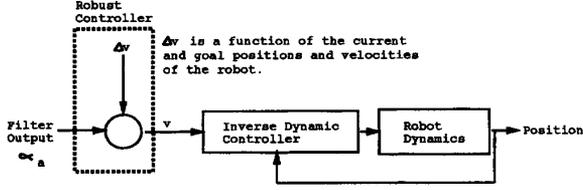


Fig. 2. Robust control structure.

Design of the Repulsive Filter: The goal of the repulsive filter is to provide information to the control system about obstacle velocity and position. This is accomplished using a frequency domain interpretation of obstacle motion. For example, repulsive forces that increase rapidly in magnitude indicate that either an obstacle is approaching the robot rapidly or that the robot is approaching an obstacle rapidly. The system can thus determine how quickly obstacles are moving in each direction, either in Cartesian or joint space.

In our system, we have implemented the repulsive filter as a lead filter. Filter parameters are chosen so that the frequency response is a linear for expected relative speeds of the obstacles and the robot. The low frequency cutoff prevents arbitrarily small repulsive forces for stationary obstacles. The high frequency cutoff reduces the effect of random noise on the system.

V. ROBUST STABILIZATION OF THE ATTRACTIVE LOOP

In this section we show the conditions under which the attractive loop can be proven to be asymptotically stable given direct interaction with the robot dynamics. Our proof is similar to that presented in [30]. The main distinction of our approach is the presence of a compensator (the attractive filter) in the forward path instead of full-state feedback. Robustness to plant uncertainties is guaranteed by introducing an additional control signal $\Delta \mathbf{v}$ that is summed with the output of the attractive filter to produce a signal that is presented to the inverse dynamics controller. This resulting system is shown in Fig. 2. The robust control $\Delta \mathbf{v}$ will be derived in the course of stability proof.

The robot dynamic model for an n -link articulated robot is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (10)$$

where $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ represents the combination of the Coriolis, centrifugal, and gravitational forces.

Since the above equation is nonlinear, we choose an inverse dynamics control law to cancel the nonlinear terms

$$\boldsymbol{\tau} = \hat{\mathbf{M}}(\mathbf{q})\mathbf{v} + \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) \quad (11)$$

where $\hat{\mathbf{M}}$ and $\hat{\mathbf{h}}$ are the approximations of \mathbf{M} and \mathbf{h} used by the inverse dynamics controller. We will show that we can stabilize the robot despite errors in these approximations.

The system that results from the combination of the nonlinear plant and the inverse dynamics control law is

$$\ddot{\mathbf{q}} = \mathbf{v} + \boldsymbol{\eta}(\mathbf{v}, \mathbf{q}, \dot{\mathbf{q}}) \quad (12)$$

where $\boldsymbol{\eta}$ is given by

$$\boldsymbol{\eta} = \mathbf{E}(\mathbf{q})\mathbf{v} + \mathbf{M}^{-1}\Delta \mathbf{h} \quad (13)$$

with $\mathbf{E} = \mathbf{M}^{-1}\hat{\mathbf{M}} - \mathbf{I}$, $\Delta \mathbf{h} = \hat{\mathbf{h}} - \mathbf{h}$.

The linearized plant, therefore, has the state-space model

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{B}(\mathbf{v} + \boldsymbol{\eta}) \quad (14)$$

$$\mathbf{q} = \mathbf{C}\mathbf{y} \quad (15)$$

where \mathbf{q} is the $n \times 1$ vector of joint variables, and

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I}_n \\ 0 & 0 \end{bmatrix} \quad (16)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{I}_n \end{bmatrix} \quad (17)$$

$$\mathbf{C} = [\mathbf{I}_n, 0] \quad (18)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \quad (19)$$

Our control task is to have the robot approach a set position and velocity. We define the error, \mathbf{e} , as the difference between the current state and the desired state vector.

$$\mathbf{e} = \mathbf{y} - \mathbf{y}^d = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} \mathbf{q}^d \\ \dot{\mathbf{q}}^d \end{bmatrix} \quad (20)$$

From this, we can rewrite our system in terms of the errors

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}(\mathbf{v} + \boldsymbol{\eta} - \ddot{\mathbf{q}}^d) \quad (21)$$

We can further simplify the system by noting that, for motion planning purposes, we are not interested in a nonzero steady-state acceleration (note that gravity has already been taken into account in the initial equations). As such, we can set $\ddot{\mathbf{q}}^d = 0$.

