A PHYSICALLY GROUNDED SEARCH IN A BEHAVIOUR BASED ROBOT

GORDON CHENG and ALEXANDER ZELINSKY

Intelligent Robotics Laboratory
Department of Computer Science
University of Wollongong
Wollongong NSW 2522, Australia.
email: gordon@cs.uow.edu.au or alex@cs.uow.edu.au
Phone: (042) 213 773 Fax: (042) 213 846

Abstract

In this paper we present a theoretical analysis of a behaviour-based navigation system that we have developed for autonomous mobile robots. The robot navigates in an unknown environment, and avoids obstacles. While behaviour-based systems for robotics are currently being chosen over the well-established, traditional AI approaches, no well-defined theoretical foundation has been presented to support such systems. One of their most common problems is that they tend to suffer the problem of competing behaviours. Our work shows that the system we have developed possesses the property of completeness while performing a physically grounded search. We show how the navigation problem can be expressed in a mathematical manner, from which making improvements is substantially easier. We present experimental results of a real robot navigating in complex environment with our system.

Keywords

Mobile Robotics, Behaviour-based, Navigation, Collision avoidance

Introduction

Behaviour-based [Brooks 86] approaches in robotics are being used extensively as an alternative to AI approaches. Behaviours are normally engineered by trial and error, and complete their tasks without a proper model from which to work. For example, consider what happens when a robot comes upon an obstacle while it is navigating. An unresolvable state may eventuate from the competing move-to-goal behaviour and obstacle-avoidance behaviour. Behaviour-based systems can get into a deadlocked state between competing behaviours; this is an undesirable effect of these systems. Some of the techniques for solving such a problem are to introduce randomness [Arkin 92], or relocate the goal [Adams 90], but with those techniques, the robot does not benefit from knowledge gained from failure. Our work has mainly been concerned with improving the behaviour-based approach. In one of our approaches we introduce a Purposive Map (PM) [Zelinsky 95]. The PM is used as a resource to coordinate the arbitration of behaviours; we found that by using a PM, knowledge can be acquired, and the system becomes more robust.

Little theoretical work has been done on behaviour-based approaches. Few attempts have been made to show how behaviours will perform, and how behaviours can be improved. Some strategies have been formally stated [Lumelsky 87]. We are working with autonomous mobile robots navigating in an unknown environment, with an arbitrary number of obstacles. In our approach, the autonomous mobile robot would only require four simple navigational behaviours: Free-Space-Move; Move-to-Goal; Avoid-obstacles; and Contour-Following. These behaviours use the PM as a resource to accomplish its task. Further details about these behaviours will be given later. Our autonomous mobile robot will be guided by the potential fields until either a local minimum has been encountered, or the test for reachability becomes true. Consider a farmer travelling from one paddock to another. If he comes to a high, long fence. To find a gate, which way should he start walking? Also and how far should he go before he turns back and tries going in the other direction? At this point, the farmer will possess some of the same behaviours as our autonomous mobile robot.

Potential field methods are used for collision avoidance in robot navigation [Khatib 86]. However, their main aim is to be real-time efficient, rather than on reaching the final target. The problem with potential field methods is that they can be trapped in local minima. One of the methods for solving the local-minima problem works by defining a potential function with no or few local minima; other methods use search techniques. Such search techniques will be the main focus of this paper.

The local minima are defined as those points where the sum of the attractive potential force and the repulsion potential force is equal to zero. There are two possible types of local minima: minima in which the target is blocked; and minima in which the target is in free space. These types of local minima are shown in Figure 1.

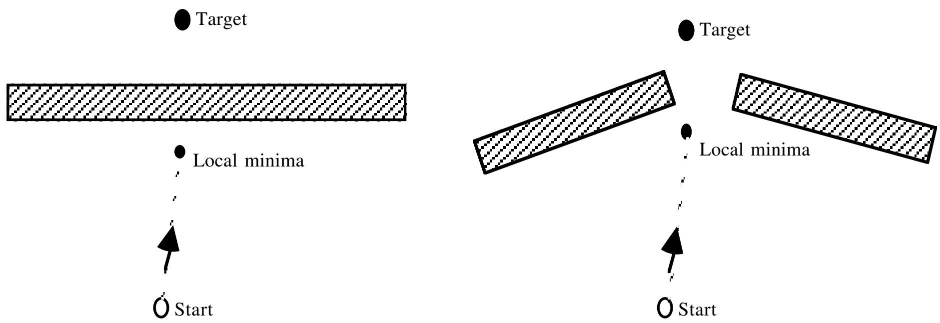

Figure 1

We will show that our approaches are mathematically sound and provide a complete search of the solution space. Being able to define the behaviour mathematically, it will be substantially easier to enhance behaviour. Also, other similar behaviours can be easily compared.

