INTEGRATING A POTENTIAL FIELD BASED PILOT INTO A MULTIAGENT NAVIGATION ARCHITECTURE FOR AUTONOMOUS ROBOTS

Manikanth Mohan
Kanpur Genetic Algorithm Laboratory(KanGAL)
Indian Institute of Technology, Kanpur, India

Dídac Busquets, Ramon López de Màntaras, Carles Sierra
Artificial Intelligence Research Institute(IIIA-CSIC)
Campus UAB, 08193 Belleterra, Barcelona, Spain

Keywords: Potential Fields, Obstacle-Avoidance, Autonomous Robot Navigation, Multi-Agent Systems, Bidding Strategies

Abstract: In this paper we present a new Pilot for a Multi-Agent based control architecture for Autonomous Robots. This Pilot is based on the use of virtual potential fields and is easy to implement yet effective. The Pilot functions as an autonomous agent in a complex Multi-Agent Architecture for the control of an autonomous robot. In this architecture, various agents are responsible for different tasks, and they might have to compete and cooperate for the successful completion of a particular navigation mission. The interaction between different agents is achieved through the use of a bidding mechanism. In this respect, we also present a way to define the bid for the new pilot, so that the pilot can be integrated easily into the Multi-Agent Architecture. The pilot has been tested on navigation experiments involving a real robot and it has given successful and encouraging results. We also deal with some problems of this pilot and ways to get around them.

1 INTRODUCTION

Robot systems for navigating through unknown environments has been a focus area of research for many years now, with many application areas including rovers, handling hazardous materials, and search and rescue missions, to name a few. Real time obstacle avoidance is a major concern in Robotic systems (Arkin, 1998; Latombe, 1996) for their critical importance to the success in any real life application of robotics. (Manz et al., 1991) presents a good comparison of some real time obstacle avoidance methods. The use of virtual potential fields (PF) was introduced in (Khatib, 1985) and since it has been explored and discussed to a great deal (Masoud and Masoud, 2000; Khosla and Volpe, 1988). A modified implementation based on PF can be found in (Koren and Borenstein, 1991). Using this as a base idea we have developed an agent, the *Pilot*, which takes care of real time obstacle avoidance. This agent is a part of the overall Multi-Agent System where various agents are responsible for different tasks, and they have to compete and cooperate for the successful completion of a particular navigation mission. The interaction between different agents is achieved through the use of a bidding mechanism. See (Busquets et al., 2003) for further details.

2 THE NEW PF BASED PILOT

The Pilot is responsible for safely commanding the motors that control the robot to move in a given direction. It implements the requests for motion received from the Navigation system (if they are safe), and is responsible for avoiding obstacles and for providing reflex reaction when the robot makes an unexpected collision. With the information of the known obstacle locations, it uses a PF based technique, to decide the turn necessary for avoiding collisions. The Pilot also generates a bid, indicating the urgency of the action it is proposing. Since obstacle avoidance is of maximal importance, the bid, in critical conditions, should be higher than the bids coming from the Navigation system. However, it should not be set to the highest possible value, 1, so that there is the possibility of adding a new system that can override the Pilot (e.g. a manual override). Additionally the Pilot also makes bids for camera control, to look ahead when required (e.g. for gathering additional information about obstacles). This bid is based on a function that raises the bid depending on the distance traveled since the last time the robot looked forward. Although this is also an important function of the Pilot, in this paper we only focus on the obstacle avoidance module.

Mohan M., Busquets D., López de Màntaras R. and Sierra C. (2004).
INTEGRATING A POTENTIAL FIELD BASED PILOT INTO A MULTIAGENT NAVIGATION ARCHITECTURE FOR AUTONOMOUS ROBOTS.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 287-290
DOI: 10.5220/0001129402870290

