MOTION PLANNING FOR MOBILE ROBOTS IN DYNAMIC ENVIRONMENTS

Jing Ren, Kenneth.A. McIsaac and Xishi Huang
University of Western Ontario
London ON, Canada N6G 1H1

Keywords: Potential field, multi-robot motion planning, stability, dynamic environment.

Abstract: In this paper, we present a motion planning technique for a multi-robot team in a complex dynamic environment. We define a cell-based navigation control law that can guide the robot team through the environment while avoiding collisions with both static and dynamic obstacles and other team members. To illustrate our techniques, we consider a robot team motion planning problem in a complex "maze" with obstacles of arbitrary shape. First, we assign potential values to a set of landmarks based on their shortest distance to the goal, and then we use a spline function to generate a potential field for the entire workspace, which is inherently free of undesired local minima. Simulation results show that robots can successfully transport materials along an optimal and collision-free path and reach the goal in a complex and dynamic maze environment. Finally we prove the derived control law is stable in all times.

1 INTRODUCTION

A central issue in mobile robotics is navigation strategy. Potential field approaches are widely used for motion planning in mobile robotics because of their simplicity and elegance. In Koditschek's basic formulation (Koditschek 1989), a scalar field comprising artificial "hills" (representing obstacles, or other robots) and "valleys" (attractive positions) in the robot's world map lead naturally to a stable path towards a "low-energy" goal position. Extensive work has been done for single robot navigation. But much less investigation is devoted to a team navigation in the dynamic and complex environment.

Although single robots can play an important role in many areas, the use of multi-robot teams has a number of potential advantages over single robot systems. A group of robots working together can accomplish the task of a complex, purpose-built system in a fraction of the time, and the built-in redundancy of having many team members leads to a more robustness and fault tolerance.

Dynamic environment is a recurring challenge in motion planning. Robots providing services in sewer systems, office buildings, supermarkets or even private homes must be able to adapt on-line to unpredictable dynamic obstacles, such as people going about their own business. For a single robot, Esposito (Esposito 2002) proposed a technique to treat unpredictable obstacles as dynamic constraints that limit the choices of feasible trajectories. In this paper, we modified and extended this technique to a robot team.

2 PROBLEM STATEMENT

We consider a team of robots operating in a complex and dynamic maze environment that is populated with static obstacles of arbitrary geometry and a number of unpredictable moving obstacles. The configuration q_i of each robot is given by the vector $q_i = (x_i, y_i)$ of the position of its center of mass. We also define $q = (q_1, q_2, ..., q_Q)$ as the state vector of the robot team, Q is the number of the robots.

The robots task is to transport materials in a maze, tracking the shortest path from any starting point to a defined goal position and avoiding collision with the environment (fixed obstacles); with their team members, and with a set of randomly moving, unpredictable dynamic obstacles.

Ren J., A. McIsaac K. and Huang X. (2004).
MOTION PLANNING FOR MOBILE ROBOTS IN DYNAMIC ENVIRONMENTS.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 361-364
DOI: 10.5220/0001128503610364

3 POTENTIAL FIELD CONSTRUCTION

Our idea of potential field construction is based on the shortest distance transform. The potential values of a set of known landmarks are first defined according to their shortest distance to the goal. Then we use a spline function based on these known potential values to generate a potential field for the entire workspace. Along with this shortest-distance-based potential field, our navigation function includes avoidance of known dynamic obstacles (other robots in the team, modelled by a generalized Gaussian function). Unknown dynamic obstacles are treated as run-time constraints on the motion plan that are only activated when obstacles are detected "near" the navigating robot.

3.1 Distance Transform

The simple example maze in Figure 1 illustrates our technique of using landmarks to map the workspace into a node network based on the distance transform. In the figure, the dark lines represent feasible paths through the maze from point A to point D. Points B1, B2, C1 and C2 are considered ***waypoints***, which essentially represent "forks" where the planning algorithm will have to make choices. In our present work, the task of generating these waypoints is left to the programmer, although we believe it will be a simple matter to automate this step.