The navigation behaviours of an autonomous mobile robot

The autonomous mobile robot in this study is able to rotate through 360 degrees. The robot houses a number of distance sensors, which help it to sense surrounding obstacles. The system will be monitored and coordinated with the aid of a PM. The autonomous mobile robot possesses the three navigational behaviours that we mentioned previously. It also has knowledge of its current and previous locations.

The three basic behaviours are as follows:

Free-Space-Move behaviour: if the robot can sense free space directly in front, it moves forward toward the free space for some period of time. This is very human like; if we saw a free space in the direction in which we wanted to move, we would proceed forward.

Potential Field Methods: these will be used to guide the robot. This is obtained by summing the total attractive and repulsive potentials. Further details of potential field methods can be found in [Latombe 91]. The use of potential field methods will allow the robot to possess two sub-behaviours: Move-to-Goal and Avoid-Obstacles.

Contour-Following behaviour will guide our autonomous mobile robot in either direction about an obstacle, regardless of the shape of the obstacle. The test for reachability and the escaping technique form part of this behaviour. The escaping technique will not only a escape from behind a obstacle, but it will also be able to find an opening. The farmer would be able use this technique, as well as our robot. The first technique has been experimentally demonstrated by [Zelinsky 95].

The environment

The environment contains a Start and a Target. The task given to our autonomous mobile robot is to move from the start to the target, if a continuous path exists. The environment also contains an arbitrary number of obstacles that can be either of a convex or concave shape. These obstacles may be connected (very much like a real world environment).

The avoidance scheme

Procedure:

1. Calculate the potential forces by summing all attractive and repulsive potentials, then move toward the target using the potential forces until:
   (a) the total potential force becomes zero. Check:
      (i) if the Target is reached, the robot stops.
      (ii) if a local minimum has been encountered, m_i is defined; Goto step 2.
2. Check for free space. If free space is ahead, the robot moves forward. Otherwise Do step 3.
3. Start contour following:
   (a) Define a line segment (m_i, o_i) between the local minimum m_i and the object o_i. This line segment is used to test for target reachability.
   (b) Contour-Following : the robot starts in one direction (eg. counter clockwise) for a set distance *n*, and moves forward until either the distance *n* is reached, or it escapes. If it does not escape before the distance *n* is reached, the robot is released, as in step 1. If it returns to the local minimum, and repeats the process in the other direction. If it does not escape, it will again return to the minimum. Now the robot will have travelled in both directions. *n* will then be doubled, and the whole process repeated. Testing for reachability is done while the robot is travelling in each iteration. The basic motion of our robot is shown in Figure 2.

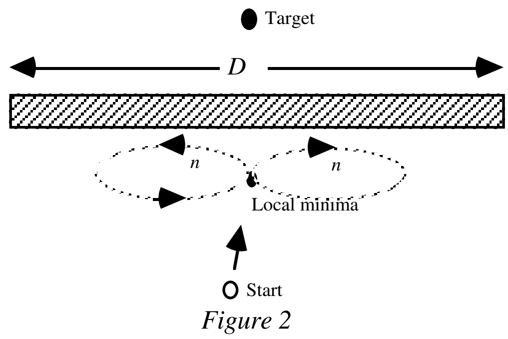

Figure 2

It is clear from this procedure that the most of the execution time will be spent escaping from an obstacle. This is where our analysis begins.

Analysis:
The total length of the path produced by this escaping technique should never exceed the limit of

$$(1) \qquad \rho = \begin{Bmatrix} \frac{5D}{2} - 4n & m > 0 \\ 3n & m \le 0 \end{Bmatrix} \text{ where } m = \left\lceil \log_2\left(\frac{D}{2n}\right) \right\rceil$$

where *D* is the total facing length of the obstacle, *n* is the trial distance before returning to the minimum, and *m* is defined as the number of turns required to escape.

