Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems Raleigh, NC July 7-10, 1992

# Dealing with Unexpected Moving Obstacles by Integrating Potential Field Planning with Inverse Dynamics Control

Rob Spence Seth Hutchinson
The Beckman Institute for Advanced Science and Technology
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

## Abstract

*We present a motion planning/control system that deals with moving obstacles whose trajectories are not known a priori. An artificial potential field planner is tightly coupled with a robust inverse dynamic controller, allowing the robot to avoid obstacles in real-time while retaining the benefits of inverse dynamic control. Our implementation of the artificial potential field planner uses digital filtering techniques to shape the input signal to the inverse dynamic controller, so that system's response to moving obstacles will depend, not only on the position of those obstacles, but also on their velocity relative to the robot. We prove stability along solution trajectories of the system in the absence of obstacles, and discuss stability issues that arise when obstacles are present.*

## 1 Introduction

Moving a robotic manipulator to a goal configuration while avoiding collisions with obstacles has been a key problem in robotics research over the last decade. The artificial potential field method of motion planning has been used to solve this problem, both in the robotics community [5, 1, 6, 8, 9] and in the artificial intelligence community [3, 4]. Typically, such planning systems do their work off-line, given as input the initial and goal configurations of the robot, and a complete description of the workspace (including descriptions of the obstacles in the workspace).

A recurring problem with standard potential field planners is performance, especially in the presence of moving obstacles. Most planners do not have a method of dealing with obstacles for which there is little or no *a priori* information. In addition, the interaction between the planner and the dynamic control system is extremely limited. As such the planner is less able to utilize the favorable aspects of control, notably rapid response and good transient performance.

It is interesting to note that the method of artificial potential fields was originally proposed for *real-time* obstacle avoidance, not off-line motion planning [6]. To implement the real-time approach, artificial repulsive forces induced by obstacles and artificial attractive forces induced by the goal position are incorporated directly into the robot control system. As a result, the robot is able to avoid obstacles in the workspace, while retaining the high performance of a real-time control system.

We propose a control system architecture to implement both an artificial potential fields planner and a real-time controller in the same system. We model the attractive potential field as an attractive control loop, with the repulsive forces acting as a disturbance upon that loop (this is illustrated in Fig. 1). We first analyze this system under the assumption that the repulsive field is zero everywhere (which corresponds to the assumption that there are no obstacles in the work space). The analysis proves that our controller is asymptotically stable (i.e. that it converges to the goal configuration) when no obstacles are present. We then extend our discussion to the case of a non-zero repulsive field generated by obstacles, including moving obstacles. It is well known that in the presence of obstacles there does not, in general, exist a potential function with a single minimum at the goal configuration [7] (sometimes referred to as a *global navigation function*[10]). Therefore, in this case we clearly cannot prove convergence to the goal configuration. However, we argue that the resulting system is still bounded given that a set of weak assumptions hold.

## 2 Motion Control Using Artificial Potential Fields

Before we present the control structure, we briefly review the artificial potential field approach to real-time obstacle avoidance. This method was originally reported by Khatib [6], and a tutorial description can be found in [10].

Using the artificial potential field method, we treat the robot manipulator (represented as a point in configuration space) as a particle under the influence of an artificial potential field $U$. The field $U$ is the sum of an attractive field and a repulsive field

$$U(\mathbf{q}) = U_{att}(\mathbf{q}) + U_{rep}(\mathbf{q}), \qquad (1)$$

where $\mathbf{q}$ represents a point in the robot's configuration space. The attractive field typically depends only on the goal configuration and current manipulator position, while the repulsive field typically depends on the distance from manipulator to obstacles in the work space. The manipulator motion is determined by the

0-7803-0737-2/92$03.00 1992©IEEE

artificial force exerted on the manipulator

$$\mathbf{F}(\mathbf{q}) = -\nabla U(\mathbf{q}). \quad (2)$$

The planning component of the artificial potential field technique uses the generated field to form a path for the robot to follow. The planner starts with the robot's initial position and determines the direction of the steepest gradient of the field. The next point in the trajectory is determined by moving a set distance along this gradient. It should be noted that this is usually accomplished off-line, so complete knowledge of the workspace is assumed. Any dynamic properties of the environment must be completely modeled before the robot motion begins.

