



October 1997

Motion Planning in Humans and Robots

R. Vijay Kumar

University of Pennsylvania, kumar@grasp.upenn.edu

Milos Zefran

University of Pennsylvania

James P. Ostrowski

University of Pennsylvania, jpo@grasp.cis.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/penniur_papers

Kumar, R. Vijay; Zefran, Milos; and Ostrowski, James P., "Motion Planning in Humans and Robots" (1997). *Penn IUR Publications*. 15.
http://repository.upenn.edu/penniur_papers/15

The Eighth International Symposium of Robotics Research; Hayama, Japan, October 3-7, 1997.

This paper is posted at Scholarly Commons. http://repository.upenn.edu/penniur_papers/15

For more information, please contact libraryrepository@pobox.upenn.edu.

Motion Planning in Humans and Robots

Abstract

We present a general framework for generating trajectories and actuator forces that will take a robot system from an initial configuration to a goal configuration in the presence of obstacles observed with noisy sensors. The central idea is to find the motion plan that optimizes a performance criterion dictated by specific task requirements. The approach is motivated by studies of human voluntary manipulation tasks that suggest that human motions can be described as solutions of certain optimization problems.

Comments

The Eighth International Symposium of Robotics Research; Hayama, Japan, October 3-7, 1997.

Motion planning in humans and robots

Vijay Kumar Miloš Žefran J. Ostrowski

General Robotics and Active Sensory Perception (GRASP) Laboratory
University of Pennsylvania, 3401 Walnut St., Philadelphia, PA 19104-6228

email: kumar@central.cis.upenn.edu

Abstract

We present a general framework for generating trajectories and actuator forces that will take a robot system from an initial configuration to a goal configuration in the presence of obstacles observed with noisy sensors. The central idea is to find the motion plan that optimizes a performance criterion dictated by specific task requirements. The approach is motivated by studies of human voluntary manipulation tasks that suggest that human motions can be described as solutions of certain optimization problems.

1 Introduction

In recent years, we have developed a computational framework for generating open-loop motion plans for manipulation tasks in a deterministic environment (Žefran and Kumar 1997; Žefran et al. 1996). The emphasis in these tasks is on the kinematic and dynamic interactions between the robot(s) and the environment. This paper summarizes the approach and an extension to uncertain environments.

Our approach is grounded in the framework of continuous mathematics. We argue that in robotic systems, this framework allows modeling and synthesis of behavioral patterns that are essentially discrete in nature. At a different level of representation, it may be productive to represent such systems using a discrete-event framework. However, since our intention is to generate reference trajectories for closed loop control, we will pursue the continuous approach.

The motivation for the proposed planning paradigm comes from our studies of human manipulation. Human motions appear to minimize a certain integral cost functional (Desai et al. 1997; Flash and Hogan 1985; Kawato 1990; Garvin et al. 1997). We use this idea to compute motion plans for robot systems governed by (non-linear) dynamics, and moving amidst obstacles. Depending on the level at which the system is

modeled, we can obtain optimal kinematic trajectories or optimal actuator forces. We show that such motion plans can also encode discrete behaviors. In the last part of the paper, we discuss how this framework can be extended to handle uncertainty in the environment. In particular, we show that as more information about the world becomes available at finer levels of granularity, the motion plans can be efficiently refined.

2 Trajectory planning

When studying motion planning for artificial systems it is beneficial to study the formation of motion in humans. We can model the human body as an articulated linkage of rigid bodies, similar to how we model robots. In humans, there are many instances of redundancies that make the mappings between the task, joint, and actuator spaces non-invertible, suggesting that humans possess mechanisms for resolving these redundancies. The principles that govern human motion planning can therefore provide insight into motion planning for robots.

The hypothesis that voluntary, planar, reaching tasks (in R^2) performed under relaxed conditions may be described by optimality criteria has been explored by Flash and Hogan (1985), Kawato (1990) and others. Two natural questions that arise are (a) do such optimality criteria extend naturally to higher dimensions; and (b) can they explain tasks that involve constraints. In our studies of human manipulation, we have focused on tasks in $SE(2)$, the set of translations and rotations in a plane, and in which the left and right arm cooperatively grasp the object. In this case, the task space is no longer Euclidean. Further, the grasp introduces constraints due to the closed kinematic chain and the requirement of force closure. Our observations (Garvin et al. 1997) show that the kinematic properties of the measured trajectories exhibit a high degree of repeatability (within a subject and across subjects). The trajectories are approximately straight lines