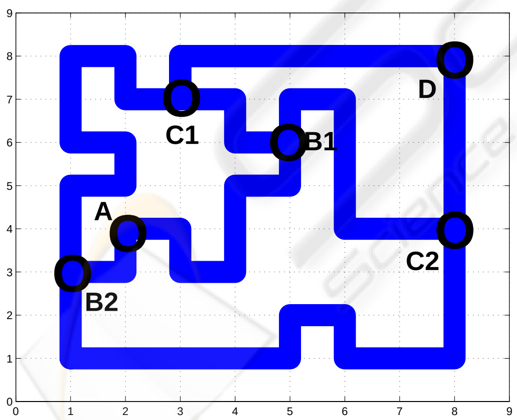

Figure 1: The original map of a simple maze. A is the starting point, and D is the goal point. B1, B2, C1 and C2 are all waypoints, where the planner must make a decision.

After finding the required set of landmarks, we can generate a node network. The equivalent network for the maze of Figure 1 is given in Figure 2. In this graph, one node is associated with each landmark, and the values of edges connecting nodes are given by the distance between landmarks. Since there are no forks between pairs of landmarks, there is a non-ambiguous value for each of these edges.

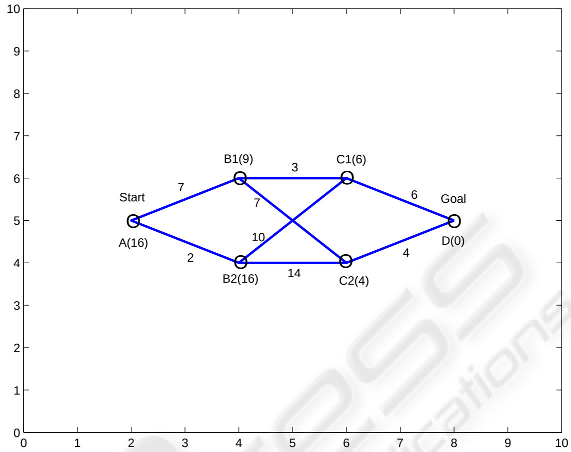

Figure 2: Node network corresponding to the workspace of Figure 1. The numbers in parentheses represent the shortest distance (potential value) of each node to the goal. The numbers on the edges represent the distance between the associated pairs of nodes.

Remark: With this model of the environment, there are various methods to find the optimal path through a graph for different applications. In our simulation, we use the techniques of dynamic programming to generate optimal path.

3.2 Construction of the Potential Field Model

In this section, we will discuss how we develop a potential field model of the workspace that incorporates both obstacles and robots.

We define a bicubic spline over the workspace. The potential values of landmarks are the shortest distances to the goal. Potential values at cell corner points that fall inside feasible regions are also the shortest distances to the goal, which is the addition of the potential values of the landmarks and the distance to those landmarks. Potential values at cell corner points that fall inside obstacles can be defined essentially arbitrarily, provided they are given a larger value than that of neighboring cell corners in feasible region.

Given any position $q = (x, y)$ in the workspace, the bicubic spline defines a potential field $VS(q)$ in cell (i, j) of the form

$$VS(q) = S_{i,j}(x, y) = \sum_{k=0}^{3} \sum_{l=0}^{3} a_{k,l}^{(i,j)} (x - x_i)^k (y - y_j)^l \quad (1)$$

where (x_i, y_j) is the position of left bottom corner point of cell (i, j), $a_{k,l}^{(i,j)}$'s are constants determined by the potential values of all the cell corners in the entire workspace. According to the spline theory, $VS(q)$ has continuous second derivative in the entire workspace.

The potential field thus created gives a smooth approximation of the optimal distance from each point in the maze to the goal. Note that although we include a "start" point in our dynamic programming analysis, our interpolating spline function gives the optimal path from *any* arbitrary starting point anywhere in the maze. Figure 3 shows the potential field for an example maze.

