Proc. 2002 Australasian Conference on Robotics and Automation
Auckland, 27-29 November 2002

Producing Complex Behaviours on Trajectory Velocity Learning Mobile Robots

Koren Ward
School of Information Technology
and Computer Science
University of Wollongong,
Australia 2522
koren@uow.edu.au

Abstract

This paper describes a novel and effective method for devising complex behaviours on a mobile robot that is considerably easier to implement than most existing robot control methods. Instead of designing multiple behaviours and a subsumption layer for selecting behaviours, this method provides the robot with only one behaviour (goal seeking) and produces more complex behaviours by altering the robot's perception of the environment so that significant objects appear to have different shapes. For example, a robot playing robot soccer can be made to always take shots at the goal by placing an illusionary object around the ball which has an open end away from the goal. Thus, by being attracted to the ball and repelled by the sides of the illusionary object the robot is directed onto the ball in the direction of the goal. This effectively causes the robot to always line up and strike the ball toward the enemy goal. Experimental, results of this goal shooting behaviour are provided. These results demonstrate both the simplicity and effectiveness of this approach at controlling a robot to perform a task that would be considerably more difficult to implement with other methods.

1 Introduction

Various behaviours are often used to describe and quantify the motion of a mobile robot. Such behaviours can be referred to as simple behaviours (like *obstacle avoidance* and *wandering*) or more complex behaviours (like *goal seeking*, *box pushing* or *collecting cans*). Complex behaviours can be achieved by arbitrating a number of simple behaviours, via a behaviour arbitration system, or by combining simple behaviours by some other means. In the following section a brief description of various techniques for implementing complex behaviours on a mobile robot is given. Following this an alternative approach is described that implements complex behaviours by altering the robot's perception of objects. Experimental results are provided to demonstrate the effectiveness of this method.

2 Implementing Complex Behaviours on Mobile Robots

One method used for implementing goal seeking behaviour on a mobile robot involves the use of Artificial Potential Fields (APFs). The APF algorithm for goal seeking behaviour was first developed by [Krogh, 1984] and uses an occupancy grid to calculate artificial forces for guiding the robot's motion toward a goal while avoiding any objects it encounters. To achieve this a *potential function* is used to describe the environment space in terms of artificial forces. Generally, repulsive forces are generated by obstacles and are dependent on their distance from the robot. An attractive force is generated by the goal which is usually assumed to be independent of the separation distance between the goal and the robot [Latombe, 1991].

Although APF techniques appear simple in principal, there are a number of limitations that can inhibit efficient robot motion or prevent the goal from being achieved:

- Trap situations can occur due to the presence of *local minima* where the net force is equal to zero.
- Closely spaced obstacles may fail to be negotiated due to the forces from the surrounding objects being too strong with respect to the goal force.
- Repulsive forces experienced simultaneously from opposite sides may result in oscillations occurring in between objects or in narrow corridors.
- The robot may be slow to realise the presence of objects at startup or upon entering unknown regions of the environment.
- Wheel slip or hard bumps can misalign the occupancy grid with the environment causing inappropriate responses.
- The artificial forces acting upon the robot only serve to indicate the appropriate direction of motion and do not indicate appropriate robot velocity.
- Slow command evaluations are likely, particularly with larger occupancy grids.

An alternative method for achieving complex behaviours on a mobile robot is Subsumption Architecture [Brooks, 1986]. Subsumption architecture does not employ a symbolic world model. Instead control is achieved by the *suppression* or *activation* of a number of simple behaviours as depicted in Figure 1.

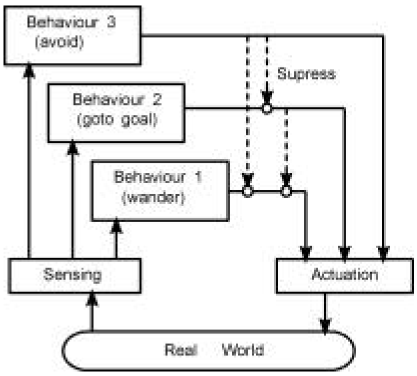

Figure 1 Subsumption architecture.