Once this trajectory is determined it is fed to a controller, usually as a series of setpoints. After receiving one of these inputs, the controller will move the robot from its initial position (or the last set point) to the goal with a transient performance that will depend upon the dynamics of the control loop.

This technique has two drawbacks:

1. The planner must compromise between the number of set points (and inversely the distance between set points) and the time it will take to compute the path. In other words, the smaller the distance between set points the more accurate the motion, but the longer the planner will take to generate the path. If the planner trades off accuracy for speed, then the robot might oscillate around the goal position or behave erratically near obstacles.

2. The control system is not equipped to recognize the presence of obstacles. If the planner has not sufficiently accounted for the control system's transient motions, or obstacles appear unexpectedly in the workspace after the motion begins, a collision could occur.

By integrating the control and the planning architectures, we minimize the effect of these problems, improving the performance of the manipulator while at the same time decreasing the likelihood of contact with unexpected obstacles in the workspace.

# 3 Control System Architecture

Two digital filters in the system shape the signals corresponding to input torques to produce acceleration values. In the standard potential fields algorithm, these filters would simply be constant gain, with no dynamics. These gains correspond to the attractive and repulsive gains in the standard potential field algorithm. The two digital filters in our system give the designer a great deal of flexibility in choosing the system potential functions.

The vector sum of the attractive and repulsive accelerations appears as the input signal to the inverse dynamic controller. This controller (also known as a *computed torque* controller) translates the desired accelerations to the necessary voltage into the joint motors that will result in these accelerations.

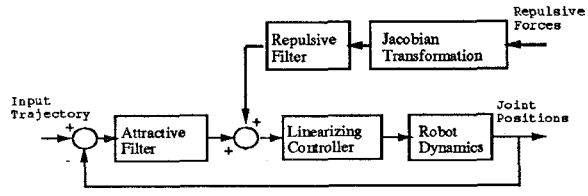


Figure 1: Controller Block Diagram

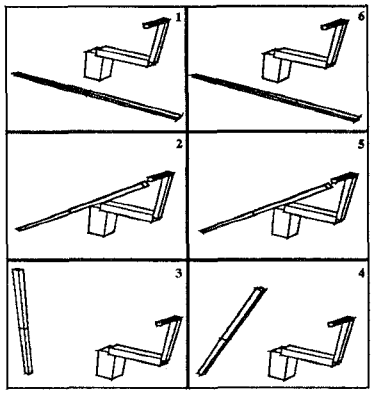


Figure 2: Repetitive Task - 1 Cycle

## 3.1 Attractive Potential

As can be seen in figure 1, the attractive forces are developed in two stages. First the distance to the goal, expressed in joint space, is determined. Then a digital filter is used to transform this distance into a vector of joint accelerations. This filter is used to shape the transient response of the robot as it moves from its initial configuration to the goal.

## 3.2 Repulsive Potential

The repulsive torques (one formulation of which is described in the Appendix) are also transformed into joint accelerations using a digital filter. The repulsive forces are regarded as a disturbance entering the system just prior to the robot dynamics. Disturbances are, by nature, unpredictable. However, we have the added benefit, since the repulsive forces are purely imaginary, of shaping these forces as we see fit, subject to the requirement that they maintain the characteristic of growing arbitrarily large as the robot approaches an obstacle.

We are interested not only in the magnitude of the repulsive torque signal but also how it is changing over time. Rapidly changing repulsive torques suggest that either the manipulator is approaching an obstacle (or another robot) very quickly or that the robot is itself being approached by some moving object. Another possibility of interest is that the robot is nearing a

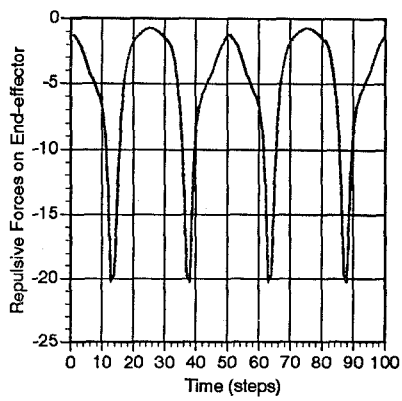


Figure 3: Periodic Repulsive Forces

subset of the workspace where another manipulator is moving in a repetitive pattern (e.g. a 'slave' mechanism performing a standard pick-and-place task.) This example is shown in Fig. 2 where a Stanford arm is standing off from a planar manipulator that is moving continuously from a horizontal to a vertical position and back again.