Proof:
To show this, consider moving in either direction. The worst case would be to start from the middle of the facing obstacle; the most number of trials would be required. At most the distance *n* required to escape *D* would be $\frac{D}{2}$ in one single step.

A sequence of trials will have occurred, of the form $n, n, 2n, 2n, 4n, 4n, \ldots$. This can be expressed as $2^i \cdot n$. With both directions being explored, we rewrite this as $2 \times (n, n, 2n, 2n, 4n, 4n, \ldots)$ and $4 \times 2^i \cdot n$. This would form the series $4n + 8n + 16n + 32n, \ldots$.

Therefore we can write,

$$(2) \qquad 4\sum_{i=0}^{m-1} 2^i \cdot n$$

This represents the total distance travelled, in *m* steps, before escaping. And the *mth* step would not need to exceed $\frac{D}{2}$.

From these sequences, we can see that the term *m* can be derived from:

$$(3) \qquad 2^m \cdot n \ge \frac{D}{2}$$

$$(4) \qquad \therefore m = \left\lceil \log_2\left(\frac{D}{2n}\right) \right\rceil$$

From this we can be write:

$$(5) \qquad \rho = 4n\sum_{i=0}^{m-1} 2^i + \frac{D}{2}$$

And hence:

$$(6) \qquad \rho = \begin{Bmatrix} 4n\sum_{i=0}^{m-1} 2^i + \frac{D}{2} & m > 0 \\ 3n & m \le 0 \end{Bmatrix}$$

Substituting for the summation, we can rewrite this as:

$$(7) \qquad \rho = 4n(2^m - 1) + \frac{D}{2}$$

By substituting (4) into (7), we will get (1).

If $m > 0$ then n must be less than $\frac{D}{2}$. Therefore equation (7) will be used.

If $m \le 0$ then $\rho = 3n$, and n is greater than or equal to $\frac{D}{2}$. When this condition holds, the best case would be $1n$; otherwise, if the robot starts in the longer direction of D, the worst case is $3n$.

Improving the behaviour

One method of improving this behaviour would be to double n each time the same local minimum is reached. This is illustrated as follows.

From our previous analysis, we can see that by altering the search scheme, substantial improvements can be made. By changing 3 (b) in the previous procedure:

(b) Contour-Following : the robot starts in one direction (eg. counter clockwise) for a set distance n, and moves forward until either the distance n is reached, or it escapes. If it does not escape before the distance n is reached, it returns to the local minimum, and repeats the process in the other direction, for distance $2n$. On each return to the local minimum the distance is to be doubled. The same test for reachability is done while the robot is travelling in each iteration. The basic motion of our robot is shown in Figure 3.

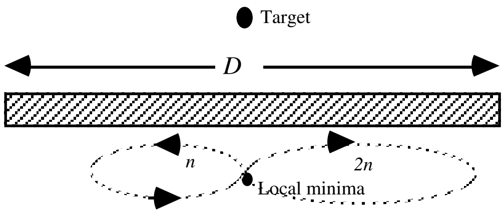

Figure 3

Analysis:

The length of the path produced by this escaping technique should never exceed the limit of

$$(8) \qquad \rho = \begin{cases} \frac{3D}{2} - 2n & m > 0 \\ 3n & m \le 0 \end{cases} \text{ where } m = \left\lceil \log_2\left(\frac{D}{2n}\right) \right\rceil$$

where D is the total facing length of the obstacle, n is the trial distance for each turn. m is defined as the number of turns required to escape.

Proof:

To show this, consider moving in either direction. The worst case would be to start from the middle of the facing obstacle; the largest number of trials will be required. At most the distance n required to escape D would be $\frac{D}{2}$ in one single step.

A sequence of trials will have occurred, of the form $n, 2n, 4n, 8n, \ldots$. This can be expressed as $2^i \cdot n$. With both directions being explored, we rewrite this as $2n, 4n, 8n, \ldots$. This forms the series $2 \times (n + 2n + 4n + 8n + \ldots)$.

Therefore we can write:

$$(9) \qquad 2\sum_{i=0}^{m-1} 2^i \cdot n$$

Which in turn is:

$$(10) \qquad \rho = 2n(2^m - 1) + \frac{D}{2}$$

This will, in turn, give us equation (8) by substitution.

From this second proof we see no reason why improvment would not be gained through incresing the multiple amount from double to a higher value on each return of the same local minimum. A probability study would be ideal in determining this value, but is beyond the scope of this paper.