Although subsumption architecture has proven itself effective at achieving a variety of complex behaviours, it can be difficult to implement. Each behaviour has to be devised and thoroughly tested and it can often be unpredictable if the subsumption layer and given behaviours will result in appropriate control when the behaviours interact. Furthermore, it is difficult to provide subsumption robots with the ability to adapt to unknown or changed environments. Although it has been shown that the subsumption layer can be adapted to perform optimally in differing environments [Mataric, 1997], adapting the low-level behaviours to suit different environments is considerably more difficult (as shown by [Connel and Mahadevan, 1992].

Goal seeking behaviour can also be produced by a mobile robot behaviour learning method called Trajectory Velocity Learning (TVL) [Ward and Zelinsky, 2000]. This has the advantage of being adaptive and capable of seeking goals without the need to switch behavours, as explained in the following section. Furthermore, in Section 4 it is explained how other more complex behaviours can be implemented easily with TVL by simply altering the robot's perception of the environment so that significant objects appear to the robot to have different shapes.

3 Trajectory Velocity Learning

TVL differs to other robot behaviour learning methods in that the robot learns associations between sensors and appropriate trajectory velocities as depicted in Figure 2.

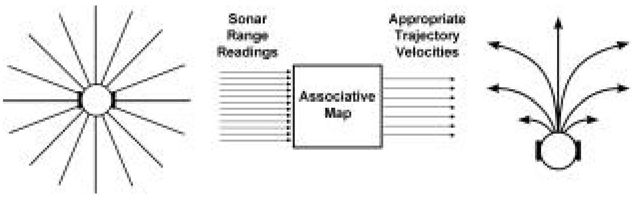

Figure 2. Learning a map between sensors and appropriate trajectory velocities.

So instead of learning output responses, as in Reinforcement Learning (RL), TVL robots actually learn how fast they can traverse their predefined trajectory commands. For example, Figure 3(a) shows a robot and range readings that could emerge from the robot's sonar sensors near an internal corner. By providing these range readings to the learnt associative map, shown in Figure 3(c), the robot receives information indicating the appropriate velocities that should be used for following its available trajectory commands shown in Figure 3(b). Here the map informs the robot that trajectories on the right can be traversed more quickly than the other trajectories, which collide with the walls. In the following sections it is explained how this trajectory velocity information can be used to produce a variety of mobile robot behaviours. However, before explaining this it is necessary to firstly understand just how appropriate trajectory velocities can be learnt automatically by the robot.

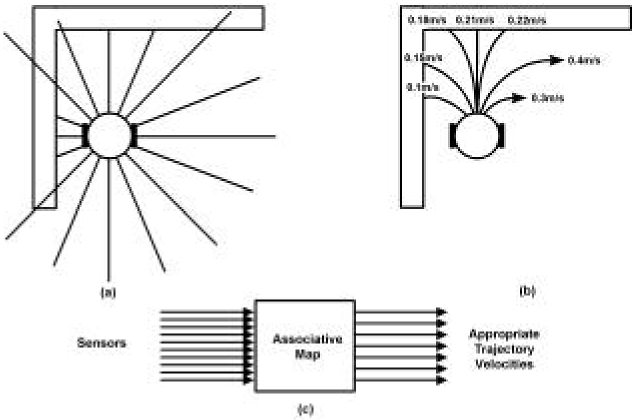

Figure 3 Mapping sonar sensor range readings to appropriate trajectory velocities.

3.1 Learning Appropriate Trajectory Velocities.

One way appropriate trajectory velocities can be learnt is by providing the robot with a special trajectory velocity learning behaviour. This special learning behaviour works by randomly selecting trajectories and slowly following each until a collision or full circle occurs, as shown in Figure 4.

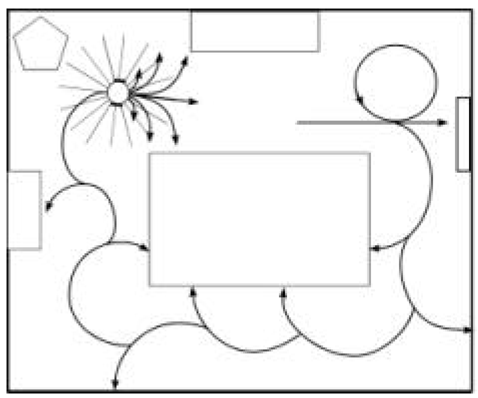

Figure 4 Learning appropriate trajectory velocities by locating trajectory collision points.

