EgressBug: A Real Time Path Planning Algorithm for a Mobile Robot in an Unknown Environment

K.R. Guruprasad*

Department of Mechanical Engineering
National Institute of Technology Karnataka, Surathkal
Mangalore 575025, India
krgprao@gmail.com

Abstract. This paper addresses the problem of path planning for a mobile robot in a region occupied by finite number obstacles. The region and the obstacles are not known *a priori* to the robot. We present an online path planning algorithm called EgressBug, that makes the robot reach any specified point in the space, and stops by reporting failure when the specified goal point does not belong to the free space. The proposed EgressBug algorithm uses simple move toward a point and wall following behaviors. The algorithm is illustrated with the help of examples, and paths generated by the EgressBug algorithm are compared with those generated by the Bug2 and TangentBug algorithms.

Keywords: Path planning, Bug algorithms, mobile robot.

1 Introduction

The problem of path planning for a mobile autonomous robot to reach a specified goal point has attracted many researchers in the past. Off-line path planning algorithms rely on complete a priori information about the region of interest. Some of the off-line approaches such as visibility graph and tangent graph[1] based algorithms generate an optimal (shortest) path from any start location to any specified goal point, using search algorithms. However, such optimality cannot be ensured by a path planning algorithm, when the region is occupied by obstacles which are not known to the robot a priori.

Several approaches have been proposed in the literature in the past to solve the path planning problem in an unknown region. One of the widely used scheme is use of artificial potential functions where the goal point globally attracts, while the obstacles repel the robot [2, 3]. One of the problems associated with artificial potential function based approaches is that the robot may get stuck in a local minimum and fail to reach the goal point. Several methods are discussed in the literature to overcome this problem.

* K.R. Guruprasad is Currently visiting the Department of Computer Science, University of Nebraska at Omaha, Omaha, NE 68182, USA.

P.S. Thilagam et al. (Eds.): ADCONS 2011, LNCS 7135, pp. 228–236, 2012.

Another interesting class of sensor-based path planning algorithms extensively discussed in the literature is the family of 'Bug algorithms'. These algorithms use simple behaviors such as wall following and moving toward a point using sensors to detect obstacles. Two algorithms namely Bug1 and Bug2 were proposed by Lumelsky et al. [4]. Several versions of Bug algorithms have been proposed since then [5–10]. Ng and Bräunl [11] provide a comparison of several Bug algorithms. In TangentBug algorithm [8], the robot makes locally optimal decisions. However, it is well known that the global optimality cannot be assured with locally optimal decisions.

In this paper, we propose an online path planning algorithm called EgressBug, to make the robot reach a specified goal point from a given start location. If no path from a given start location to a specified goal location exists, then the EgressBug algorithm stops reporting failure. We use simple behaviors such as wall following and moving toward the goal, which require minimal computational effort.

2 EgressBug Path Planning Algorithm

We consider a possibly unbounded space $Q \subset \mathbb{R}^2$ which is occupied by a set of bounded obstacles $O = \{O_1, O_2, \ldots, O_K\}$. Neither Q nor obstacles are necessarily known *a priori*. We consider a wheeled robot which is equipped with sensors to detect obstacles. The robot is capable of localizing itself with reference to a global frame of reference. We solve the problem in the configuration space [12] where the robot is represented as a point. The position and heading direction of the robot at time t are $\mathbf{X}(t)$ and $\phi(t)$ respectively. We define Q_{free} as the connected component of $Q \setminus O$ containing a point $S = X(0)$. For a vector A, we use the notation $\hat{A}$ to denote a unit vector along A.

Table 1. Basic behaviors of the robot

Behavior/state	**Description**
Start	Initialize
Move toward target (MT)	Move toward a target point G at a constant speed $v > 0$.
Wall follow left (WFL) or Wall follow right (WFR)	Follow a wall, on left or right side, at a constant speed $v > 0$.
End with success	Goal reached, that is, $X(t) = G$. Stop.
End reporting failure	Goal cannot be reached. Stop.

We solve the path planning problem in two steps. First, we present a simple algorithm, namely, EgressBug1 algorithm, which will attempt to make the robot reach a specified goal point G. Second, we present an EgressLoop algorithm, that makes the robot egress from the loop it might have encountered using the EgressBug1 algorithm. The EgressBug1 and EgressLoop algorithms are then combined