There is an attractive filter on each joint. We can represent all of these filters-together with the state-space model

$$\dot{\mathbf{z}} = \mathbf{A}_a\mathbf{z} + \mathbf{B}_a\mathbf{u} \quad (22)$$

$$\alpha_a = \mathbf{C}_a\mathbf{z} + \mathbf{D}_a\mathbf{u} \quad (23)$$

where $\mathbf{u} = \mathbf{q}^d - \mathbf{q}$ is the difference between the desired and current position (the negative of the error as we have defined it) and \mathbf{z} is the vector of state variables of the attractive filter. Since this model lumps together the attractive filters for all of the joints, it should be noted that both the input \mathbf{u} and the output α_a are vectors, with dimension equal to the number of joints of the robot.

The input to the attractive filter is zero in steady state, so the origin is an equilibrium position for the filter states. We will investigate system stability around this equilibrium point.

As shown in Fig. 2, we define the input \mathbf{v} to the nonlinear inner-loop controller as

$$\mathbf{v} = \alpha_a + \Delta \mathbf{v} \quad (24)$$

where $\Delta \mathbf{v}$ (which will be derived below) is an input to correct for errors in the dynamic model.

The filter and linearized robot equations become

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{A}\mathbf{e} + \mathbf{B}(\alpha_a + \Delta \mathbf{v} + \boldsymbol{\eta}) \\ &= \mathbf{A}\mathbf{e} + \mathbf{B}(\mathbf{C}_a\mathbf{z} + \mathbf{D}_a(\mathbf{q}^d - \mathbf{q}) + \Delta \mathbf{v} + \boldsymbol{\eta}) \\ &= \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{C}_a\mathbf{z} - \mathbf{B}\mathbf{D}_a\mathbf{C}\mathbf{e} + \mathbf{B}\Delta \mathbf{v} + \mathbf{B}\boldsymbol{\eta} \\ (\dot{\mathbf{z}}) &= \mathbf{A}_a\mathbf{z} + \mathbf{B}_a(\mathbf{q}^d - \mathbf{q}) \\ &= \mathbf{A}_a\mathbf{z} - \mathbf{B}_a\mathbf{C}\mathbf{e} \end{aligned}$$

We can combine these subsystems into an augmented state space representation by defining a new state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \mathbf{z} \end{bmatrix} \quad (25)$$

which gives us the state-space model

$$\dot{\mathbf{x}} = \bar{\mathbf{A}}\mathbf{x} + \bar{\mathbf{B}}(\eta + \Delta\mathbf{v}) \quad (26)$$

where

$$\bar{\mathbf{A}} = \begin{bmatrix} A - BD_cC & BC_c \\ -B_cC & A_c \end{bmatrix} \quad (27)$$

$$\bar{\mathbf{B}} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (28)$$

We can design our attractive filters so that $\bar{\mathbf{A}}$ is a stable matrix. This design can be accomplished by making sure that the control loops on each joint are stable (since there is no cross-coupling between joints). Each joint will have a double-integrator linked with an attractive filter. The attractive filter for each joint can be designed using any number of techniques from linear system theory. As long as the individual loops for each joint are stable, the entire system, given by (26), is stable.

Given that the nominal system (for $\eta = 0$) is stable, we will now show that we can guarantee stability for $\eta \neq 0$ (i.e. a system in which the model is not perfectly accurate). The control input $\Delta\mathbf{v}$ can be chosen by the designer to accomplish this. Before proceeding with the derivation of $\Delta\mathbf{v}$, we make the following assumptions about the magnitudes of the uncertainties.

1. $\|M^{-1}\hat{M} - I\| = \|E\| \leq \varepsilon < 1$ for some ε , for all $\mathbf{q} \in \mathcal{R}^n$.
2. $\|\Delta h\| \leq \phi(e, t)$ for a known function ϕ , bounded in t .

It has been shown in [30] that we can always find \hat{M} that will satisfy the first assumption, given known upper and lower bounds on M (such bounds on M always exist).

The process for choosing $\Delta\mathbf{v}$ is as follows:

- 1) Find a function $\rho(\mathbf{x}, t)$ such that

$$\|\Delta\mathbf{v}\| \leq \rho(\mathbf{x}, t) \quad (29)$$

$$\|\eta\| \leq \rho(\mathbf{x}, t) \quad (30)$$

From these restrictions and the two assumptions, we can derive a value for $\rho(\mathbf{x}, t)$ implicitly using

$$\begin{aligned} \|\eta\| &= \|E(\mathbf{q})\mathbf{v} + M^{-1}\Delta h\| \\ &\leq \epsilon\|\mathbf{v}\| + \bar{M}\phi(\mathbf{e}, t) \\ &\leq \epsilon\|\Delta\mathbf{v}\| + \epsilon\|\alpha_a\| + \bar{M}\phi(\mathbf{e}, t) \\ &\leq \epsilon\rho(\mathbf{x}, t) + \epsilon\|\alpha_a\| + \bar{M}\phi(\mathbf{e}, t) \end{aligned}$$

where \bar{M} is the upper bound on M^{-1} .