When a collision occurs, as shown in Figure 4, a preset constant deceleration rate is used to calculate appropriate trajectory velocities leading up to the collision point. These calculated velocities are then associated with the sensor data that occurred at each time step leading up to the collision point. The resulting training patterns are then used to train Fuzzy Associative Maps (FAMs) as explained in [Sudkamp and Hammell, 1994]. If the robot happens to select a trajectory that leads into free space, which results in the robot completing a full circle, the trajectory's maximum velocity is used to associate with the sensor data for training the FAMs.

By using the robot to learn associations between sensor range readings and trajectory velocities the robot learns to perceive its environment in terms of appropriate trajectory velocities via its sonar sensors eliminating the need for object locations to be tracked when control decisions are made. Furthermore, the use of a learnt associative map to look up trajectory velocities directly from sensor data enables trajectory velocities to be determined quickly. This results in fast response times and can allow more trajectories to be considered as candidates during each time step.

3.2 TVL Mobile Robot Behaviours

By being able to perceive appropriate trajectory velocities, the robot can use this information to produce a variety of behaviours by selecting fast trajectories with respect to some predefined criteria.

3.3 Object Avoidance Behaviour

If the robot is given a single instruction to *follow fast*[1] *trajectories nearest to the forward direction* object avoidance behaviour becomes automatically exhibited as shown in Figure 5. This occurs because trajectories which lead into free space are perceived (from the learnt map) as having faster velocities than trajectories that collide with nearby objects.

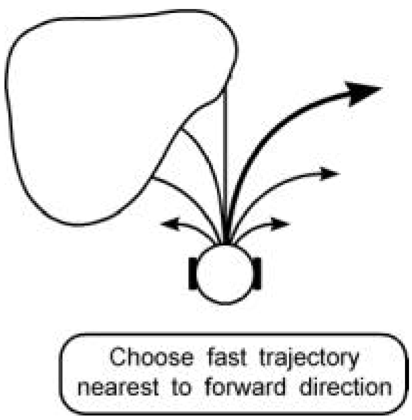

Figure 5 Object avoidance behaviour on a TVL robot.

3.4 Wall Following Behaviour

Similarly, wall following behaviour can also be produced by providing the robot with a single instruction to *follow fast trajectories nearest to the closest object*. This will cause the robot to follow walls in the direction nearest to the direction it is facing. Alternatively, left wall following behaviour can be produced by instructing the robot to *follow fast trajectories nearest to the right of the closest object.* Likewise, right wall following behaviour can be produced by instructing the robot to *follow fast trajectories nearest to the left of the closest object, as shown in Figure 6*.

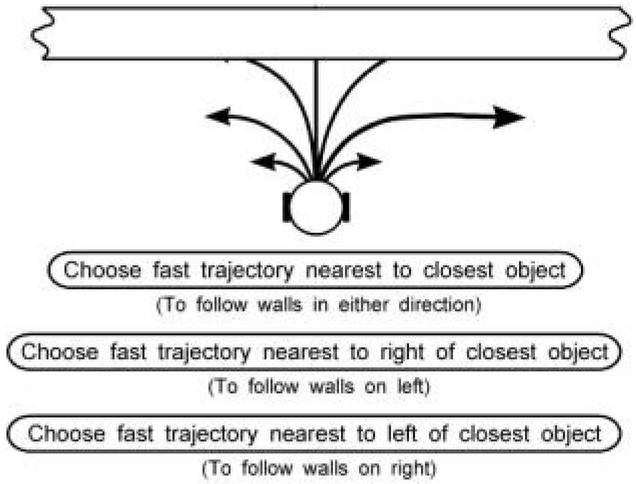

Figure 6 Wall following behaviour on a TVL robot.

3.5 Goal Seeking Behaviour

Goal seeking behaviour is also possible by providing the robot with an instruction to *follow fast trajectories toward the perceived goal location*, as shown in Figure 7. Fortunately, this also produces an implied obstacle avoidance capability without the need to switch behaviours because any convex object encountered between the robot and the goal will cause the direct trajectory to be perceived to be slower than those which lead around the obstacle. However, if the robot encounters a deep crevice simply following a fast trajectory toward the goal may not escape the crevice. To escape deep local minimums the robot could attempt to follow walls in both directions for increasing periods of time.