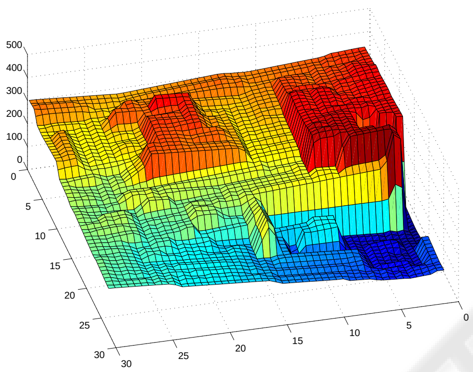

Figure 3: Potential field of an example maze. The lowest point is the goal. The potential values of points in the maze depend both on their type (obstacle or path) and also on the shortest distance to the goal. Although the user must only distinguish obstacles from safe points, we can see clearly from this plot that points of the same type take on smaller and smaller potential values as they get closer and closer to the goal.

4 MOTION PLANNING

4.1 Navigation Function

To construct the ***navigation function*** for each robot we use the two part formula:

$$V_i(q) = VS(q_i) + \sum_{j \neq i}^{j \subseteq G_i} VR(q_i, q_j) \tag{2}$$

where G_i is defined as the set of team members within the ***protective space of agent i***, $VS(q_i)$ represents the optimal potential field generated by our interpolating cubic spline, and the functions $VR(q_i, q_j)$ represent the repulsor functions between pairs of robots i and j.

$$VR(q_i, q_j) = e^{-\frac{1}{2}\left(\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{\sigma^2}\right)^C} \tag{3}$$

using the generalized Gaussian repulsor function.

4.2 Control Law without Moving Obstacles

For every system state, q, we associate with each robot, i, a control law, $u_i(q)$, of the form:

$$u_i(q) = -\alpha \frac{\frac{\partial V_i(q)}{\partial q_i}}{\left|\frac{\partial V_i(q)}{\partial q_i}\right|}, i = 1, 2, ..., Q \tag{4}$$

In Equation 4.2, the operator $\frac{\partial V_i(q)}{\partial q_i}$ represents the gradient of $V_i(q)$ with respect to only q_i.

4.3 Control Law with Moving Obstacles

For every system state, q, we associate with each robot, i, a ***feasible control generating function***, to move the robot away from nearby unmodelled obstacles while still making progress to the goal, $Z_i(q)$, of the form:

$$Z_i(q) = -(1 - \beta^2)^{\frac{1}{2}} \frac{\partial V_i(q)}{\partial q_i} + \beta \left[\frac{\partial V_i(q)}{\partial q_i}\right]^{\perp} \tag{5}$$

provided $\beta^2 < 1$. With this definition for $Z_i(q)$, we define the control law for the robots as:

$$\dot{q}_i = \alpha \frac{Z_i(q)}{|Z_i(q)|} \tag{6}$$

In this control law, the feasible control function, $Z_i(q)$, generates a unit vector direction for the robot, and the constant velocity parameter α is used to choose the robot's speed.

5 STABILITY ANALYSIS

We can define a Lyapunov function for the team of the form:

$$\begin{aligned} V^X(q) &= \sum_{i=1}^{Q} V_i^{X_i}(q) - \sum_{i=1}^{Q} \sum_{j \geq i+1}^{j \subseteq G_i} VR(q_i, q_j) && (7) \\ &= \sum_{i=1}^{Q} VS(q_i) + \sum_{i=1}^{Q} \sum_{j \geq i+1}^{j \subseteq G_i} VR(q_i, q_j) && (8) \end{aligned}$$

where the step from Equation 7 to Equation 8 is justified by the fact that $VR(q_i, q_j) = VR(q_j, q_i)$.