By setting the term on the right-hand side of the final inequality equal to $\rho(\mathbf{x}, t)$, we can solve for $\rho(\mathbf{x}, t)$ as

$$\rho(\mathbf{x}, t) = \frac{1}{1 - \epsilon} [\epsilon\|\alpha_a\| + \bar{M}\phi(\mathbf{e}, t)] \quad (31)$$

Note that α_a is a function of \mathbf{x} , specifically

$$\alpha_a = [-D_cC \ C_c]\mathbf{x} \quad (32)$$

- 2) Find a positive definite matrix P such that

$$\bar{\mathbf{A}}^T P + P\bar{\mathbf{A}} + Q = 0 \quad (33)$$

where P and Q are positive definite, symmetric matrices. Since $\bar{\mathbf{A}}$ is Hurwitz, this will always be possible.

- 3) The control $\Delta\mathbf{v}$ is then

$$\Delta\mathbf{v} = \begin{cases} -\rho(\mathbf{e}, t) \frac{\bar{\mathbf{B}}^T P \mathbf{e}}{\|\bar{\mathbf{B}}^T P \mathbf{e}\|} & \text{if } \|\bar{\mathbf{B}}^T P \mathbf{e}\| \neq 0 \\ 0 & \text{if } \|\bar{\mathbf{B}}^T P \mathbf{e}\| = 0 \end{cases} \quad (34)$$

where $\Delta\mathbf{v}$ has been constructed to satisfy (29).

We are now in a position to make some assessments of the stability of the system. Let

$$V(\mathbf{e}, \mathbf{z}) = \mathbf{x}^T P \mathbf{x} \quad (35)$$

where V is a function of the state variables that satisfies the following two criteria:

- 1) V has a unique minimum, which is achieved when the manipulator is at the goal configuration.
- 2) Along any trajectory of the system, the value of V never increases (i.e., $\dot{V} \leq 0$).

That the first criterion is satisfied follows from the fact that V is quadratic in the state variables and P is a positive definite matrix. Furthermore, the only minimum of V occurs when $\mathbf{x} = 0$. This is when the robot is at the goal and the filter states are zero.

For the second criterion, we take the time derivative of V

$$\begin{aligned} \dot{V} &= \dot{\mathbf{x}}^T P \mathbf{x} + \mathbf{x}^T P \dot{\mathbf{x}} \\ &= \mathbf{x}^T (\bar{\mathbf{A}}^T P + P\bar{\mathbf{A}}) \mathbf{x} + 2\mathbf{x}^T P \bar{\mathbf{B}} (\Delta\mathbf{v} + \eta) \\ &= -\mathbf{x}^T Q \mathbf{x} + 2\mathbf{x}^T P \bar{\mathbf{B}} (\Delta\mathbf{v} + \eta) \end{aligned} \quad (36)$$

The term in Q is less than zero away from the equilibrium point, since Q is positive definite. It suffices, therefore, to show that the rest of the expression is less than or equal to zero. To simplify notation, let $\mathbf{w} = \bar{\mathbf{B}}^T P \mathbf{x}$. If $\mathbf{w} = 0$, then the last term of eq. (36) is zero and the entire expression is negative definite. If $\mathbf{w} \neq 0$, then

$$\Delta\mathbf{v} = -\rho \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad (37)$$

and the last term of (36) (neglecting the constant factor) becomes

$$\mathbf{w}^T \left(-\rho \frac{\mathbf{w}}{\|\mathbf{w}\|} + \eta \right) = \frac{-\rho \mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} + \mathbf{w}^T \eta \quad (38)$$

$$\leq -\rho \|\mathbf{w}\| + \|\mathbf{w}\| \|\eta\| \quad (39)$$

$$= \|\mathbf{w}\| (-\rho + \|\eta\|) \quad (40)$$

$$\leq 0 \quad (41)$$

since $\|\eta\| \leq \rho$.

Showing that V satisfies the two criteria given above is not sufficient for proving asymptotic stability using Lyapunov's second method [21]. In particular, although we have shown that \dot{V} is negative along solution trajectories of the system, we can not guarantee the existence of a solution since V is not continuous. In a similar proof, Spong and Vidyasagar [30] note that it is possible to show the existence of a solution in a generalized (set-theoretic) sense.