Fig. 3 shows the forces normal to the plane of motion of the planar arm as seen by the end-effector of the Stanford arm. We can translate this force to joint torques using the manipulator Jacobian.

In these situations, obstacles moving in the environment at different speeds relative to the main robot will generate repulsive torques of varying frequency acting upon the joints of the robot. The repulsive filter, however, enables the designer to design a potential function that will compensate differently for different frequency inputs, while still maintaining closed-loop stability. This allows the robot to react differently to environmental variables occurring at different speeds.

For example, the Stanford arm shown in Fig. 2 may be able to reach a goal beyond the planar robot if the planar robot is moving at a sufficiently slow speed. A standard potential field planner will be unable to distinguish the relative speed of the planar manipulator. By introducing a repulsive filter with high-pass characteristics, the designer can cause the repulsive forces to generate increasingly larger torques as the speed of the planar manipulator increases. With the torques acting upon it, the Stanford arm will be unable to extend into the region of a quick moving planar manipulator, but will be able to pass should the planar arm move more slowly. The DC characteristics of the repulsive filter will determine the robot's reaction to static obstacles.

## 4 Stability of Attractive Loop

Our first goal is to show the conditions under which the attractive loop can be proven to be asymptotically stable given direct interaction with the robot dynamics.

The robot dynamic model for an $n$-link articulated robot is as follows:

$$M(\mathbf{q})\ddot{\mathbf{q}} + h(\mathbf{q}, \dot{\mathbf{q}}) = \tau \tag{3}$$

where $\mathbf{q}$ is the vector of joint variables, $M(\mathbf{q})$ is a symmetric positive definite matrix that is usually dependent on the robot position, and $h(\mathbf{q}, \dot{\mathbf{q}})$ represents the combination of the Coriolis, centrifugal, and gravitational forces. $\tau$ is the vector of joint torques. These equations of motion were developed using Lagrangian energy relationships.

Since the above equation is clearly nonlinear, we choose an inverse dynamics control law [12] to cancel the nonlinear terms

$$\tau = \hat{M}(\mathbf{q})v + \hat{h}(\mathbf{q}, \dot{\mathbf{q}}) \tag{4}$$

The "hat" superscript refers to the fact that the model of the robot is not exact. We will need to show that, despite these inaccuracies, we can still stabilize the robot. Furthermore we wish to build confidence that for small differences between the actual dynamic parameters and the modeled dynamics, the transient characteristics of the attractive loop will still behave in a nearly nominal fashion.

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

$$\ddot{\mathbf{q}} = \mathbf{v} + \eta(\mathbf{v}, \mathbf{q}, \dot{\mathbf{q}}) \tag{5}$$

where $\eta$ is given by

$$\eta = E(\mathbf{q})\mathbf{v} + M^{-1}\Delta h \tag{6}$$

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

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

$$\dot{\mathbf{y}} = A\mathbf{y} + B(\mathbf{v} + \eta) \tag{7}$$

$$\mathbf{q} = C\mathbf{y} \tag{8}$$

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

$$A = \begin{bmatrix} 0 & I_n \\ 0 & 0 \end{bmatrix} \tag{9}$$

$$B = \begin{bmatrix} 0 \\ I_n \end{bmatrix} \tag{10}$$

$$C = \begin{bmatrix} I_n & 0 \end{bmatrix} \tag{11}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{12}$$

We want the robot to approach a set position and velocity. We define the *error* as the difference between the current states and the desired states.

$$\mathbf{e} = \mathbf{y} - \mathbf{y}^d = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} \mathbf{q}^d \\ \dot{\mathbf{q}}^d \end{bmatrix} \tag{13}$$

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

$$\dot{\mathbf{e}} = A\mathbf{e} + B(\mathbf{v} + \eta - \ddot{\mathbf{q}}^d) \tag{14}$$

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

The attractive filter also has a state-space model, given by:

$$\begin{aligned} \dot{\mathbf{z}} &= A_a\mathbf{z} + B_a\mathbf{u} && (15)\\ \alpha_a &= C_a\mathbf{z} + D_a\mathbf{u} && (16) \end{aligned}$$

where $\mathbf{u} = \mathbf{q}^d - \mathbf{q}$ is the difference between the desired and current position. It should be noted that the input to the attractive filter is zero in steady state. One of the equilibrium positions for the filter states is the origin. We will investigate system stability around this equilibrium point.