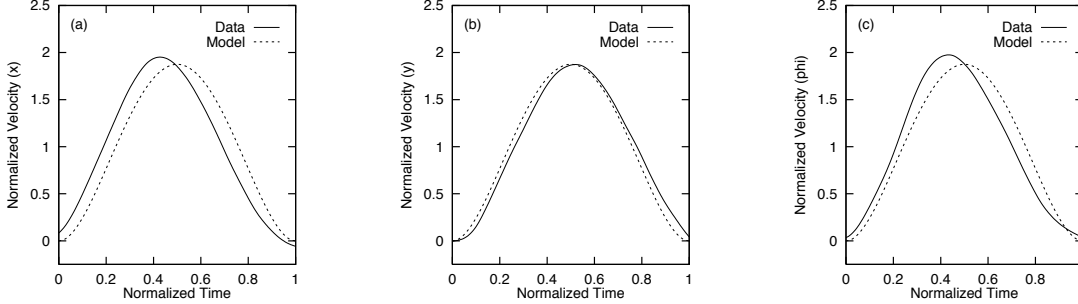


Figure 1: Measured velocity profiles for a planar two-handed manipulation by humans (solid) and those predicted by the minimum-jerk hypothesis (dotted) in the frontal plane (a), sagittal plane (b) and for rotation (c).

and have a smooth velocity profile. These trajectories depend only on the relative position and orientation of the initial and goal configurations and not on the global position and orientation. In other words, the trajectories are *left invariant*, or independent of the inertial reference frame.

In order to formalize planning of smooth motions that involve translations as well as rotations, and to study the invariance properties of different motion planning schemes, it is convenient to formulate kinematic motion planning in the framework of differential geometry and Lie groups. Using purely geometric ideas also makes the planning method independent of the description (parameterization) of the space¹. Geometric analysis reveals that it is necessary to define the concept of distance (a Riemannian metric) and a method of differentiation (an affine connection) in space before the notion of smoothness can be defined (Žefran and Kumar 1997). Since a Riemannian metric naturally leads to an affine connection, once a metric is chosen, trajectories with different smoothness properties can be generated. Further, by properly choosing a metric, we can obtain trajectories with desired invariance properties. A metric that is particularly interesting for motion planning and produces left-invariant trajectories is the kinetic energy metric. It embodies the inertial properties of a rigid body for which we wish to plan the trajectories.

Given a Riemannian metric $\langle \cdot, \cdot \rangle$, the problem of finding a smooth kinematic trajectory $\gamma(t)$ can be formulated as:

$$\min_{\gamma} \int_a^b \left\langle L\left(\gamma, \frac{d\gamma}{dt}\right), L\left(\gamma, \frac{d\gamma}{dt}\right) \right\rangle dt. \quad (1)$$

where L is a vector valued function that is a local measure of smoothness of the trajectory and usually depends on the affine connection correspond-

ing to the chosen metric. For example, the general expression for the minimum-jerk cost functional is:

$$J_{\text{jerk}} = \int_a^b \langle \nabla_V \nabla_V V, \nabla_V \nabla_V V \rangle dt. \quad (2)$$

In the equation, $V = \frac{d\gamma}{dt}$ is the velocity vector field and ∇ is the affine connection obtained from the chosen Riemannian metric. The resulting trajectories are given by the Euler-Lagrange equation:

$$\nabla_V^5 V + R(V, \nabla_V^3 V)V - R(\nabla_V V, \nabla_V^2 V)V = 0,$$

where R denotes the metric-dependent tensor describing the curvature properties of the space.

The measured trajectories and those minimizing the jerk cost functional (obtained by solving (1-2)) are shown in Figure 1 for a typical motion. This suggests that the observed motions are well predicted by the minimum-jerk hypothesis originally proposed by Flash and Hogan (1985).

The cost functional of the form (1) is not completely general, but it allows us to obtain generalized spline motions. For example, a maximally smooth trajectory that is C^1 continuous (*i.e.*, it can satisfy arbitrary boundary conditions on the velocities) is a generalization of a cubic spline and can be obtained by minimizing the integral of the acceleration along the trajectory:

$$J_{\text{acc}} = \int_a^b \langle \nabla_V V, \nabla_V V \rangle dt.$$

Figure 2 shows generalized cubic splines that satisfy boundary conditions on positions and velocities and pass through a given intermediate configuration for two different choices of the metric for the space².