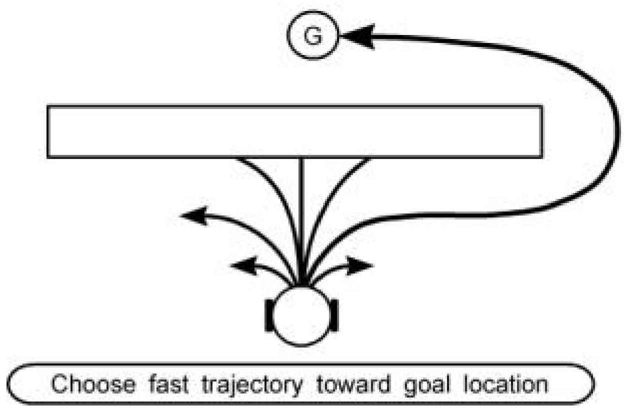

Figure 7 Goal seeking behaviour on a TVL robot.

---

[1] A "fast" trajectory is one that is a high proportion of its maximum velocity.

3.6 Adjustable Object Clearance Distance

By using a variable velocity threshold[2] to determine if perceived trajectory velocities are considered to be fast or slow, the robot's wall clearance distance can be adjusted. Lowering the velocity threshold results in walls being followed more closely and cautiously at lower speed as shown in Figure 8 (a) & (c). Conversely, raising the threshold causes the robot to maintain larger object clearances and results in the robot moving faster and more competently through the environment when performing its behaviours as shown in Figure 8 (b) & (d).

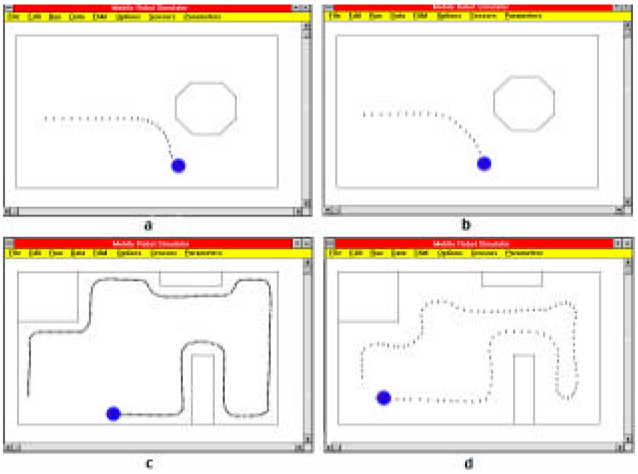

Figure 8 Simulator showing how object clearance distances can be controlled with a velocity threshold.

3.7 Learning Trajectory Velocities with Fuzzy Associative Memories

To learn appropriate velocities associated with all 7 trajectories, 7 FAMs are used, as shown in Figure 9. To reduce the input search space, each FAM's inputs are connected only to a sector of sensors that are adjacent to each FAM's respective trajectory.

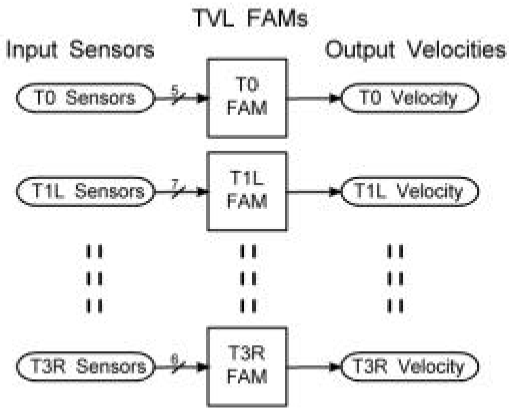

Figure 9 Learning trajectory velocities with 7 FAMs

Fuzzy Associative Maps (FAMs) (see [Kosko, 1992]) are very suitable for learning associations between sensor data and trajectory velocities because they can be incrementally trained via compositional rule inference as explained in [Sudkamp and Hammell, 1994]. Furthermore their output is generalized by taking the weighted average over a small neighborhood of map entries.