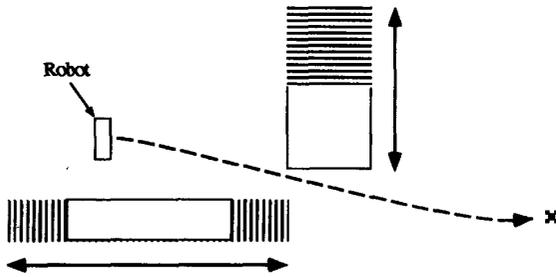


Fig. 3. 2D polygon simulation.

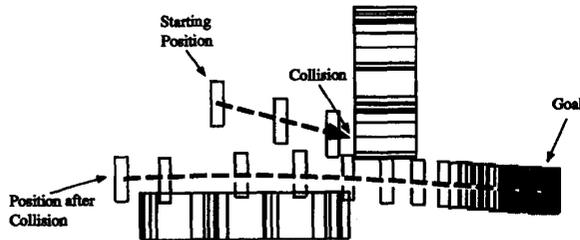


Fig. 4. Possible collision with moving obstacle.

VI. RESULTS

In this section we evaluate our system's performance in three test scenarios: (1) a polygonal robot translating in the plane among moving polygonal obstacles, (2) a two-link articulated planar arm moving among polygonal obstacles, and (3) a Stanford arm moving in a workspace that is shared by a two-link planar arm.

A. A Polygonal Robot Translating Among Polygonal Obstacles

In this scenario, two rectangular obstacles translate back and forth along a "track" as shown in Fig. 3. The robot is shown in its initial position, and its goal position is indicated by the 'X'.

Fig. 4 illustrates the type of collision that can occur if the motion planner/controller is not given the ability to consider the velocity of the moving obstacles. Such a collision can occur in any system with a delay between sensing and action, if that system relies only on instantaneous position information. The figure also illustrates the possibility of a discontinuity in the commanded motion of the robot if a standard repulsive potential function is used. (In the figure, this is illustrated by showing the position of the robot after the collision as being far removed from the point of collision). This discontinuity can also be seen the time response of the X and Y position, shown in Fig. 5.

We now examine the performance of our planner/controller in the same situation. The attractive and repulsive gains are unchanged from the previous scenario. However, we now have the added benefit of the attractive and repulsive filters. Both filters are lead filters, with a pole at $s = -20$ and a zero at $a = -0.1$, discretized using the bilinear transformation [25]. The path is shown in Fig. 6 with a position time response in Fig. 7 and associated repulsive forces in Fig. 8. Here, two obstacles are in motion (one moves back and forth along the

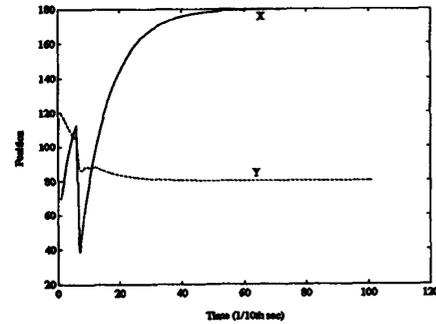


Fig. 5. Time response of robot position.

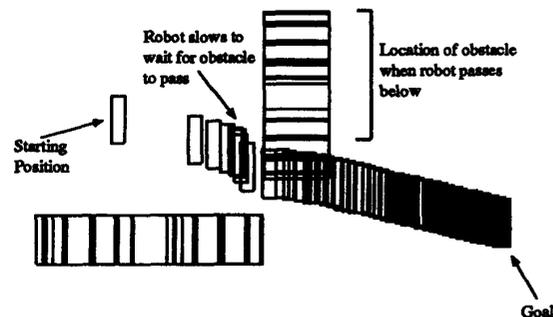


Fig. 6. Motion of robot under our controller/planner system.

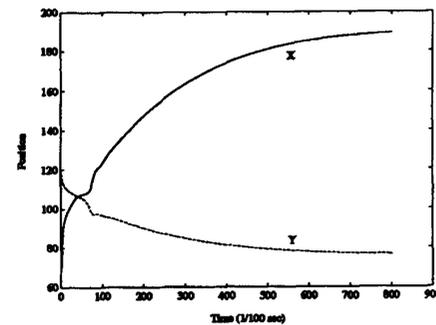


Fig. 7. Time Response for the motion.

horizontal direction, while the other moves back and forth along the vertical direction), and the robot passes between them without collision. Note that since the motion of the obstacles is illustrated by showing the obstacles in multiple positions, it appears that a collision has occurred. This is merely an artifact of the method of display. As can be seen from the figures, the motion and the forces are smooth.