While kinematic motion plans may not be adequate for some applications, they have the advantage that they can be easily computed. Some

¹This is particularly important in the context of rotations in three dimensions.

²More complicated, three-dimensional examples are presented in (Žefran and Kumar 1997).

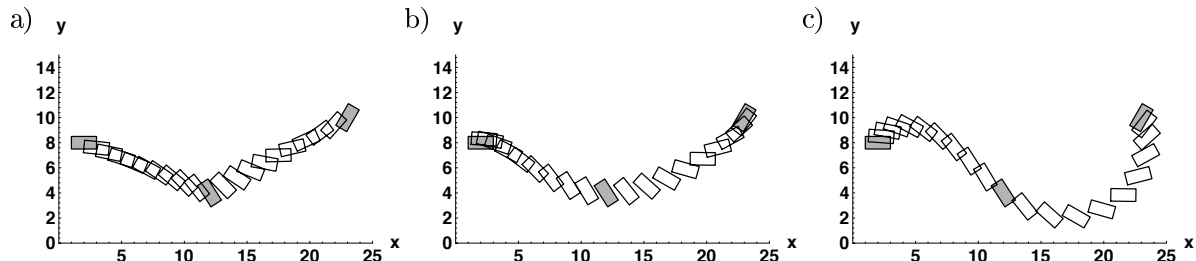


Figure 2: Examples of trajectories between specified configurations: (a) Piece-wise smooth geodesics; (b) Maximally smooth interpolant computed from a positive definite left-invariant metric; (c) Maximally smooth interpolant computed from an indefinite bi-invariant metric.

important problems have explicit, closed-form solutions (Žefran and Kumar 1997). If a detailed dynamic model of a mobile robot system is not available, it may be desirable to simply determine the smoothest trajectory that satisfies the required boundary conditions. In such a situation, left-invariance (invariance with respect to the choice of the inertial frame) is desirable. Further, if there is a 1-1 map between trajectories in the task space and the actuator forces that generate them, the kinematic motion plan uniquely determines the motion.

3 Dynamic constraints

Walking, grasping, and cooperative manipulation are tasks in which multiple articulated linkages are strongly dynamically coupled and the coordination of these interactions becomes critical for motion planning. A further characteristic of these tasks is that the dynamic equations may change as the system moves. For example, in a grasping task, the dynamic equations will change if a contact between a finger and the object is broken or if a new contact is established.

There are several properties that a dynamic motion planning method should possess: (a) the method must account for the dynamics of the system and provide the task space trajectory, the joint space trajectory, and the actuator forces; (b) it is desirable that trajectory generation and resolution of kinematic and actuator redundancies are performed within the same framework; (c) the method must be capable of dealing with additional equality and inequality constraints, such as kinematic closure equations, nonholonomic constraints, joint or actuator limits, and constraints due to obstacles; (d) since there are usually one or more natural measures of performance for the task, it is desirable to find a motion with the best performance; (e) one would like to develop plans that are robust with respect to modeling errors; and (f) explicitly incorporate the robot's ability

to use sensory information for error correction.

These requirements can be satisfied by formulating motion planning as a variational problem. In other words, given a performance criterion in the form of an integral cost functional:

$$J = \Psi(x(t_1), t_1) + \int_{t_0}^{t_1} L(x(t), u(t), t) dt, \quad (3)$$

and given the dynamic equations of the system:

$$\dot{x} = f(x, u, t) \quad (4)$$

the problem becomes that of finding a trajectory in the time interval $[t_0, t_1]$, that satisfies the dynamic equations (4), minimizes the cost functional (3) and satisfies possible additional equality and inequality constraints.

We have developed an efficient numerical technique for solving such problems based on finite-element techniques and finite-dimensional optimization. In addition, we have devised a simple scheme to solve problems in which the dynamic equations change as the system moves. This framework has been applied to several practical examples such as two-arm coordinated manipulation with frictional constraints, multi-fingered grasping and coordination of multiple mobile manipulators (Žefran et al. 1996). The approach allows us to solve in reasonable time complex motion planning problems that have not been attempted before³.

Figure 3 shows an example⁴ of the motion plan for two mobile manipulators negotiating the free space between two obstacles. In human, planar grasping and manipulation tasks with two arms, the observed trajectories and the distribution of

³The typical time for a system with a 20-dimensional state space on a 200 MHz SGI (for example, the system shown in Figure 3) is two minutes.