2.1 Mathematical Formulation: Point Obstacles and the Target

For developing the PF-method, we follow the usual approach where the magnitude of the virtual force is dependent on the square of the distance between the robot and the obstacle. The direction of the force is that of the line joining the obstacle and the robot. If the body in question is an obstacle, then the force is repulsive in nature meaning that the robot and obstacles hypothetically have *"charges"* of the same sign. The target *"charge"* is of the opposite nature, thereby providing an attractive force on the robot. We can express the virtual force on the robot due to an obstacle o, $\mathbf{F_o}$, by (we denote vectors in boldface):

$$\mathbf{F_o} = \frac{K}{d^2}\hat{\mathbf{u}}$$

where K is a constant of propotionality, d is the ditance between the robot and the obstacle, and $\hat{\mathbf{u}}$ is a unit vector in the direction from o to the robot. The obstacles are initially represented as points (for landmarks and simple obstacles) and lines (for linear obstacles), but are later *"grown"* to become circles and rounded rectangles respectively. These areas, referred to as prohibited areas, represent the obstacles they contain. In our case, the growing size is kept slightly larger than the diameter of the robot. The calculation of the forces is made using the centers of the *"grown"* circles. We represent the radius of this circle by R_{min}. If the robot is inside this prohibited region, (a situation we denote by the name *IN_DISTRESS*) then the force exerted by this body is of constant magnitude away from the obstacle, $\mathbf{F_{max}} = \frac{K}{R_{min}^2}\hat{\mathbf{u}}$. Therefore, the final equation expressing the force due to an obstacle is given as:

$$\mathbf{F_o} = \begin{cases} \frac{K}{d^2}\hat{\mathbf{u}} & \text{if } d > R_{min} \\ \mathbf{F_{max}} & \text{if } d \leq R_{min} \end{cases}$$

To take into account the direction of the target, and the fact that no matter where the robot is our aim is to reach the target, we take the attractive (i.e. negative) force of the target to be a constant value. We therefore define the target force, $\mathbf{F_t}$, by:

$$\mathbf{F_t} = -A_t\hat{\mathbf{v}}$$

where A_t is a constant of proportionality, and $\hat{\mathbf{v}}$ is a unit vector in the direction of t.

So far we have defined the forces for the point obstacles and the target, we are yet to define the force for linear obstacles, which we treat in the next section.

2.2 Linear Obstacles

The natural extension for calculating repulsive forces for linear obstacles is by integrating the force (per unit length) over the length of the obstacle. But, this causes a non-uniform distribution of forces, for example, it is stronger at points which lie near the perpendicular bisector of the line obstacles, compared to the points on the sides. Due to this, the presence of another obstacle nearby can cause the robot to collide into the linear obstacle. So, the magnitude of the repulsive force is calculated by taking the nearest point from the robot to the linear obstacle, while the direction is given from its center (see Figure 1). This new method for dealing with linear obstacles ensures that the repulsive force is high always whenever the robot is near the linear obstacle, independent of the orientation with respect to the robot.

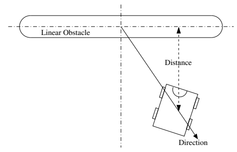

Figure 1: Calculation of force for a Linear Obstacle

2.3 Diverting Targets

Under the overall architecture of the robot, the Navigation system communicates the current target to the Pilot. This may differ from the initial target given at the beginning of the mission because the Navigation system can set up temporary targets (called diverting targets), in the form of crossing a virtual line joining two landmarks to help the robot escaping from a difficult or blocked position. Here, we would like the Pilot to lead the robot across the virtual line without colliding with the landmarks at both ends. Simply putting an attractive force towards the line can cause the robot to crash into one of the landmarks at the end. Another easy way out is by putting an attractive force towards the middle of the line. But this might not be optimal, specially if the diverting line is very long. We therefore approach this problem in a different manner. We define a corridor perpendicular to the line target, which is at a safe distance from the landmarks at both ends. If the robot is inside this corridor, the attractive force is simply perpendicularly towards the line target; otherwise it is towards the center of the line. This makes sure that the Pilot leads the robot across the lines without risking a collision.