We define the input $\mathbf{v}$ to the nonlinear inner-loop controller as

$$\mathbf{v} = \alpha_a + \Delta\mathbf{v} \tag{17}$$

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

The filter/linearized robot equations become

$$\begin{aligned} \dot{\mathbf{e}} &= A\mathbf{e} + B(\alpha_a + \Delta\mathbf{v} + \eta)\\ &= A\mathbf{e} + B(C_a\mathbf{z} + D_a(\mathbf{q}^d - \mathbf{q}) + \Delta\mathbf{v} + \eta)\\ &= A\mathbf{e} + BC_a\mathbf{z} - BD_aC\mathbf{e} + B\Delta v + B\eta\\ \dot{\mathbf{z}} &= A_a\mathbf{z} + B_a(\mathbf{q}^d - \mathbf{q})\\ &= A_a\mathbf{z} - B_aC\mathbf{e} \end{aligned}$$

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

$$\mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \mathbf{z} \end{bmatrix} \tag{18}$$

which gives us the state-space model

$$\dot{\mathbf{x}} = \bar{A}\mathbf{x} + \bar{B}(\eta + \Delta\mathbf{v}) \tag{19}$$

where

$$\bar{A} = \begin{bmatrix} A - BD_aC & BC_a \\ -B_aC & A_a \end{bmatrix} \tag{20}$$

$$\bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tag{21}$$

We can design our attractive filter so that $\bar{A}$ is a stable matrix. This is can be accomplished by noting that for each joint we have a double-integrator linked with an attractive filter. The block diagram for an individual joint is shown in figure 4 where $G_c(s)$ is the transfer function for the attractive filter. The attractive filter for each joint can be designed using techniques from linear system theory. As long as the individual loops for each joint are stable, $\bar{A}$ is stable (Hurwitz).

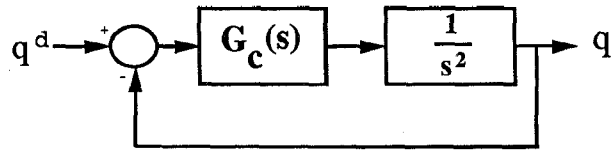


Figure 4: Linear Block Diagram for a Single Joint

Even though the nominal system (for $\eta = 0$) is stable, we still have to show that we can guarantee stability for $\eta \neq 0$ or, in other words, a system where the model is not perfectly accurate. The control input $\Delta v$ can be chosen by the designer to accomplish this.

First, it is necessary to make a few assumptions about the magnitudes of the uncertainties.

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

It has been shown [12] that we can always find $\hat{M}$ that will satisfy Assumption 1 given known upper and lower bounds on $M$.

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

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

$$\begin{aligned} \|\Delta\mathbf{v}\| &\leq \rho(\mathbf{x},t) && (22)\\ \|\eta\| &\leq \rho(\mathbf{x},t) && (23) \end{aligned}$$

From these restrictions and Assumptions 1-2, we can derive a value for $\rho(\mathbf{x},t)$ implicitly using

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

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

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

$$\rho(\mathbf{x},t) = \frac{1}{1-\epsilon}\left[\epsilon\|\alpha_a\| + \bar{M}\phi(\mathbf{e},t)\right] \tag{24}$$

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

$$\alpha_a = [\ -D_e C \quad C_e\ ]\,\mathbf{x} \tag{25}$$

2. Find a positive definite matrix $P$ such that

$$\bar{A}^T P + P\bar{A} + Q = 0 \tag{26}$$

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

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

$$\Delta\mathbf{v} = \begin{cases} -\rho(\mathbf{e},t)\frac{\bar{B}^T P\mathbf{e}}{\|\bar{B}^T P\mathbf{e}\|} & \text{if } \|\bar{B}^T P\mathbf{e}\| \neq 0 \\ 0 & \text{if } \|\bar{B}^T P\mathbf{e}\| = 0 \end{cases} \tag{27}$$

where $\Delta\mathbf{v}$ does satisfy (22)

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

$$V(\mathbf{e},\mathbf{z}) = \mathbf{x}^T P\mathbf{x} \tag{28}$$

$V$ 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. $\dot{V} \leq 0$). For asymptotic stability, $\dot{V} < 0$ everywhere but the origin.