B. A Two-Link Planar Arm

In the second scenario, our planner/controller is used in conjunction with a two-link planar arm (i.e., a two-link arm, with both links constrained to lie in a plane), as shown in

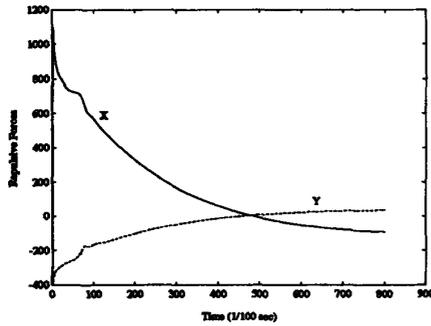


Fig. 8. Repulsive Forces for the motion.

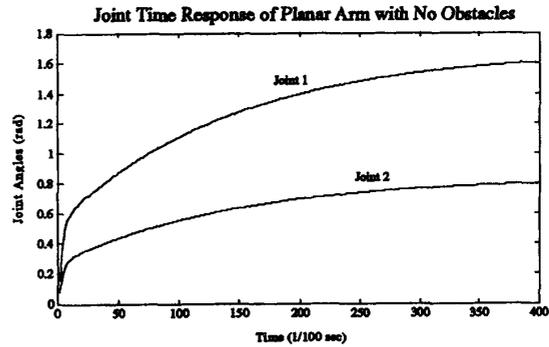


Fig. 11. Joint response of the planar arm.

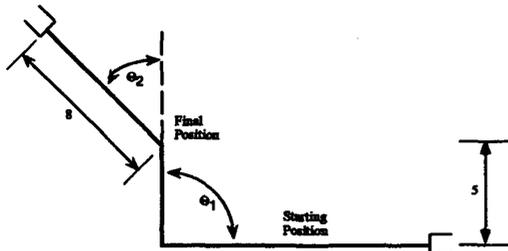


Fig. 9. Planar robot initial position and goal.

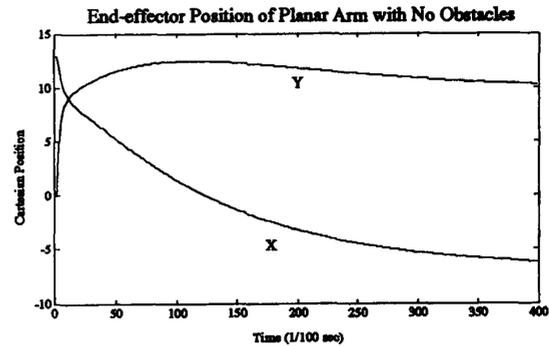


Fig. 12. Time response of the end effector.

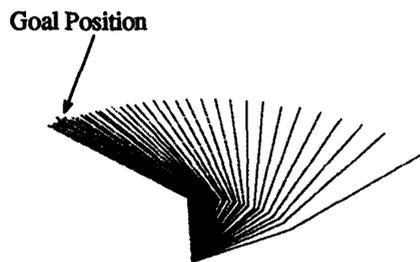


Fig. 10 Motion of the planner arm motion in the absence of obstacles.

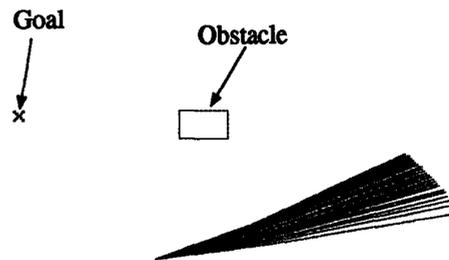


Fig. 13. The planar arm in a local minimum.

Fig. 9. Again, both the attractive and repulsive filters are implemented as lead filters with pole at -20 and zero at -0.1 . The first link of the robot is 5 units long and second link is 8 units long. We present three separate simulations with this scenario: (1) the robot moving with no obstacles in the workspace, (2) the robot converging to a local minimum, and (3) the robot moving from its goal position to avoid a single translating obstacle and then returning to its goal position.

The purpose of the first simulation is to examine the performance of the attractive loop. Since all of the closed-loop poles are real, we expect the motion to be overdamped. That this is the case can be seen from Figs. 10-12. Fig. 10 shows the motion of the robot, Fig. 11 shows the position response of each joint, and Fig. 12 shows the position response of the end effector. As can be seen in the figures, the overshoot is not exactly zero. This is a result of discretization, which adds some uncertainty to the transient parameters. However, the error is small and the overall performance is not significantly affected.

By introducing a stationary obstacle, we illustrate the performance of the system as it converges to a local minimum. This is shown in Fig. 13. The joint time response, shown in Fig. 14, remains overdamped.

Finally, a moving obstacle is introduced into the workspace, as shown in Fig. 15. The obstacle moves to the left at a speed of 0.1 units per time step. The boundary distance is the same as that for the local minimum simulation, with $\rho_0 = 15$. The robot is initially placed at the goal position, so the first motion in the figure is to avoid moving obstacle. After the obstacle has passed, the robot begins to return to its goal position. We note that if the dynamic response of the obstacles is not considered, a collision results, similar to the case shown in the previous section. This situation is illustrated in Fig. 16. From top to bottom in the three frames, the obstacle moves in a straight line from right to left.