2.4 Putting it all together

Once we have defined the virtual forces exerted by obstacles and targets, we are ready to define how the total force is computed. The total virtual force on the robot is found by vector addition of all obstacle and target forces. So, if $O = \{\text{All known obstacles}\}$, then the net force on the robot, $\mathbf{F}_{net}$, is given by:

$$\mathbf{F_{net}} = \sum_{\forall o \in O} \mathbf{F_o} + \mathbf{F_t}$$

We can express this in terms of its x and y components as: $\mathbf{F_{net}} = F_x\mathbf{i} + F_y\mathbf{j}$ and the turn angle θ by $\theta = tan^{-1}\left(\frac{F_y}{F_x}\right)$. The direction of motion of the robot is in the direction of the net force, and thus, θ is the turn angle returned by the Pilot. Along with this, the Pilot also returns a bid (see Section 3) which in some sense indicates the urgency of taking this turn.

In our implementation the values of the parameters are: $K = 18$, $A_t = 50$ and $R_{min} = 0.40m$. These values were arrived at after considerable experimentation using the simulator and the real robot. These values can also be derived analytically. The following equation sets the relation between A_t and d_{min}: $A_t = \frac{0.76 \cdot K}{d^2_{min}}$, where d_{min} is half of the minimum distance between two obstacles the robot would go through. We do not include the whole analysis due to space limitations. With the parameters found experimentally, $d_{min} = 0.52m$, thus, the Pilot is able to drive the robot between two obstacles 1 meter apart.

3 INTEGRATION INTO THE MAS

As mentioned earlier, the Pilot receives requests from the Navigation system to move the robot towards the target and it also proposes its own actions for real-time obstacle avoidance. Thus, it has to compete with the Navigation system in order to have its proposed actions executed. For this competition, the Pilot sends a bid along with its turn angle.

The Pilot executes the turn of the winning bid (the higher bid wins). This competition ensures that if the robot is near an obstacle, the Pilot's action is executed because its bid will be higher than that of the Navigation system. Similarly, if the robot is in a safe region, then the Pilot's bid is low. This helps us in achieving what we want, i.e. that the Pilot should come into play only when the robot might get into trouble and not interfere with the navigation otherwise. Usual PF based methods suffer from this problem, that they interfere with the normal movement of the robot even when it is unnecessary. But, by treating the new Pilot as an agent in a Multi-Agent System, where agents compete and cooperate for the resources of the robot, we are able to work around this problem.

Now we need to come up with a good way to define the bid of the robot, so that it meets the needed requirements (i.e. the bid be high, if the robot is near or heading towards an obstacle and is low otherwise). But achieving this is not difficult and is in fact very intuitive. We first find the maximum repulsive force generated by any individual obstacle, $G_{max} = max\{|\mathbf{F_o}| : \forall \mathbf{o} \in \mathbf{O}\}$ We then define the bid of the Pilot, bid_p as:

$$bid_p = \gamma_p \left(\frac{G_{max}}{|\mathbf{F_{max}}|}\right)$$

The multiplication factor of γ_p is to scale the bid within the interval [0, γ_p], so that if necessary, we can incorporate another agent in the future which has the ability to overrule the bid of the Pilot. In the current implementation, the value of γ_p is set to 0.9.

The reason why we use the maximum repulsive force instead of the net force or the total repulsive force is because the presence of obstacles nearby can hamper the magnitude of the force. By considering the maximum repulsive force, it becomes very unlikely that the bid of an agent from the Navigation System will win, if the robot is very near to any obstacle. We see that by limiting the maximum force due to an individual obstacle to G_{max}, the generation and definition of the Pilot's bid is easy.

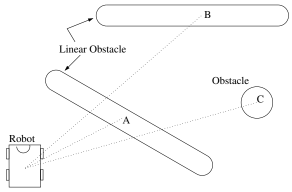

Figure 2: More than one obstacle in same direction

4 SOME PROBLEMS AND SOLUTIONS

The Pilot at this stage was able to perform the basic duties of avoiding the obstacles and also lead the robot to the target. Over some runs of the robot, some problems were noticed. The two most important are described below, along with the changes made to counter them.

More than one obstacle in the same direction: Consider the situation shown in Figure 2. There are three repulsive forces on the robot, one from each of the obstacles, producing a strong repulsive force away from the region. However, a force from A is sufficient, because obstacles B and C are shielded by A. In this case it is better not to compute the force for all the obstacles that are hidden behind the linear obstacle. To do this, we consider only those obstacles which are directly facing the robot. If the line segment from the obstacle (or center of linear obstacles) to the robot intersects any other obstacle, it does not contribute any

Table 1: Results of the performance of the two pilots.