4. Implementing Complex Behaviours on a TVL Robot.

To implement behaviours more complex than goal seeking on a TVL mobile robot 3 methods were considered:

1. Using subsumption for arbitrating the robot's acquired low-level behaviours.
2. Engaging the robot's goal seeking behaviour and placing multiple prioritised goals strategically within the environment.
3. Engaging the robot's goal seeking behaviour, with one or more goals associated with significant objects in the environment, and by also altering the appearance of significant objects within the environment. This effectively alters the way the robot approaches significant objects.

As goal seeking behaviour is inherently acquired on a TVL robot without any need to switch behaviours, there appeared little benefit to be gained from implementing subsumption on a TVL robot except for escaping local minimums [Ward and Zelinsky, 2000].

Placing multiple prioritised goals strategically within the environment proved useful for implementing homing behaviour on a TVL robot [Ward et al 1999]. However, when this approach was used for implementing other complex behaviours (like shooting goals within a robot soccer field) placement of the prioritised goals proved more difficult. This occurred mainly because of the need to constantly recalculate all the locations of the prioritised goals with respect to the soccer goal mouth, ball position and robot's location. The location of objects and field boundaries also had to be reckoned when deciding where prioritised goals should be placed or the prioritised goals could end up outside the field.

4.1 Implementing Complex Behaviours with Fictitious Objects

Due to the complexities of implementing prioritised goals for controlling the robot's motion in a game of soccer, a different approach was adopted. This involved placing fictitious objects (hallucinations) within the environment that would cause the robot's motion to alter whenever it approached these objects. Figure 10 shows a simulation of a TVL robot with a fictitious object shown drawn around the ball. By attempting to avoid this object (as the plotted path shows) the robot is guided onto the ball in the direction of the goal. This fictitious object is also implemented with semi-transparent sides so that when the robot enters the fictitious object to approach the ball, the object appears transparent to the robot's sonar sensors (as shown in Figure 10). This allows the robot to strike the ball unimpeded.

[2] The velocity threshold determines if a trajectory velocity is above a specified proportion of its maximum velocity.

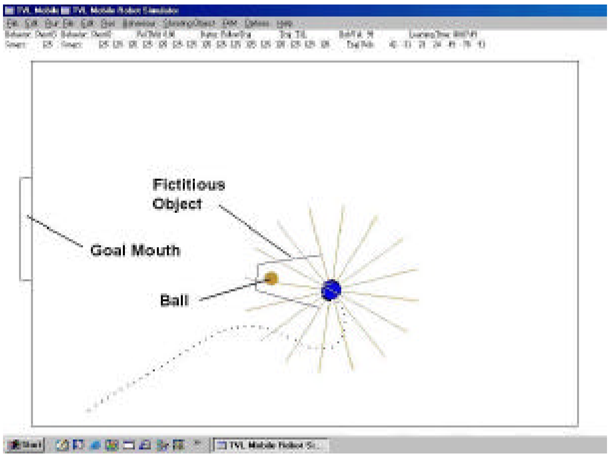

Figure 10. Placing fictitious objects (hallucinations) within the robot's environment to control the robot's motion.

To produce this effect on the robot the location of any fictitious objects within the environment are maintained in the robot's controller. During each time step, a sonar scan of the environment is made with the robot's sonar sensors and the co-ordinates of the fictitious objects is calculated by considering the goal mouth position and the ball location. (Note: the location of the ball, goal mouth, and field corner posts are obtained from the robot's video camera via colour coding and triangulation.) Once the fictitious object's position is calculated from the perceived goal mouth and ball locations, the sonar data is processed so that it appears as if it has reflected off the fictitious objects.

4.2 Processing Sonar Data to Detect Fictitious Objects

To process the sonar data so that it appears as if it has reflected from the sides of the fictitious objects, any sonar beam that is estimated to cut across any external sides of any fictitious object, at an angle of less than 15 degrees off normal, is shortened to its intersection point with the object. All other sonar beams are left at their environmental sensed values. By delivering the altered sonar data to the robot's TVL FAMs, trajectory velocities are returned as if the fictitious object actually exist within the environment. Consequently, avoidance behaviour is exhibited away from the fictitious objects if the robot is on a collision course with any of the fictitious object's sides. By making the ball a goal to be seeked by the robot, the robot is always guided around the fictitious objects and onto the ball in the direction of the goal mouth. Thus, by engaging the robot in goal seeking behaviour, with the ball as the goal to be seeked, goal shooting behaviour is exhibited without any need for the robot to switch it low-level behaviours.