Testing for reachability

Testing for reachability provides a way of preventing the robot from over-searching, if no solution exists. By determining that no solution exists, resources will not be wasted (eg. battery power). Whenever a local minimum occurs, m_i is defined for the i_{th}, i=1,2,3,4... for each local minimum the robot encountered. If, for m_i, i is 1 and a local minimum occurs. This can only mean that either no obstacle was encountered while travelling from the Start to the Target, or the target cannot be reached.

Procedure:
Test for reachability: when the local minimum is reached, a line segment from this local minimum m_i to the obstacle is defined. This line is normally determined by the sensor facing the obstacle. While the robot is trying to escape this local minimum, if it crosses this line segment, it can only mean that at each point on its path, the robot will cross the surface of the obstacle if it tries to move towards the target. Therefore, the target is unreachable. Consider the two extreme cases: the target being inside a closed room; and the target being outside a closed room. This is illustrated in Figure 4.

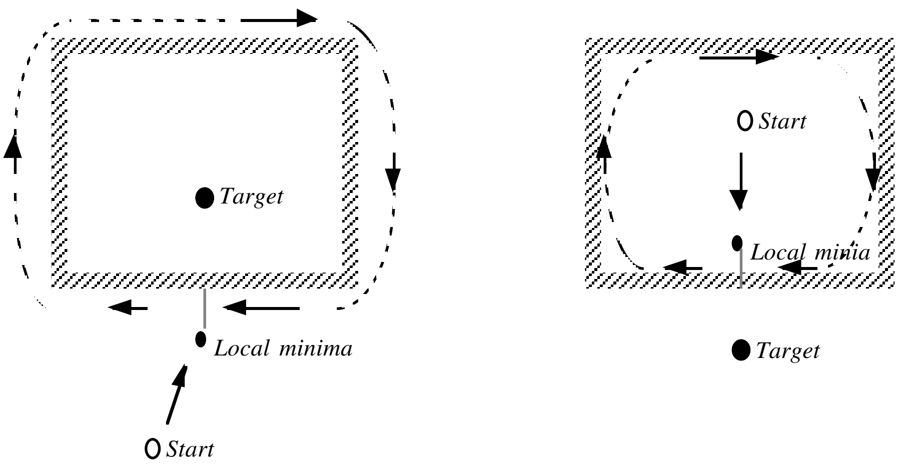

Figure 4

Figure 5 shows the motion that the robot would take inside a maze, and how the test for reachability will still apply.

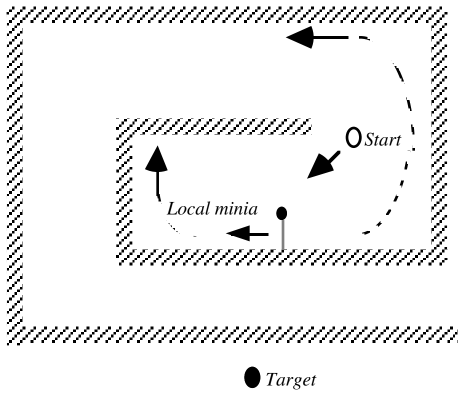

Figure 5

Testing for line intersection is shown as:

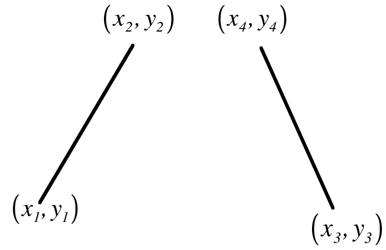

Figure 6

$$\lambda = \frac{(x_3 - x_1)(y_3 - y_4) - (y_3 - y_1)(x_3 - x_4)}{(x_2 - x_1)(y_3 - x_4) - (x_3 - y_1)(x_3 - x_4)} \qquad (11)$$

When $0 \le \lambda \le 1$, the robot has crossed the line segment (m_i, o_i). If $\lambda = 0$ the robot crossed at the local minimum. If $\lambda = 1$ the robot crossed at the obstacle (this could never happen physically). This could be useful for adjusting sensor errors. Other information can also be obtained from λ. For example: if λ becomes ∞, the robot is moving parallel to the line segment (m_i, o_i). If $\lambda > 1$ the robot's movement will intersect with the projection of the line segment (m_i, o_i) further on. If $\lambda < 0$ the robot's movement intersected with line segment (m_i, o_i) further back. This information may be used for determining the direction of the robot's movement.

