Proc. 2001 Australian Conference on Robotics and Automation
Sydney, 14-15 November 2001

Rapid Simultaneous Learning of Multiple Behaviours with a Mobile Robot

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

Abstract

Although various robot behaviour learning methods have been available for some time they generally are too slow to be of much practical use. This paper provides a brief introduction to a novel robot behaviour learning method called Trajectory Velocity Learning and provides some details on implementing Trajectory Velocity Learning on sonar robots with differential drive wheels. The main advantage of Trajectory Velocity Learning is that it enables a mobile robot equipped with range sensing devices to automatically acquire multiple adjustable behaviours quickly and simultaneously.

1 Introduction

The development of robots that can automatically learn behaviours and adapt to changed environments has been a relentless challenge that has confronted robotics for a number of decades. The two most common techniques used to provide mobile robots with adaptive behaviours are Reinforcement Learning (RL) and Genetic Algorithms (GAs). (See [Kaelbling, 1996], [Materic and Cliff, 1996] and [Pollack *et al*, 2000] for a concise survey of RL and GA robot learning methods.) Unfortunately, both these robot behaviour learning methods suffer from major problems which greatly limit both the amount of learning possible within any set period and the amount of sensing able to be provided to the robot. When RL is used to learn low level behaviours successful results are difficult to achieve due to the credit assignment problem. Particularly when the robot is equipped with considerable sensing. Alternatively, if GAs are used to learn robot behaviours, improvements to the robot's behaviour can often take days due to the time it takes to evaluate the performance of possible control solutions on the physical robot. (For examples of RL and GA robot behaviour learning experiments see: [Floreano and Mondada 1996], [Colombetti and Dorigo, 1994], [Connel and Mahadevan, 1992], and [Asada *et al*, 1995].)

To overcome the slow learning time associated with traditional robot behaviour learning methods we have developed a robot behaviour learning method called Trajectory Velocity Learning (TVL) (see [Ward and Zelinsky, 2000]). The main difference between traditional robot behaviour learning methods and TVL is that in traditional robot behaviour learning methods like reinforcement learning the robot learns output commands or actions whereas with TVL the robot learns appropriate trajectory velocities for negotiating predefined trajectory commands.

By learning associations between sensors and trajectory velocities the robot does not suffer from the credit assignment problem or the fitness evaluation problem and can learn certain robot behaviours (eg object avoidance, wall following, goal seeking) much faster. Furthermore these behaviours can be learnt simultaneously and it is also possible to alter the robot's wall clearance distance by adjusting a single parameter.

The following sections provide a brief introduction to TVL with details on how to implement TVL on a sonar robot with differential drive wheels.

2 Trajectory Velocity Learning

To overcome the slow learning times associated with existing unassisted robot learning methods and to enable robots with considerable sensing to learn in real time we decided to alter the actual learning task. Instead of performing the difficult task of learning associations between sensor inputs and output responses, as for example in conventional RL, TVL uses the robot to learn associations between sensors and appropriate trajectory velocities as depicted in Figure 1.

Like RL, each input vector is comprised of the robot's immediate sensor range readings. However, instead of learning a map that delivers appropriate output robot commands, as in RL, TVL learns a map that delivers velocities that are appropriate for negotiating the robot's predefined trajectory commands.

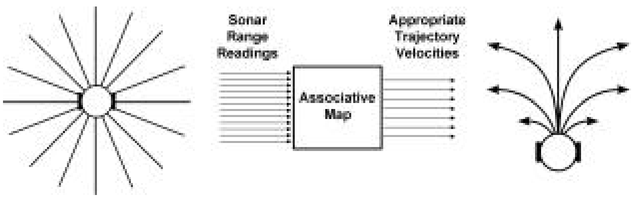
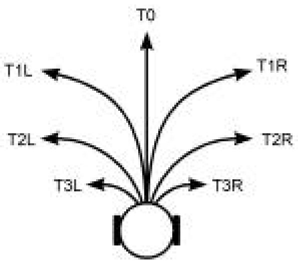


Figure 1 Learning a map between sensors and trajectory velocities.

Figure 2 describes the seven trajectory commands we implemented on our Yamabico robot for conducting our TVL experiments. Each predefined trajectory command is either a line straight ahead or an arc to the left or right of preset radii. A maximum velocity is also specified for each trajectory command and represents the velocity at which each trajectory should be traversed in free space. Since the Yamabico robot is slightly top heavy, it is appropriate to make the forward trajectory, and the trajectories with larger radii, faster than trajectories which turn the robot sharply.



Trajectory	Radius m	Max Velocity m/s
T0	N/A	0.6
T1L, T1R	2.0	0.5
T2L, T2R	1.0	0.4
T3L, T3R	0.3	0.3

Figure 2 Each output trajectory command has a predefined radius and maximum velocity for moving in free space.