5. Simulator Experimental Results

Simulated experiments with various fictitious objects shown in Figure 11 demonstrate the potential of this approach to implementing complex behaviours on TVL robots. However, some enhancements to the fictitious objects were required to prevent the robot from becoming trapped within the fictitious objects and to facilitate lining up the ball with the centre of the goal mouth.

As previously mentioned, to prevent the robot from getting trapped, the sides of the fictitious objects are implemented in such a way as to make them appear semi-transparent. This results in the robot's sonar sensors appearing to reflect off the external opaque sides of the fictitious objects. However, the internal sides appear transparent to the sonar sensors which makes the fictitious object appear invisible to the robot from the inside, as can be seen in the simulator screen dumps in Figures 10 and 11. Thus, when inside any fictitious object, the robot moves through the sides of the objects to the outside as if it was not there. Yet from the outside, the fictitious object's sides appear opaque and are avoided.

Also, to improve the accuracy of the robot's goal shooting bahaviour a second higher priority goal to be seeked first, is placed in the centre of the opening of the fictitious objects as shown in Figure 11. Therefore, by seeking the higher priority goal first, the robot becomes accurately lined up to take a shot at the goal mouth by seeking the ball at speed.

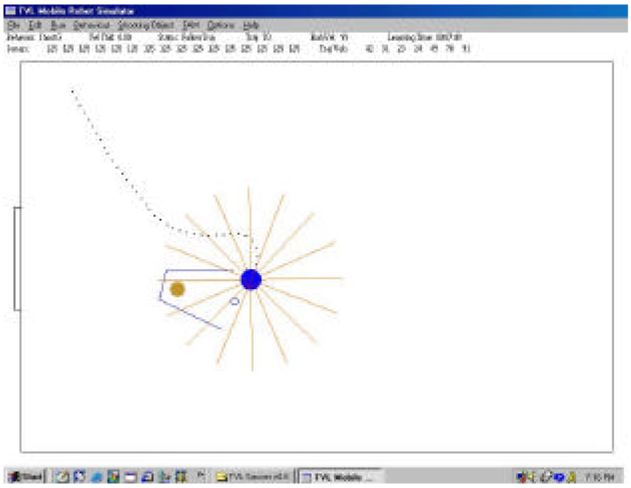

Figure 11. Simulator showing the robot getting into position to take a shot at goal.

Figure 11 shows a screen dump of the simulated robot while lining up the ball to take a shot at the goal. The robot's plotted path shows how the fictitious object is avoided. This effectively guides the robot around behind the ball and onto the first seeked goal which lines the robot up for a shot at the goal mouth.

6. Robot Experimental Results

To conduct goal shooting experiments with the Yamabico robot a video camera was mounted on the robot and used to locate the position of the goal mouth, ball and corner posts of the soccer field as shown in Figure 12. This was achieved by colour coding these landmarks and the ball and by triangulating their locations whenever these landmarks, or the ball, appeared within view of the camera.

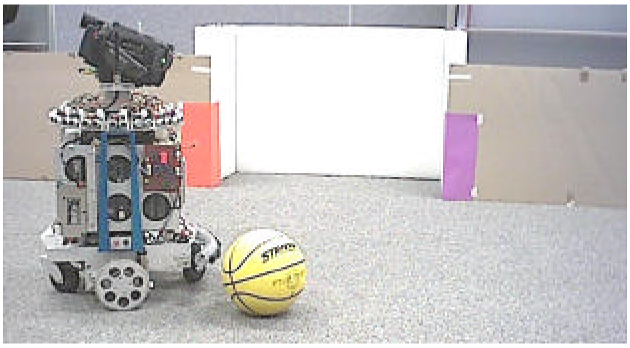

Figure 12. Yamabico robot equipped with sonar sensors and a video camera in the soccer field environment.