Results

Our implementation of this scheme used a Nomad200 robot simulator navigating in an unknown environment. The Nomad robot was able to perform the same experiments in a similar fashion. In most of the experiments the value of n is normally set to the maximum sensor range. The test for reachability is done with the assistance of the Nomad's global coordinate system. Figure 7 shows a Nomad200 escaping from a wall of boxes.

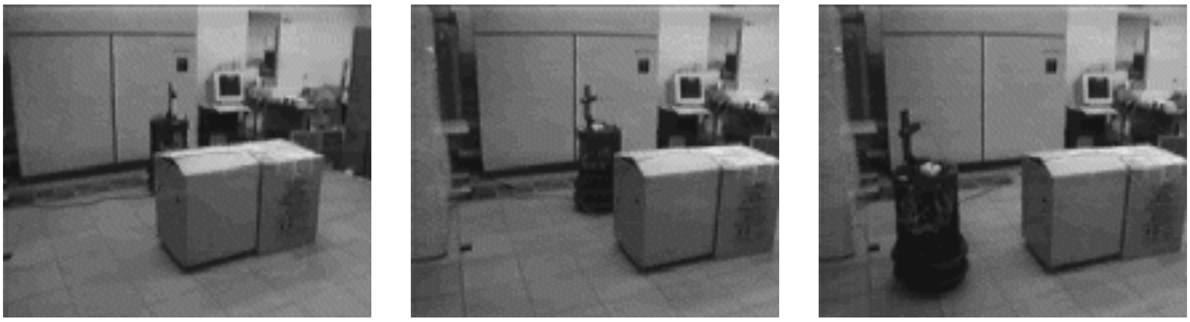

Figure 7

Conclusion

This paper has presented a method for analysing behaviour-based schemes for navigation in an unknown environment. As far as we are aware, only [Lumelsky 87] has formally attempted to state navigation algorithms for unknown environments in a similar manner. This paper has shown that the behaviours we have developed are mathematically sound and contain the property of completeness (ie. if there is a solution, it will be found, otherwise it will terminate). The behaviours can be expressed in a mathematical form to allow further analysis and improvement. We also believe that this analysis can easily be applied to other behaviours.

From a practical viewpoint, a system that can determine completeness means that resources will not be wasted (eg. the battery power of the robot).

Further studies could improve the avoidance scheme - for example, by using a Fibonacci sequence. Also, a probability study can be undertaken to determine the best *nth* terms from which to start the search.

References

[Adams 90] MD .Adams, H Hu and PJ Probert,"Towards a Real-Time Architecture for Obstacle Avoidance and Path Planning in Mobile Robots", Proceedings of IEEE on Robotics and Automation Conference, pp 584-589, 1990.

[Anderson 90] TL Anderson and M Donath, "Animal Behaviour as a Paradigm for Developing Robot Autonomy", Robotics and Autonomous Systems, vol. 6 pp 146-168 1990.

[Arkin 92] RC Arkin, RJ Clark and A Ram, "Learning Momentum: On-line Performance Enhancement for Reactive Systems", Proceedings of IEEE Robotics and Automation Conference, pp 111-116, 1992.

[Brooks 86] RA Brooks, "A Layered Intelligent Control System for a Mobile Robot", IEEE Trans. on Robotics and Automation, RA-2, April 1986.

[Khatib 86] O Khatib , “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, International journal of Robotics Research, 5(1), pp 90-98, 1986.

[Latombe 91] J-C Latombe, “Robot motion planning”, Boston: Kluwer Academic Publishers, c1991.

[Lumelsky 87] VJ Lumelsky and AA Stepanov, “Path-Planning Strategies for a Point Mobile Automation Moving Amidst Unknown Obstacles of Arbitrary Shape”, Algorithmica, Special Issue: Robotics. pp 403-430, Volume 2 1987.

[McDermott 92] D McDermott, “Robot Planning”. pp 55-79, AI Magazine SUMMER 1992.

[Zelinsky 95] A Zelinsky, T Suehiro and H Tsukune, “Using Augmentable Resource to Robustly and Purposefully Navigate a Robot”, , Proceedings of IEEE Robotics and Automation Conference, pp 2586-2592, Volume 3 1995.