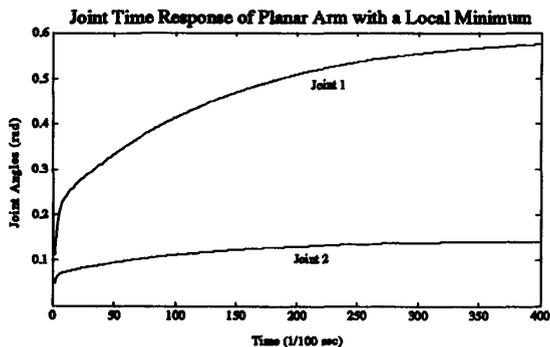


Fig. 14. Joint time response of planar arm in the case of a local minimum.

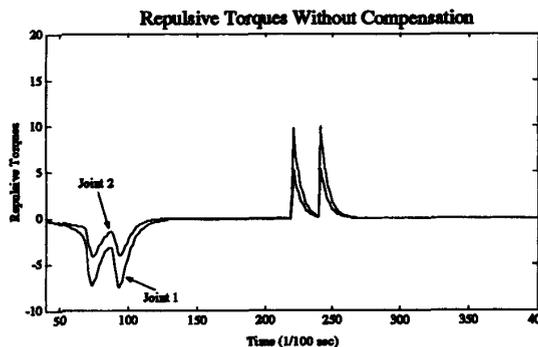


Fig. 17. Repulsive torques without compensation.

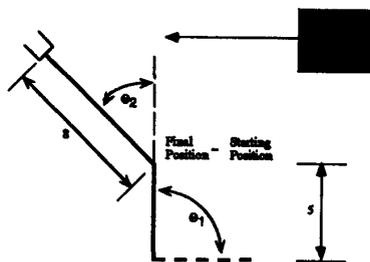


Fig. 15. Planar arm in the presence of a moving obstacle.

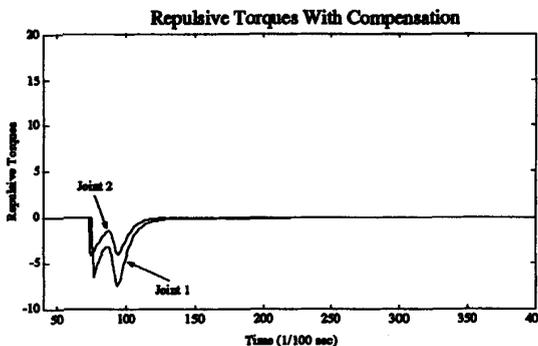


Fig. 18. Repulsive torques with compensation.

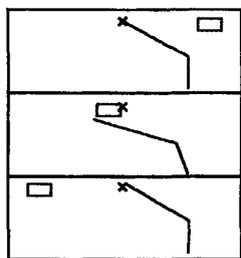


Fig. 16. Motion of the planar arm in the presence of a moving obstacle.

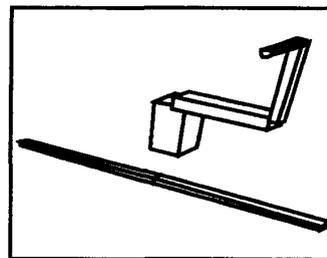


Fig. 19. 3D Multi-robot simulation.

In the examples shown thus far, obstacles approach the robot, resulting in increased repulsive force. However, what is not shown in these examples is that as obstacles move away from the robot, a negative repulsive force is created essentially pulling the robot into the “wake” of the moving obstacle. This problem is alleviated simply by adding a compensator to the repulsive loop that zeros such repulsive forces. Fig. 17 shows the output of the repulsive filter without compensation, and Fig. 18 shows the output of the repulsive filter with compensation.

C. Stanford Arm and Planar Arm

In the final scenario, we examine the behavior of two robots moving in a three-dimensional workspace, specifically a Stanford arm and a planar robot. Here, we consider only

the first four links of the Stanford arm. In this example, we use our planner/controller only with the Stanford arm, and treat the second robot as a moving obstacle. The initial positions for the two arms are shown in Fig. 19. The Stanford arm’s goal is to proceed to a point beyond the plane of motion of the planar arm. The planar arm starts at joint position (0, 0) (i.e. horizontal) and moves to joint position (90, 0) (i.e. vertical).