The same fictitious object that was implemented on the simulator (see Figure 11) was also implemented on the robot. Processing the sonar range readings to detect the fictitious objects was done by also using the same technique as was used for the simulator (as explained in Section 4.2). Prior to commencing the goal shooting experiments, the robot was engaged in 15 minutes of learning trajectory velocities within the soccer field environment as explained in Section 3.1. The learnt trajectory velocities were saved to a file so the robot could initialise its FAMs with the learnt trajectory velocity data when switched on

To test the robot's goal shooting ability the robot and ball were placed at various locations in the environment and the robot's goal shooting behaviour was activated. Unlike the simulator, the robot was not initialised with its location within the environment or the ball's position. To obtain this information at start up, the robot is rotated until the goal post landmarks and the ball are located. If the ball is not located the robot's behaviour defaults to object avoidance until the ball and soccer goal are found.

Once localisation was achieved in the experiments the goal shooting behaviour exhibited by the robot was comparable with that achieved on the simulator. Generally, the ball would get successfully bumped into the soccer goal as long as it was positioned so that the robot could get behind it to bump it into the goal, as Figure 13 shows.

Figure 13. Photo of robot shooting a goal.

However, when the ball was positioned very close to the walls comprising the field boundaries, the robot often exhibited difficulty getting at the ball. This problem was overcome by only enabling the fictitious objects (ie the goal shooting behaviour) when the ball was in a position where a shot at the goal was possible. When the ball happened to be too near to a boundary wall for a shot at goal to be possible, the robot's normal goal seeking behaviour would cause the ball to be bumped in an arbitrary direction. This effectively moves the ball to another location in the environment where a shot at goal may be possible.

7. Conclusion

Although various methods exist for implementing complex behaviours on mobile robots, these methods are generally difficult to implement and often uncertain of producing adequate results. To address this difficulty, this paper has presented a novel method for implementing complex behaviours on mobile robots by utilising Trajectory Velocity Learning together with fictitious objects for guiding the robot's motion. This behaviour implementation method was tested by implementing gaol shooting behaviour on a mobile robot. It was shown that by being attracted to the ball and repelled by the sides of the illusionary object the robot is directed onto the ball in the direction of the goal. This effectively causes the robot to always line up and strike the ball toward the enemy goal without any need for the robot to switch behaviours. Although more work is needed to demonstrate the effectiveness of this method at achieving a variety of complex tasks on mobile robots, the experimental results reported in this paper demonstrate both the simplicity and effectiveness of this approach at controlling a robot to perform a task that would be considerably more difficult to implement with other methods.

References

[Brooks, 1986] R.A.Brooks. *A Robust Layered Control System for a Mobile Robot*, IEEE Journal of Robotics and Automation, Vol.2, No.1, pp. 14-23, March 1986.

[Connel and Mahadevan, 1992], J. Connell and S. Mahadevan. *Rapid Task Learning for Real Robots*, in: Robot Learning, by J. Connell and S. Mahadevan, Kluwer Academic Publishers, 1993.

[Kosko, 1992] B.Kosko. *Neural Networks and Fuzzy Systems*, Englewood Cliffs, NJ, Prentice Hall, Inc. 1992.

[Krogh, 1984] B. H. Krogh. *A Generalized Potential Field Approach to Obstacle Avoidance*, In: Robotics Research: The Next Five Years and Beyond, Bethlehem, PA. 1984.

[Latombe, 1991] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[Mataric, 1997] M. J. Mataric', *Reinforcement Learning in the Multi-Robot Domain*, Autonomous Robots, Vol.4, pp. 73-83, 1997.

[Sudkamp and Hammell, 1994] T. Sudkamp and R, J, Hammell.. *Interpolation, Completion and Learning Fuzzy Rules*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 24 No. 2 pp 332-342, February 1994.

[Ward *et al*, 1999] K. Ward, A Zelinsky and P. J. McKerrow, *Learning to avoid objects and dock with a mobile robot*, Proceedings of the Australian Conference on Robotics and Automation, Brisbane, pp. 132-137, 1999.

[Ward and Zelinsky, 2000] K. Ward and A. Zelinsky. *Acquiring Mobile Robot Behaviours by Learning Trajectory Velocities.* Autonomous Robots Journal, Vol.9, No.2, pp. 113-133, September, 2000.