⁴In more complicated examples, we assume that an initial path that satisfies the geometric constraints is available. There is extensive literature that addresses the generation of such a path (Latombe 1991). Our method is not specifically designed to solve such problems as the piano-movers problem.

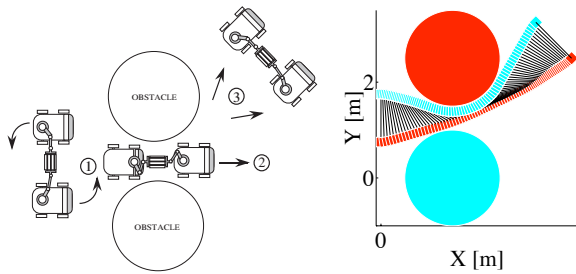


Figure 3: A motion plan for two mobile manipulators negotiating a narrow corridor.

forces suggest a maximally smooth variation of the joint torques (Desai et al. 1997). Letting the integrand in Equation (3) be the norm of the vector of the rate of change of torques, and imposing constraints due to the dynamics, the obstacles, and the nonholonomic nature of the mobile platforms, gives us the trajectory shown in Figure 3.b. It is interesting to note that the system shows some apparently discrete behaviors – the two mobile manipulators reconfigure from a “march-abreast” formation to a “follow-the-leader” formation, then follow a straight-line to the closest obstacle, pursue a trajectory that hugs that obstacle until they are beyond the constriction, before reconfiguring back to the “march-abreast” formation and taking the unconstrained straight-line path to the goal. These behaviors can be represented in a discrete-event framework, but they are naturally generated with a continuous method by requiring smoothness of motion.

The next example reinforces the basic idea of being able to generate discrete behaviors in a continuous framework. Consider two fingers with joint limits rotating a circular object pivoted about a fixed axis in a horizontal plane (Figure 4.a). The workspaces of the fingers are cones of angles 2α centered along the x axis and vertices at the origin, located diametrically across each other ($-\alpha \leq \theta_1 \leq \alpha$, $\pi - \alpha \leq \theta_2 \leq \pi + \alpha$). The task is to rotate the object through an angle $\Delta\varphi$, but during any finite time interval only one finger is allowed to be on the object. If there are limits on finger movement ($2\alpha < \Delta\varphi$), it is necessary to regrasp to complete the task. To guarantee the continuity of the positions and velocities, the cost functional is chosen to be the L^2 norm of the actuator torques.

Figure 4.b shows the results for the workspace $\alpha = 15^\circ$, $\Delta\varphi = 60^\circ$. During the first third of the maneuver, finger 2 rotates the object through 15° before it reaches a workspace limit. Meanwhile, finger 1 positions itself at -15° and subsequently rotates the object through almost 30° in the second stage of the task. While finger 1 is rotating

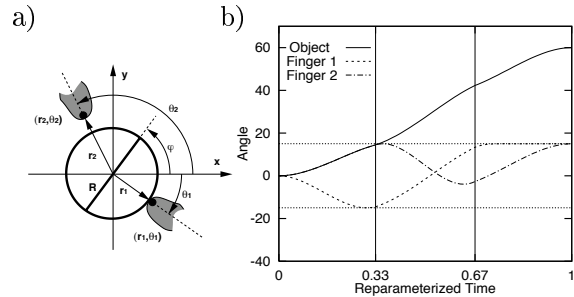


Figure 4: (a) Two fingers rotating a wheel through $\Delta\varphi = 60^\circ$. (b) Angles φ (solid), θ_1 (dashed) and $\theta_2 - \pi$ (dot-dash) for the workspace $\alpha = 15^\circ$ (dotted).

the object, finger 2 moves back towards the middle of its allowable workspace so that it can complete the rotation of the object in the third stage. During this stage, finger 1 stays at the upper edge of its workspace. The results are consistent with our intuition of what the best solution would be – both fingers move through the minimal distance necessary, but in a smooth fashion, to complete the task.

4 Planning with uncertainty

We now address the motion planning problem when only a nominal, parametric model of the environment is available, and the model may be subject to errors. We still assume a deterministic model of the robot and exact state information. However, we assume no prior distribution for the noise or uncertainty in the environment. Instead, we assume a sensor system and estimation algorithms that return, along with the measurement of each parameter in the model, a confidence interval for each parameter in which the true value lies. Examples of such confidence set-based estimation algorithms are discussed in (Kamberova et al. 1997).