2.1 Defining Appropriate Trajectory Velocities

Although the specified maximum velocities can be considered appropriate for negotiating trajectories which happen to lead into free space, lesser velocities are more appropriate for trajectories which happen to be on collision courses with objects. The purpose of the associative map in TVL is to indicate to the robot the appropriate velocities of each output trajectory option with respect to collision distances with objects. 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 2 and 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 are defined and how this information can be learnt automatically by the robot.

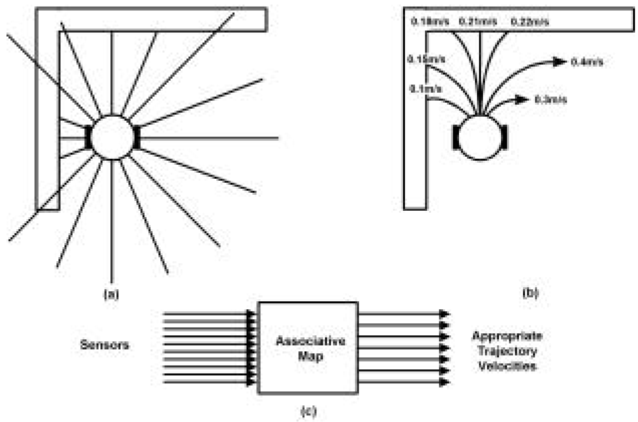


Figure 3 Mapping sonar sensor range readings to appropriate trajectory velocities.

2.2 Appropriate Control of Trajectory Velocities

After having decided the robot's trajectory commands and the velocities at which the robot should negotiate these trajectories in free space, the next task is to define how the robot's velocity should change when obstacles are encountered. Ideally, a robot following a trajectory that is on a collision course with an object should slow down and come to a safe halt just before coming into contact with the object. The deceleration rate should also be considerably less than the robot's maximum deceleration rate to prevent abrupt changes to the robot's motion. For our Yamabico robot, we decided on a uniform deceleration rate of $-0.5\mathrm{m/s}^2$ for all seven trajectories, as shown in Figure 4(a). This is approximately one third of the robot's maximum possible deceleration rate on most hard surfaces.

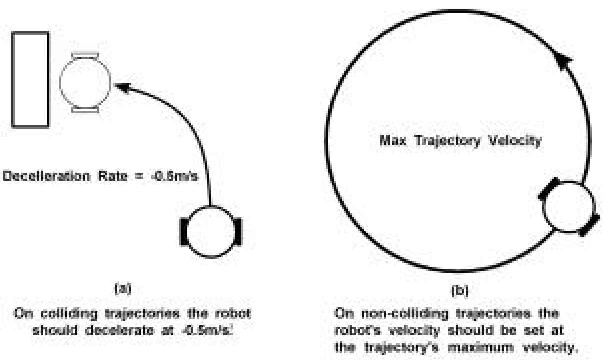


Figure 4 Defining appropriate velocities for negotiating trajectories.

Thus, the objective of appropriate velocity control is for the robot to set its velocity to the engaged trajectory's maximum velocity when the engaged trajectory does not collide with any obstacles, as shown in Figure 4(b). Alternatively, if the engaged trajectory is on a collision course with an obstacle, the robot should set its velocity in accordance with the robot's predefined deceleration rate and the collision distance with the obstacle so that the robot becomes stationary before a collision occurs, as shown in Figure 4(a). By appropriately controlling the robot's velocity in this manner the robot ideally can never collide with an object, even if it maintains a course directly toward an object and typically it would exhibit appropriate velocity control as it negotiates its environment as shown in Figure 5.

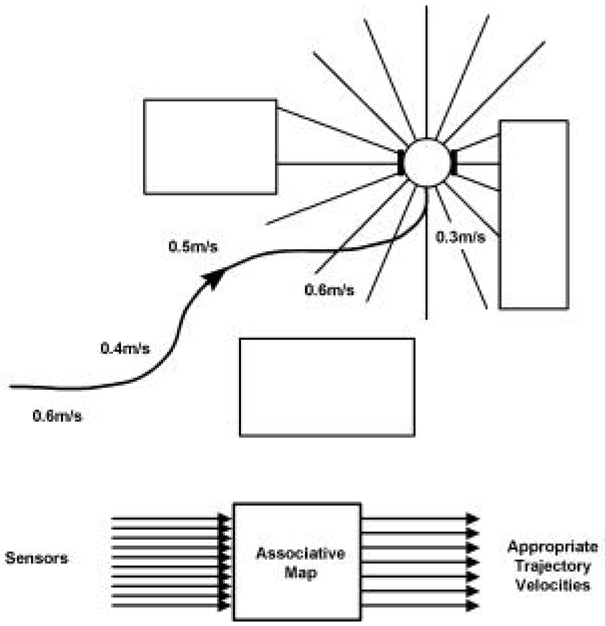


Figure 5 Negotiating an environment while appropriately controlling trajectory velocities.