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 $\mathbf{x} = 0$ which is when the robot is at the goal and the filter states are zero.

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

$$\begin{aligned} \dot{V} &= \dot{\mathbf{x}}^T P\mathbf{x} + \mathbf{x}^T P\dot{\mathbf{x}} \\ &= \mathbf{x}^T(\bar{A}^T P + P\bar{A})\mathbf{x} + 2\mathbf{x}^T P\bar{B}(\Delta\mathbf{v} + \eta) \\ &= -\mathbf{x}^T Q\mathbf{x} + 2\mathbf{x}^T P\bar{B}(\Delta\mathbf{v} + \eta) \end{aligned} \tag{29}$$

The term in $Q$ is obviously 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 then or equal to zero. To simplify notation, let $\mathbf{w} = \bar{B}^T P\mathbf{x}$. If $\mathbf{w} = 0$ then the last term of (29) is zero and the entire expression is negative definite. If $\mathbf{w} \neq 0$ then

$$\Delta\mathbf{v} = -\rho\frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{30}$$

and the last term of (29) (neglecting the gain of 2) becomes

$$\begin{aligned} \mathbf{w}^T\left(-\rho\frac{\mathbf{w}}{\|\mathbf{w}\|} + \eta\right) &= \frac{-\rho\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|} + \mathbf{w}^T\eta & (31) \\ &\leq -\rho\|\mathbf{w}\| + \|\mathbf{w}\|\|\eta\| & (32) \\ &= \|\mathbf{w}\|(-\rho + \|\eta\|) & (33) \\ &\leq 0 & (34) \end{aligned}$$

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

Unfortunately, showing that $V$ satisfies the two criteria given above is not sufficient for proving global asymptotic stability using Lyapunov's second method [11]. 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. This issue is raised in a similar proof for the stability of inverse dynamics control under state feedback in [12], where it is pointed out that we can show the existence of a solution in a set-theoretic sense.

### 4.1 Stability in the Presence of Obstacles

When obstacles are present in the workspace, the repulsive potential will, in general, be non-zero. The repulsive acceleration $\alpha_r$ acts as a disturbance input into the robot dynamics. If the dynamics of the repulsive filter are stable, $\alpha_r$ will converge to a constant, possibly zero value, so long as the robot does not come in contact with any obstacle. We can limit obstacle contact by using the repulsive force formulation shown in Appendix A. This adopts the standard notation of allowing repulsive force gains upon a control point to grow unboundedly as the point approaches an obstacle. In a continuous time formulation of the closed loop system, the repulsive forces will eventually overcome the attractive force and cause the robot to accelerate away from the obstacle. Most control systems, however, are implemented in discrete time. Accordingly the chances of contact with an obstacle will also be a function of the sampling time. If the obstacles are moving, the chances of contact will depend upon how fast the obstacles move relative to the sampling rate. Note that anytime a standard planner is combined with a digital controller, the sampling time serves as a maximum rate of adaptability for the system. Accordingly, no *a priori* planner will be able to react to moving obstacles faster then the planner shown here (assuming the controller on the standard planner system has an equal or lower sampling rate) although the method by which it reacts might be more sophisticated.

If the steady-state value of $\alpha_r$ is small, but still non-zero, the robot position will converge to a point in the workspace distinct from the goal (a local minimum.) Various techniques have been formulated to combat this problem (e.g. random walks [2]). Since most of these rely upon using another planning technique to remove the robot from the minimum and then reengaging the potential field planner, there is nothing to prevent these techniques from being combined with the architecture shown here.