Game theory provides the natural framework for solving problems with uncertainty. The motion planning problem can be formulated as a two-person zero sum game (Basar and Olsder 1982) in which the robot is a player and the obstacles and the other robots are the adversary. The goal is to find a control law that yields the best performance in the face of the worst possible uncertainty (a saddle-point strategy). The motion planning problem can be solved with open loop control laws (as we have done thus far) or with closed loop control laws (feedback policies). Rather than develop the notation and the theory that is required for the framework of game theory, we present representative examples and discuss

optimal open loop and closed loop control laws.

The approach in the previous sections for generating open-loop trajectories for deterministic systems can be extended in an obvious fashion to problems with uncertainty. The uncertainty in the environment is incorporated through conservative bounds on the feasible regions in the state space. This effectively amounts to making the obstacles bigger, reflecting the uncertainty in the position and the geometry of the obstacles⁵. With the additional sensory information that becomes available during the execution of the plan, the bounds on the feasible regions can be made less conservative and the open-loop plan can be refined accordingly. This method is attractive because our numerical method lends itself to efficient remeshing and refining and the computational cost of refining an open loop plan is an order of magnitude less than the cost of generating an initial open loop plan, when the changes in the model remain small. Thus, open loop plans may be recursively refined.

An example of this approach is demonstrated in Figure 5.a where two nonholonomic vehicles, Robot 1 and Robot 2, are to interchange their positions while moving through a narrow corridor formed by two obstacles. Each robot replans (refines) the initial open loop trajectory at the points shown by the markers. The shading of each robot in this diagram represents the time elapsed, moving from an initial dark shading (at $t = 0$) to a light shading (at $t = 1$). Neither robot knows the other's task or planned route. Each robot determines its open loop control based only on its estimate of the current position and orientation of the other robot and the obstacle. Thus, the robots change their motion plans only when the two robots are about to collide. While the refinement of the plans is locally optimal, it is clearly not globally optimal and the resulting trajectories are more expensive than needed. In this simulation, Robot 1 is given a priority over Robot 2, and so follows a path that is closer to being optimal.

While it is possible to incorporate modeling uncertainties using such approaches, they invariably lead to sub-optimal paths. Further, these paths are designed to stay away from areas that have even a very small probability of being occupied by an obstacle. There is clearly a trade-off between the conservative strategy of skirting the uncertain boundaries of the obstacle and the more aggressive strategy that incorporates bet-

ter sensory information about the obstacle as it gets closer to it. Such an aggressive strategy requires feedback control, suggesting that the motion planning should be reformulated as a search for the optimal feedback control law. In this formulation, it is necessary to concurrently consider the dynamics of the system and the problem of estimating the geometry of the environment.

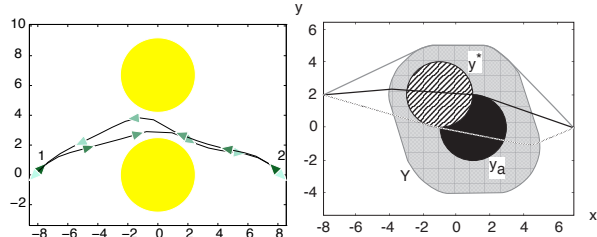


Figure 5: (a) Successive refinement of the plans for two autonomous robots A and B. (b) A comparison of the worst-case path with the optimal feedback law with that generated by the open loop control law.

A simplified but realistic problem that is mathematically tractable is discussed below. We assume a single robot and an obstacle (or obstacles) that can be observed with a sensor. The sensor estimates the position of the obstacle with some uncertainty bounds depending on the distance between the robot and the obstacle. We consider a simple model for the robot dynamics:

$$\dot{x} = u, \quad (5)$$

where the vector x is the position of the robot and u is the vector of inputs. The obstacles (including other robots) define a set of points in R^2 parameterized by a vector $y \in R^m$. The initial model of the environment is denoted by y_0 . $d(x, y)$ is a distance function whose value is the Euclidean distance between the nearest pair of points, one on an obstacle and the other on the robot. We use $\hat{y} \in R^m$ to denote the estimated obstacle(s). The basic idea is that $\hat{y} \rightarrow y$ as $d(x, y) \rightarrow 0$. An example is provided by a simple sensor model:

$$\hat{y}_i = y_i + (y_{0,i} - y_i)e^{-\beta d(x, y)^{-1}}, \quad (6)$$

where the exponential law is scaled by the parameter β so that the effect is approximately linear in an appropriate neighborhood of the obstacle, and levels off to the initial (worst-case) value further away.