An important characteristic of the two-robot system is the speed of the planar arm relative to the speed of the Stanford arm. For example, if the planar arm moves slowly relative to the Stanford arm, the planner/controller will allow the Stanford arm to pass above the planar arm on the way to its goal. If the planar arm then moves upward, the Stanford arm will be forced

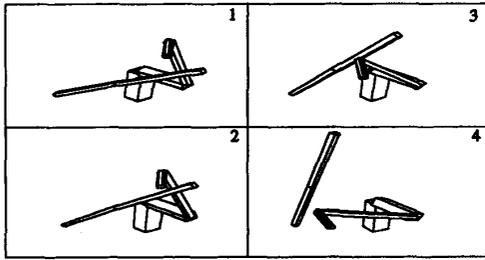


Fig. 20. Stanford arm avoids and passes planar arm.

back to allow the planar arm to pass. If, on the other hand, the planar arm moves at a much higher speed than the Stanford arm, the Stanford arm will effectively wait for the planar arm to pass before moving to its goal position. This is a key distinction between our approach and traditional potential field planners (which use only instantaneous position information to compute repulsive forces). We note that if the dynamic response of the planar arm is not considered, a collision results.

In our simulation, the speed of the planar arm was much higher than that of the Stanford arm. The resulting motion is shown in Fig. 20. The planar arm rises steadily toward its goal under the control of a standard PD control algorithm. The Stanford arm waits until the planar arm has passed and then slips around the end-effector of the planar arm to proceed toward the goal.

VII. CONCLUSION

We have presented an integrated architecture for real-time planning and control of robot motions, for a robot operating in the presence of moving obstacles whose trajectories are not known a priori. The architecture consists of three separate control loops. An inner loop is used to linearize robot dynamics, and two outer loops are used to implement a potential field motion planning algorithm. These outer control loops correspond to an attractive force (which attracts the robot to its goal configuration), and to repulsive forces (which repel the robot from the moving obstacles).

Our architecture is unique in both its control theoretic and its motion planning aspects. From a control theory perspective, our approach is unique in that the outer control loops are used to effect both desirable transient response and collision avoidance. From a motion planning perspective, our approach is unique in that the dynamic performance of both the robot and the moving obstacles are considered, unlike traditional artificial potential field planners, which take into account only the instantaneous relative positions of the robot and the obstacles.

ACKNOWLEDGMENT

The authors are grateful to Michael Barbehenn, Steve LaValle, Andres Castano, and the anonymous reviewers for many helpful comments.

REFERENCES

- [1] S. Arimoto and F. Miyazaki, "Stability and robustness of PID feedback control for manipulators of sensory capability," in *Proceedings of the 1st International Symposium on Robotics Research*, pp. 783-799, 1983.
- [2] J. Barraquand, B. Langlois and J. C. Latombe. "Numerical potential field techniques for robot path planning," *IEEE Trans. on Syst., Man and Cybern.*, vol. 22, no., 2, 224-241, 1992.
- [3] J. Barraquand and J. C. Latombe, "Robot motion planning: A distributed representation approach," *International Journal of Robotics Research*, vol. 10, no. 6, pp. 628-649, Dec. 1991.
- [4] L. Cai and A. A. Goldenberg, "Robust control of position and force for a robot manipulator in non-contact and contact tasks," in *Proceedings of the American Control Conference*, pp. 1905-1922, 1989.
- [5] E. Cheung and V. Lumelsky, "Development of sensitive skin for a 3D robot arm operating in an uncertain environment," in *Proc. IEEE Int'l Conference on Robotics and Automation*, May 1989.
- [6] E. Cheung and V. Lumelsky, "Motion planning for a whole-sensitive robot arm manipulator," in *Proc. IEEE Int'l Conference on Robotics and Automation*, May 1990.
- [7] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Pub. Co., Inc., Reading, Mass., 1989.
- [8] J. de LaMadrid and M. Gini, "Path tracking through uncharted moving obstacles," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 20, no. 6, pp. 1408-1422, Nov.-Dec. 1990.
- [9] G. Franklin, J. Powell and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company, Inc., 1991.
- [10] N. Hogan, "Impedance control: An approach to manipulation," *ASME Journal of Dynamic Systems and Measurement*, vol. 107, pp. 1-7, 1985.
- [11] N. Kehtarnavaz and S. Li, "Collision-free navigation scheme in the presence of moving obstacles," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 808-813, 1988.
- [12] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-96, 1986.
- [13] O. Khatib and J.-F. LeMaitre, "Dynamic control of manipulators operating in a complex environment," in *Proceeding 3rd International CISM-IFTOMM Symposium*, pp. 267-282, 1978.
- [14] D. E. Koditschek, "Exact robot navigation by means of potential functions: Some topological considerations," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 1-6, Raleigh, NC, 1987.
- [15] D. E. Koditschek, "Robot planning and control via potential functions," in *The Robotics Review 1*, pp. 349-367. MIT Press, 1989.
- [16] V. G. Kountouris and H. E. Stephanou, "Dynamic modularization and synchronization for intelligent robot coordination: The concept of functional time-dependency," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 508-513, Sacramento, CA, April 1991.
- [17] K. Kreutz, "On manipulator control by exact linearization," *IEEE Trans. on Automatic Control*, vol. 34, pp. 763-767, Jul. 1989.
- [18] K. J. Kyriakopoulos and G. N. Saridis, "Minimum distance estimation and collision prediction under uncertainty for on-line robot motion planning," in *11th Triennial World Conference of the IFAC*, pp. 93-98, 1990.
- [19] K. J. Kyriakopoulos and G. N. Saridis, "Collision avoidance of mobile robots in non-stationary environments," in *IEEE International Conference on Robotics and Automation*, pp. 904-909, 1991.
- [20] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
- [21] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*. John Wiley and Sons, New York, NY, 1979.
- [22] T. Nagata, K. Honda and Y. Teramoto, "Multirobot plan generation in a continuous domain: Planning by use of plan graph and avoiding collisions among robots," *IEEE Trans. on Robotics and Automation*, vol. 4, no. 1, pp. 2-13, Feb. 1988.
- [23] W. S. Newman and N. H. Hogan, "High speed robotic control and obstacle avoidance using dynamic potential functions," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 14-24, Mar. 1987.
- [24] P. A. O'Donnell and T. Lozano-Pérez, "Deadlock-free and collision-free coordination of two robot manipulators," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 484-489, 1989.
- [25] Katsuhiko Ogata, *Discrete-Time Control Systems*. Prentice-Hall, Inc., 1987.
- [26] S. Premvuti and S. Yuta, "Consideration on the cooperation of multiple autonomous mobile robots," in *IEEE International Workshop on Intelligent Robots and Systems*, pp. 59-63, 1990.
- [27] J. H. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," in *Proc. of IEEE Symp. on Foundat. of Comp. Sci.*, pp. 144-154, 1985.