To show Lyapunov stability, we are required to show $V(q) \geq 0 \; \forall q$ and $\dot{V}(q) < 0 \; \forall q, t$. In Equation 8, the first term represents the potential value at the robot position, which is defined as the shortest distance to the goal and therefore is positive by construction; Actually the first part is the potential field of workspace generated from the bicubic spline function. Due to the features of our problem, we are only concerned about the relative potential difference of the positions in the workspace but not the absolute potential value of each position. Therefore in the implementation we can always add an arbitrary positive potential value to all positions and guarantee this part is positive. That the second term is also positive follows naturally from the definition of $VR(q_i, q_j)$, As a result, we have $V(q) \geq 0$ by construction.
To show that $V(q)$ is always decreasing, we begin using the form of Equation 7. For convenience, in the derivations that follow, we have replaced terms of the form $VR(q_i, q_j)$ with the short form VR_{ij}:

Thus, the second and third terms will cancel(Ren 2003), and we will have:

$$\dot{V}(q) = \sum_{i=1}^{Q} \left(\frac{\partial V_i(q)}{\partial q_i} \right)^T \dot{q}_i \tag{9}$$

If we substitute for $\dot{q}_i$ using the dynamics defined in Section 4.2, we will have:

$$\begin{aligned} \dot{V}(q) &= -\frac{\alpha}{|Z_i(q)|} \sum_{i=1}^{Q} (1-\beta^2)^{\frac{1}{2}} \left(\frac{\partial V_i(q)}{\partial q_i} \right)^T \frac{\partial V_i(q)}{\partial q_i} \\ &\quad -\frac{\alpha}{|Z_i(q)|} \sum_{i=1}^{Q} \beta \left(\frac{\partial V_i(q)}{\partial q_i} \right)^T \left(\frac{\partial V_i(q)}{\partial q_i} \right)^{\perp} \quad (10) \\ &= -\frac{\alpha}{|Z_i(q)|} \sum_{i=1}^{Q} (1-\beta^2)^{\frac{1}{2}} \left(\frac{\partial V_i(q)}{\partial q_i} \right)^T \frac{\partial V_i(q)}{\partial q_i} \end{aligned}$$

Because $\beta^2 < 1$, we will have $\dot{V}(q) < 0 \; \forall t, q$ as required.

6 SIMULATION RESULTS

In simulation, we use dynamic programming to find the shortest path to the goal from a set of known waypoints where decisions will need to be made by a path planner and the construction of a potential field based on a bicubic spline function that generates the shortest path from all points in the map with C_2 continuity.

In Figure 4, we show the results from a sample simulation of a three-robot team transporting materials in a maze. All three robots find the shortest path from their start positions to the goal, and travel the shortest path while avoiding collisions with obstacles and their fellows.

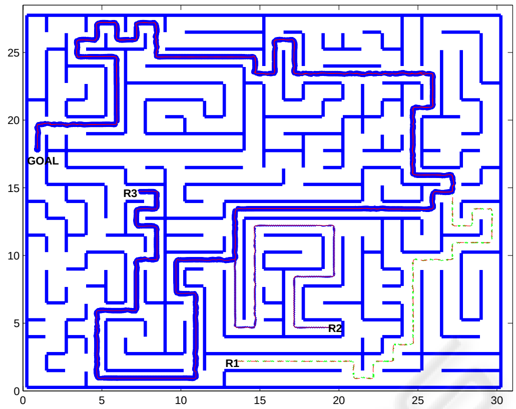

Figure 4: R1,R2,R3 are 3 robots. The different line styles represent trajectories of different robots. The three robots start in different initial positions, and all find the optimal (shortest) path to the goal, while avoiding collisions with each other.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we present a technique for an agent-based multi-robot team finding the optimal path through a complex maze in the presence of dynamic obstacles. By modifying and extending Esposito's moving strategy to a robot team, we define a control law that incorporates moving obstacles avoidance.

REFERENCES

Esposito J.M. and Kumar, V. A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at run-time. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1691–1696, Washington, May 2002.

Koditschek, D.E. Robot planning and control via potential functions. In O. Khatib, J. J. Craig, and T. Lozano-Perez, editors, *The Robotics Review 1*, pages 349–367, 1989.

J. Ren, K. A. McIsaac. A Hybrid Systems Approach to Potential Field Navigation for a Multi-Robot Team. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 3875–3880, Taipei, Sept 2003.