Robotics and Autonomous Systems 57 (2009) 877–888

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A sub goal seeking approach for reactive navigation in complex unknown environments

Chen Ye*, Phil Webb

School of Mechanical, Materials and Manufacturing Engineering, The University of Nottingham, University Park, Nottingham, NG7 2RD, United Kingdom

ARTICLE INFO

Article history:
Received 10 April 2008
Received in revised form
17 June 2009
Accepted 22 June 2009
Available online 7 July 2009

Keywords:
Mobile robot
Navigation
Unknown environment
Reactive navigation
Local minimum

ABSTRACT

Reactive-based approaches are widely used in autonomous navigation. However, in complex unknown environments, pure reactive-based navigation still poses a few challenges since it can be easily trapped by a local minimum and may produce some extra manoeuvres. This paper presents the design of a reactive-based approach for navigation in complex and unknown environments called sub goal seeking, in which depth point maps of the environment are analysed to extract free spaces around the robot. These spaces are then evaluated the one that is most likely to lead to the final goal is chosen as a sub goal. The robot then drives towards these sub goals, instead of the final goal until it is visible. By analysing the environmental structure, dead-ends within robot sensory range are able to be detected thus reducing the chance of being trapped and also reducing unnecessary manoeuvres. This paper also evaluates the performance of the sub goal seeking approach using three criteria, goal achievable ability, safety and maneuvering through extensive simulation and real mobile robot experiments.

1. Introduction

The main aim of any research on reactive navigation systems is to guide a mobile robot moving freely in unknown environments. Without being given environmental knowledge, the robot has to deal with unforeseeable circumstances using a reactive mechanism. Sensor noise, imprecise control and inaccurate localisation information make the mission more difficult to achieve. Theoretically, an ideal reactive system should be able to navigate a robot safely in the presence of any number of uncertainties and produce a high speed fluid motion.

Over the last few decades, efforts to develop such an ideal system have resulted in a number of successful approaches. The widely used artificial potential field approach [1] provides an elegant solution to the navigation problem, in which obstacles assert repulsive forces on the robot while the target asserts an attractive force. The strength of the forces is relative to the distance and orientation of the obstacles. The vector sum of these forces is used to drive the robot, thus avoiding obstacles while still keeping a track on the on the target. In simple environments, the artificial potential field approach has proved to be successful. However, some inherent limitations have been discovered, such as not passing between closely spaced obstacles, oscillation in narrow corridors and local minima [2]. Although some solutions to these limitations have been proposed [2–6], its ability to control navigation in complex environments is still limited.

The vector field histogram is another popular approach developed by Borenstein in 1991 to overcome the limitations of the artificial potential field [7]. The vector field histogram (VFH) works by manipulating a histogram created from a local occupancy grid map of the environment around the robot. Within the histogram, those openings large enough for the vehicle to pass through are identified, and a cost function is applied to every candidate opening and the opening with the lowest cost is chosen. The travel direction is then generated dependent upon the chosen opening. The VFH approach successfully overcomes some of the limitations of potential field. Its enhanced version VFH+ [8] also takes into account a simplified model of the moving robot's possible trajectories based on its kinematic limitations and thus reduces the risk of collision.

A pure collision avoidance approach called a dynamic window (DW) is also a very successful approach which can generate a smooth trajectory by considering the vehicle dynamics [9,10]. Instead of choosing the travel direction; the dynamic window selects the motion commands in velocity space. The robot's trajectory is supposed to consist of several curves; each curve is uniquely determined by the velocity vector. Obstacles are mapped onto a grid map and considered to impose restrictions on the rotational and translational velocities. Only those velocity sets that ensure that the robot can come to a stop before hitting an obstacle are considered. These velocities are called admissible velocities.

* Corresponding author. Tel.: +44 (0) 115 951 4034; fax: +44 (0) 115 951 3800.
E-mail address: chen.ye@nottingham.ac.uk (C. Ye).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.06.009

A dynamic window restricts the admissible velocities to those that can be reached within the next time interval according to the robot's acceleration capability. The dynamic window approach greatly increases the robot's obstacle avoidance performance, but expensive computation is the tradeoff.

Although these approaches achieved great successes in a number of applications, they are still challenged by complex and cluttered environments in which concave shaped structures can trap the robot and stop it achieving its goal. Some solutions have been proposed recently to solve this problem such as virtual obstacle approach [11] and virtual target approach [12]. However, these approaches make an empirical evaluation for trap situations before invoking a suitable strategy and may still get trapped in unforeseen situations. The complete solution to the local minimum problem is to implement global path planning. However, a high resolution environmental map is hard to build which poses the requirement for the capability of a navigation system to avoid or recover from trap situations on a small scale, and therefore increase the possibility to accomplish a navigational task within unknown environments. For practical reasons, these approaches experience difficulties when deployed into different environments. Also, most approaches have a few parameters which dominate the system performance and the tuning of these parameters is a time consuming operation. A finely tuned pair of parameters under one environment may fail or not be efficient when deployed into another environment [13]. An adaptive capability would make the system easy to deploy in varying environments.

This paper introduces a new sub goal seeking approach by which the chance of being trapped due to local minima is reduced. It is also able to drive a robot in complex environments without oscillation by analyzing the depth map of the environment and adapt to various environments from cluttered to open without any parameter tuning. Its design is presented in Section 2, Section 3 implements the approach and demonstrates the results and Section 4 provides conclusions and discussion.