For this problem, we are interested in obtaining a (static) feedback control law⁶ $u^* = u(x, y)$

⁵The resulting motion plan can be shown to be a saddle-point solution in the set of all open-loop strategies.

⁶Strictly speaking, the u is a function of x and \hat{y} .

that will minimize the traveled distance as well as ensure that the robot avoids the obstacle and reaches the goal. We can allow the robot to come arbitrarily close to the obstacle, but we want to prevent collisions. Thus the allowable feedback policies are restricted to ones for which $d(x(t), \hat{y}(t)) \geq 0$, through the time interval $[0, T]$.

In general, it is difficult to find even a sufficient feedback strategy $u(x, y)$ for the above problem (Rimon and Koditschek 1992). One way to simplify the computation is to parameterize the control law and find the optimal values for the parameters. For example, we can try to find the optimal linear feedback control law:

$$u(x, y) = A(x^d - x) + B(x - y). \quad (7)$$

The task then becomes to find the optimal values for the matrices A and B . For even simple problems it is difficult to find a feasible, linear feedback law. It is more practical to consider the set of all piecewise linear feedback laws. In order to find the optimal feedback policy, we can divide the path into discrete intervals in each of which the feedback parameters A and B are held constant. The task now is to determine the values of A and B in each time interval.

In Figure 5.b we show the motion plan when an obstacle is known to belong to a compact subset Y in R^m , but the exact location is unknown. The set Y is shown shaded, the worst obstacle location y^* is shown in black, and the actual object location $y_a \in R^m$ is the hatched circle. The figure shows three trajectories. The longest path (shown gray) is the most conservative one that would have to be taken using a purely open loop approach. The intermediate path (shown dotted) is the worst-case path that could possibly arise with the optimal feedback law. In other words, this is the path given by u^* for the worst-case object y^* . Finally, the shortest path (shown solid) is the path followed for the obstacle y_a , under the optimal feedback law, u^* .

This approach can be used to solve more complicated min-max (or inf-sup) motion planning problems. However, while the simplified model (5-6) may guarantee the existence of a saddle-point (Basar and Olsder 1982), this is not the case in more complicated situations. Even if there are saddle-point solutions, there may be many such solutions. Finally, the computational cost of generating a min-max solution is an order of magnitude higher than solving the open loop problem (essentially a single person game).

5 Conclusion

We have summarized our previous and ongoing work on motion planning⁷. This work has been motivated by studies of human manipulation that suggest existence of a repeatable optimal strategy for reaching and grasping. We have established a continuous framework that allows us to apply a similar strategy to robot motion planning and discussed how it can be extended to tasks in uncertain environments.

References

- Basar, T. and G. J. Olsder (1982). *Dynamic noncooperative game theory*. London: Academic Press.
- Desai, J., M. Žefran, and V. Kumar (1997). Manipulation tasks with friction assisted grasping. *IROS'97*, Grenoble, France.
- Flash, T. and N. Hogan (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience* 5(7), 1688–1703.
- Garvin, G. J., M. Žefran, E. A. Henis, and V. Kumar (1997). Two-arm trajectory planning in a manipulation task. *Biological Cybernetics*, 65–71.
- Kamberova, G., R. Mandelbaum, and M. Mintz (1997). Statistical decision theory for mobile robotics. Technical report, GRASP Lab., Univ. Pennsylvania.
- Kawato, M. (1990). Optimization and learning in neural networks for formation and control of coordinated movement. Tech. Rep. TR-A-0086, ATR Research Lab., Japan.
- Latombe, J.-C. (1991). *Robot motion planning*. Boston: Kluwer Academic Publishers.
- Rimon, E. and D. E. Koditschek (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation* 8(5), 501–518.
- Žefran, M., J. Desai, and V. Kumar (1996). Continuous motion plans for systems with changing dynamic behavior. Workshop on Algorithmic Foundations of Robotics.
- Žefran, M. and V. Kumar (1997). Rigid body motion interpolation. *Computer Aided Design*. Accepted for publication.

⁷We acknowledge the discussions with Max Mintz on game theory and sensor models. We also thank Jaydev Desai and Hong Zhang for providing the examples used in this paper.