The planner presented here could potentially serve as a "safety-margin" for a higher level planner (utilizing cell decomposition, highway methods, or some

other technique.) Instead of following a simple step input, this system can track the trajectory supplied by the higher planner, while still retaining the same obstacle avoidance and transient capabilities as before. This adds robustness for the complete system in the face of unexpected changes in the workspace that occur after the *a priori* trajectory is determined.

## 5 Conclusions

We have proposed a control system architecture that integrates potential field planning with real-time control. The resulting system is modeled as an attractive loop, with the repulsive forces acting as disturbance upon that loop. We have analyzed this system for the case of no obstacles present and have shown the control loop to be asymptotically stable given a properly designed attractive filter in the forward-path. We have also discussed the behavior of the control loop in the presence of static and moving obstacles and argued that the continuous control loop will remain stabile although the goal may not necessarily be reached unless the problem of local minima has been accounted for.

## References


[1] J.R. Andrews and N. Hogan. Impedance control as a framework for implementing obstacle avoidance in a manipulator. In David Hardt and Wayne Book, editors, *Control of Manufacturing Processes and Robotic Systems*, pages 243–251. ASME, Boston, 1983.

[2] J. Barraquand and J. C. Latombe. A monte-carlo algorithm for path planning with many degrees of freedom. In *IEEE International Conference on Robotics and Automation*, pages 1712–1717, Cincinnati, OH, 1990.

[3] W. Choi, D. Zhu, and J.C. Latombe. Contingency tolerant robot motion planning and control. In *International Conference on INtelligent Robotics and Systems*, pages 78–86, September 1989.

[4] T. Fraichard and C. Laugier. On line reactive planning for a non holonomic mobile in a dynamic world. In *IEEE International Conference on Robotics and Automation*, April 1991.

[5] Y. K. Hwang and N. Ahuja. Path planning using a potential field representation. Technical Report UICU-ENG-88-2251, University of Illinois at Urbana Champaign, 1988.

[6] O. Khatib. Real time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–96, 1986.

[7] D. E. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *IEEE International Conference on Robotics and Automation*, pages 1–6, Raleigh, NC, 1987.

[8] B.H. Krogh. A generalized potential field approach to obstacle avoidance control. In *Proc. Intl. Robotics Research Conf.*, August 1984.

[9] B.H. Krogh and C.E. Thorpe. Integrated path planning and dynamic steering control for autonomous vehicles. In *IEEE International Conference on Robotics and Automation*, pages 1664–1669, April 1986.

[10] J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[11] D. G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications.* John Wiley and Sons, NY, NY, 1979.

[12] M. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley and Sons, Inc., NY, NY, 1989.


## 6 Appendix A: Formulation of Repulsive Torque $\alpha_r$

Repulsive forces on an articulated manipulator should apply to control points all along the length of the manipulator in order to prevent any part of the manipulator from coming in contact with the obstacles. Given a collection of control points along the entire length of the manipulator, we can create the repulsive torque vector as follows:

- $F_i^p$ is the repulsive force on control point $p$ on link $i$ expressed in terms of frame 0 (the world coordinate frame).

  One common formulation [10] defines the force on the $p_{th}$ control point

  $$F_i^p = \frac{X_i^p - X_{OB_i^p}}{\|X_i^p - X_{OB_i^p}\|^2} \tag{35}$$

  where $X_i^p$ is the coordinates of the $p_{th}$ control point on link $i$ and $X_{OB_i^p}$ represents the coordinates of the obstacle point nearest to $X_i^p$. Both coordinates are expressed in the world frame. The norm used here is the standard $\|\|_2$ norm (the Euclidean distance metric).

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

  $$\tau_i = {J_{ip}}^T F_i^p \tag{36}$$

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

- $\bar{J}_{ip} = [\ J_{ip} \quad 0\ ]$ is the $6 \times n$ augmented Jacobian matrix where $n$ is the total number of links.

- $\tau_{rep}$, the vector of total repulsive torques on all joints of the robot, is

  $$\tau_{rep} = \sum_{i=1}^{n} \sum_{p=1}^{p_i} \bar{J}_{ip}^T F_i^p \tag{37}$$

  where $p_i$ is the total number of repulsive control points on link i, and tranmforms forces expressed in the world coordinate frame to forces expressed in the frame of link $i$.