Distributed Path Planning for Robots in Dynamic Environments Using a Pervasive Embedded Network

Keith J. O'Hara
The BORG Lab
College of Computing
Georgia Institute of Technology
kjohara@cc.gatech.edu

Tucker R. Balch
The BORG Lab
College of Computing
Georgia Institute of Technology
tucker@cc.gatech.edu

We investigate the application of a low-cost, pervasively distributed network to plan paths for mobile robots in environments with dynamic obstacles. We consider a heterogeneous system composed of small, embedded, immobile, possibly sensor-less, communication nodes and larger mobile robots equipped with sensors and manipulators. The embedded network serves as a pervasive communication and computation fabric, while the mobile robots provide sensing and actuation. The network is responsible for planning paths for the mobile robots even though paths are being created and destroyed dynamically. The embedded network provides nearly optimal path planning without the network nodes or the robots having global knowledge or localization capabilities.

Path planning is one of the most fundamental and well-studied problem in mobile robotics. Techniques such as D*[1] plan paths in dynamic environments where paths are created and destroyed. Inspired by these techniques, we have developed a technique for distributed path planning in environments were obstacles appear and disappear. Traditional path planning algorithms for dynamic environments typically require a single mobile robot to build a map, update the map as the environment changes, and then finally plan over the map. Instead, we use an embedded network distributed throughout the environment to approximate the path-planning space and use the network to compute the path in a distributed fashion.

The algorithm essentially works as a distributed variant of the popular wave-front path planning algorithm, or a breadth-first search from the goal, propagating paths from the goal location. The embedded nodes make up the vertices of the path planning graph, and the network connections between them are the edges of the graph. Mobile robots can then use reactive navigation to traverse the graph by visiting the vertices (i.e. the embedded nodes) to the goal. In order to respond to changes in the environment this graph has to be maintained as edges are added and removed. For power-efficiency reasons we prefer algorithms that minimize communication between the embedded nodes. In particular, we want to concentrate the effort along the path the robot is taking, thus saving power without any loss in performance.

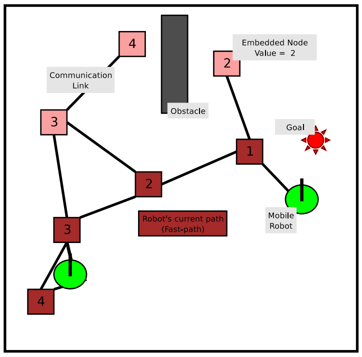

Figure 1. An illustration showing how the network guides a mobile robot to a goal location.

We assume the robots and embedded nodes communicate using a short-range medium that is occluded by the same objects that occlude navigation (e.g. walls). Line of sight between nodes implies open space for navigation. The mobile robots should be able to locally sense attractors and obstacles, and be able to estimate bearing to nearby embedded nodes.

The embedded network creates a navigation network for guiding, or routing, mobile robots in various tasks such as coverage, recruitment, and path planning. We use the network in this work strictly for path planning. In general, several navigation networks can be present in the network simultaneously. A mobile robot can then follow the navigation network corresponding to its current goal. We are able to use navigation networks to complete distributed path-planning in dynamic environments without mapping or localization.

We follow Payton's [2] virtual pheromone technique and assume the communication paths are similar to the navigation paths, and use this to propagate navigation information. By using a short-range communication medium that is occluded by obstacles to navigation the communication paths carve out free-space. As also pointed out by Payton [2], Li [3], and Batalin [4] this results in a kind of distributed physical path-planning.

To create a navigation network for a particular goal we use a distributed dynamic programming approach; specifically, a variation of the distributed Bellman-Ford algorithm. Bellman-Ford can be used to find the shortest path to a destination from all nodes. The distributed version of Bellman-Ford was created for network routing protocols. In the distributed network routing version, neighbors share their path costs and the distance between nodes is usually measured in hops. We use distributed Bellman-Ford to effectively create a tree of shortest paths from every node to the goal – this tree is the navigation network. The embedded network can be thought of as "routing" the mobile robots to their destination. However, note that the embedded nodes do not know the position of their neighbors, so they are not directing the robot in any direction. The embedded nodes can be thought of as accomplishing a type of passive routing, because, although they provide a path cost, the mobile robots make the decision where to go next.

As the mobile robots discover attractors, they broadcast this information to neighboring embedded nodes and navigation trees are created with roots near the attractor. As the information is propagated throughout the network by the embedded nodes, the hop-count, or path-cost, increases. Any mobile robot can then approach the network, access the relevant navigation network, and descend to the root of the tree, eventually reaching the original goal.

One key requirement for using navigation networks in dynamic environments is the ability of the network to respond to changes in the environment. The nodes must keep track of their neighbors and act appropriately when changes occur. One approach would be for every node to constantly monitor the status of its neighbors. Although this approach would result in speedy reconfiguration, it is also wasteful since it doesn't focus the communication on any part of the system. Instead, we would prefer the nodes along the path of the robot to monitor their neighbors often, but have the other nodes update less frequently (or not at all). Once the path has been initialized, certain nodes are designated to be on the *fast-path*, meaning they should monitor the status of their neighbors more frequently using hearbeat messages.

Two routines run in parallel on the embedded nodes - the routine responsible for receiving incoming messages and another for sending messages to neighbors. When a message is to be sent, the node will propagate a path-cost value of 1 plus the minimum path-cost value of all its alive neighbors. When an embedded node, A, receives a message it updates a data-structure that keeps track of the status of its neighbors. If this neighbor is a mobile robot then the node is on the fast-path. Also, if a node's neighbor, B, is on the fast-path and B is A's child (i.e. B's hop-count is 1 more than A's hop-count) then A is also on the fast-path. The fast-path indicates the path the robot is currently taking. Nodes on the fast-path send heartbeat messages frequently to monitor their neighbors.

A node will broadcast a message if any of the following four conditions are met. One, if a mobile robot is near, two, its heartbeat timeout has expired, three, its path-cost to the goal has changed, or four, a heartbeat request by its child node has been sent. The last condition allows for children to make sure the connection to their parent still alive. One key idea is that the fast-path changes, and propagates, when paths are destroyed or created. Because of this, we can have nodes not on the fast-path have a very high (e.g. ∞) heartbeat timeout.

We developed and analyzed two different techniques for distributed path planning when the environment is dynamic and paths are destroyed and created. One uses global monitoring and the other focuses communication using the fast-path idea. Both techniques are able to repair the plan when the environment changes and provide paths for a mobile robot to reach a goal. The first technique is able to respond to changes in the environment very quickly but does this at high communication cost. The second approach is able to respond to changes in the environment at the same speed, but with far fewer messages because it concentrates the messages along the path the robot currently resides.

The approach also has some notable limitations. For one, the system relies on the assumption that communication paths are similar to navigation paths. If this is not true then the embedded nodes will have to be more capable and sense obstacles directly or incorporate the robots' real navigation experiences.

References

[1] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1994.

[2] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone Robotics," *Autonomous Robots*, vol. 11, pp. 319–324, 2001.

[3] Q. Li, M. DeRosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, 2003.

[4] M. Batalin and G. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, 2003.