- [28] J. T. Schwartz and M. Sharir, "On the piano movers' problem: III. Coordinating the motion of several independent bodies," *International Journal of Robotics Research*, vol. 2, no. 3, pp. 97-140, 1983.
- [29] K. G. Shin and Q. Zheng, "Minimum-time collision-free trajectory planning for dual-robot systems," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 641-644, Oct. 1992.
- [30] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. John Wiley and Sons, NY, NY, 1989.
- [31] Mark W. Spong, "Control of flexible joint robots: A survey," *Technical Report UILU-ENG-90-2203, DC-116*, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Feb. 1990.
- [32] M. W. Spong and M. Vidyasagar, "Robust linear computerized design for nonlinear robotics control," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 345-351, Aug. 1987.
- [33] J. Steel, "An algorithm with real-time response for avoiding moving obstacles," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 551-556, 1989.
- [34] J. Steel and G. Starr, "Mobile robot path planning in dynamic environments," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 922-925, 1988.
- [35] C. Tsai, "Multiple robot coordination and programming," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 978-985, Sacramento, CA, Apr. 1991.
- [36] L. Tychonievich, "A maneuvering-board approach to path planning among moving obstacles," in *11th International Joint Conference on Artificial Intelligence Applications*, 1989.
- [37] L. Wang and W. Tsai, "Collision avoidance by a modified least-mean-square-error classification scheme for indoor autonomous land vehicle navigation," *Journal of Robotics Systems*, vol. 8, pp. 677-698, Oct. 1991.
- [38] Q. Zhu, "Hidden markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, pp. 390-397, Jun. 1991.

Robert Spence received his B.S. in electrical engineering from the Rose-Hulman Institute of Technology, Terre Haute, IN, in May 1990. He completed his M.S. in Electrical Engineering at the University of Illinois at Urbana-Champaign in May 1992.



Seth Hutchinson received his Ph.D. from Purdue University, West Lafayette, IN, in 1988. He spent 1989 as a Visiting Assistant Professor of Electrical Engineering at Purdue University. In 1990, Dr. Hutchinson joined the faculty at the University of Illinois, Urbana-Champaign, where he is currently an Assistant Professor in the Department of Electrical and Computer Engineering, and the Beckman Institute for Advanced Science and Technology.

Professor Hutchinson's current research interests include integration of vision, force and position sensing for robot motion planning and control, dynamic planning and control, dynamic planning of sensing strategies, constraint based reasoning, task planning for automated assembly, evidential reasoning applied to model based object recognition, and sensor integration.