	Time Taken (in sec)		Path Length (in cm)	
	Avg	Std Dev	Avg	Std Dev
GP	297.33	*81.91*	1532.04	*429.33*
PFP	243.37	*99.05*	1194.92	*198.23*

repulsive force. So in the above case , only the force from the line segment A would be considered. The line segment joining the center of B to the robot and from C to the robot intersect A, hence these two obstacles provide no repulsive force.

Effect of obstacles even after they have been passed: The obstacles continue to affect the motion of the robot even after the robot has passed by them. Even though this problem is expected from PF based approaches, we try to reduce this by setting the repulsive force to be zero if the angle between the net repulsion and the net attraction is less than 90^o. Even though this does not completely solve the problem, it helps turning towards the target easier than without this modification. Other ways that could be used to overcome this is by taking into account the orientation of the robot and its direction of motion.

5 EXPERIMENTAL RESULTS

This Pilot was implemented and tested on a real Pioneer 2 AT robot and was found to give satisfactory results. To make a comparative study, the performance of the new Pilot was tested against the Pilot that was used in the architecture till then. The earlier Pilot, called the *Geometric Pilot* (GP) was simple and used a nearest obstacle avoidance (using geometric principles) algorithm. We measure the performance of the Pilot in terms of the time taken to reach the target and the length of the path taken. Table 1 shows the averages and the standard deviations of these measures for 45 reruns on an environment of size 8 by 8 meters. The table clearly shows that the performance of the PF based Pilot is much better than the Geometric Pilot in both measures. The PF-Pilot improves the time taken by an average of 18.14%, while it improves the path length by around 22%. Therefore the new PF-Pilot is able to significantly improve both the time and the path taken by the robot.

6 CONCLUSION AND SCOPE

In this paper, we have presented the development of a new Pilot which uses a virtual potential field strategy for obstacles avoidance. By using the PF method to build an agent, as part of a Multi-Agent system, we are able to work around some problems that are inherent to PF methods. The way of defining the bid ensures that the Pilot intervenes only when necessary. The maximum bid of the Pilot is kept higher than the other systems to ensure that the Pilot gets higher priority in critical conditions. As shown by the results of the experiments, the new Pilot provides much better performance and is able to considerably improve the performance of the robot, by saving both time and path-length. We have also found ways to deal with some problems faced by the new Pilot. A simplification that we have assumed here is that obstacles are either linear in shape or are points. This assumption is usually not valid in actual outdoor environments, but most indoor environments can be approximated with such lines and points. The extension of PF methods to deal with arbitrary shapes can be very complex and an approximation to lines and points might be easier and effective rather than an exact method.

ACKNOWLEDGEMENTS

This work has been partially supported by the MCyT's project QualNavEx DPI2003-05193-C02-02 and CIRIT's project CeRTAP.

REFERENCES

Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press, Cambridge, MA.

Busquets, D., Sierra, C., and López de Màntaras, R. (2003). A multi-agent approach to qualitative landmark-based navigation. *Autonomous Robots*, Kluwer Academic Publishers, 15:129–153.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 500–505, St. Louis, Missouri.

Khosla, P. and Volpe, R. (1988). Superquadric artificial potentials for obstacle avoidance and approach. In *Proc. of the IEEE Conf. on Robotics and Automation*, pages 1778–1784, Philadelphia, USA.

Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *Proc. of the IEEE Conf. on Robotics and Automation*, pages 1398–1404, Sacramento, CA, USA.

Latombe, J. (1996). *Robot Motion Planning*. Kluwer Academic Publishers, UK.

Manz, A., Liscano, R., and Green, D. (1991). A comparison of real-time obstacle avoidance methods for mobile robots. In *Second Intl. Symposium on Experimental Robotics*, Toulouse, France. Springer-Verlag.

Masoud, S. and Masoud, A. (2000). Constrained motion control using vector potential fields. *IEEE Trans. on Systems, Man, and Cybernetics*, 30(3):251–272.