2. Sub goal seeking reactive navigation

The sub goal seeking navigation system [14] described in this paper was implemented using a Brook's behaviour architecture [15]. It consisted of three behaviours, move to goal, sub goal seeking and collision avoidance. The architecture is illustrated in Fig. 1. In this case a Laser range finder was used to provide a depth map of the robot's environment as is described further in Section 3. In the sub goal seeking approach instead of heading toward the final goal the robot heads towards a series of visible sub goals generated by the sub goal seeking behavior. The sub goal seeking behavior analyses the depth point maps of the environment from a laser range sensor, identifies the gaps (free space) around the robot and evaluates these gaps to select the gap direction as the sub goal which is most likely to lead to the final goal. By iterating this procedure, a series of sub goals are generated which lead to the final goal. The result of the move to goal behaviour is used to evaluate the cost of sub goals. The collision avoidance behaviour is designed to protect the robot from collision when an object is within a predefined safe zone.

2.1. Sub goal seeking behaviour

The sub goal seeking behavior was designed to dynamically generate a temporal sub goal that is visible to the robot at every instance and was implemented using the 3 steps shown in Fig. 2. The process iterates until the robot reaches the final goal:

Step 1: *Identify all gaps (free space) around the robot at the current position and evaluate every gap to check whether it is passable to the robot*

Step 2: *Select one of the gaps as a sub goal based on a cost function*

Step 3: *Calculate a safe turning angle and move toward the sub goal.*

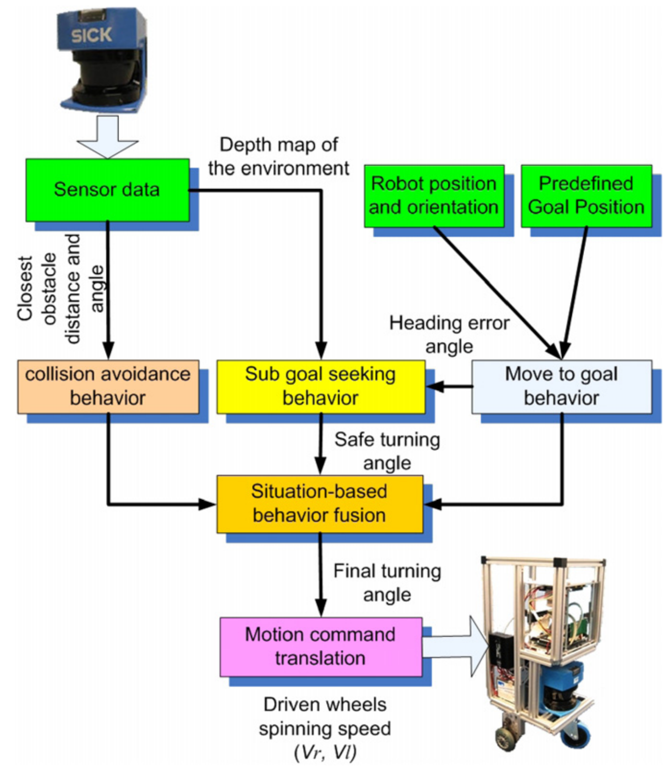

Fig. 1. Architecture of sub goal seeking approach.

2.1.1. Step 1—Identify gap

The first step is to identify all the gaps around the robot. The basic process used is to check the width of a gap and compare it with the width of the robot. Those widths smaller than the size of the robot are not identified. The sensory information is available as depth point maps. The minimum beam number n_{Min} required for a valid gap is decided by (Eq. (1)):

$$n_{\text{Min}} = \frac{1}{\theta} \times \arccos \frac{2D_S^2 - W_R^2}{2D_S^2} \tag{1}$$

where θ is the sensor angular resolution which is 1° for the laser scanner; D° is the beam point list related to the distance between obstacles and the robot perceived by the sensors, D_S is the specified detecting range, W_R is the width of the robot (Fig. 3).

Given a set of continuous distance readings $\{(D_i^{\circ} \dots D_{i+n_G}^{\circ}) | D_i^{\circ} \in D^{\circ}\}$ with each $D_i^{\circ} > D_S$, if the continuous beam number $n_G > n_{\text{Min}}$, then a gap is identified. Fig. 4 illustrates a sample with Gap II and Gap III identified. Although the gap width has been compared with the robot's width, those identified gaps may still not be passable by the robot. In Fig. 4, GAP III's width meets the requirement for a valid gap. However, it is not passable to the robot. It is therefore necessary to check whether the identified gaps allow the robot to pass. A passable gap is evaluated by checking two safe angles between the gap edges and the robot margin. In Fig. 5 θ_{LS} and θ_{RS} are considered as two safe angles that will keep the robot's edges clear of collision.

θ_{LS} is an angle that makes the robot left margin out of collision if the robot performs a θ_{LS} turn. θ_{RS} is an angle that makes that robot right margin out of collision if the robot performs a θ_{RS} turn. The values of angles θ_{LS} and θ_{RS} are obtained using the following process. Given the ith obstacle point coordination (x_{oi}, y_{oi}) and robot boundary coordinates (x_L, y_L), (x_R, y_R), an angle θ_i between the obstacle point and robot boundaries can be calculated by

$$\theta_i = \tan^{-1} \frac{x_{oi} - x_{(L,R)}}{y_{oi} - y_{(L,R)}}. \tag{2}$$