Although appropriate trajectory velocity control is simple to define, implementing this objective on a robot can be difficult, particularly when the environment is interpreted by the robot via sonar sensors. This is because most objects will only return a sonar signal back to a sonar sensor if the object's surface is almost normal to sonar sensor's beam. Therefore, some objects in front of the robot may not be detected immediately and slight movements of the robot can cause abrupt changes in range readings. Inaccurate range readings can also be produced by sonar signals which are reflected back to the sensor across multiple paths or by *cross-talk*, where one sensor receives an echo transmitted from another sensor. These inaccuracies are exacerbated by unstructured environments like our lab shown in Figure 6 where objects such as tables, chairs, bookcases, cables, etc. can interrupt, delay or scatter sonar signals making collision distances very difficult to calculate directly from sonar range readings. However, by using the robot learn associations between sonar range readings and appropriate trajectory velocities these difficulties are largely overcome.

Figure 6 Negotiating an unstructured environment with sonar sensors.

2.3 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 7.

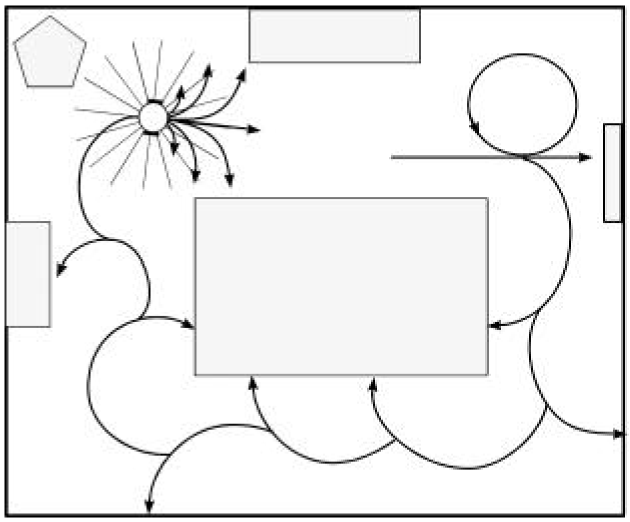


Figure 7 Learning appropriate trajectory velocities by using a special trajectory velocity learning behaviour to locate trajectory collision points.

When a collision occurs, 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 the associative map**, as explained in Section 4.10.** 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 associative map. Figure 8 describes the basic trajectory velocity learning algorithm.

```
loop
    select a random direction and trajectory;
    rotate robot to face chosen direction;
    repeat
        move one time step along chosen trajectory;
        record input vector;
    until collision or full circle occurs;
    associate appropriate velocities
    with each recorded input vector;
    update the associative map with
    the resulting training exemplars;
    if collision occurred reverse robot
    a short distance along previous path;
end loop
```

Figure 8 Basic algorithm for learning trajectory velocities.

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 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.

3.1 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 9. 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.

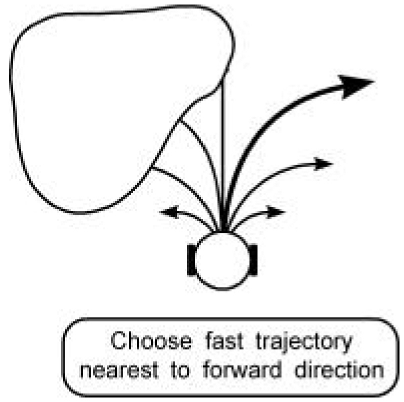


Figure 9 Object avoidance behaviour on a TVL robot.

3.2 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 10*.

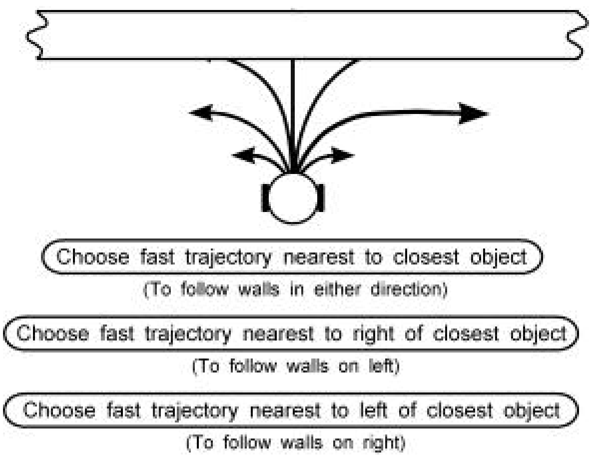


Figure 10 Wall following behaviour on a TVL robot.

3.3 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 11. Fortunately, this also produces an implied obstacle avoidance capability without the need to switch behaviours because

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

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.

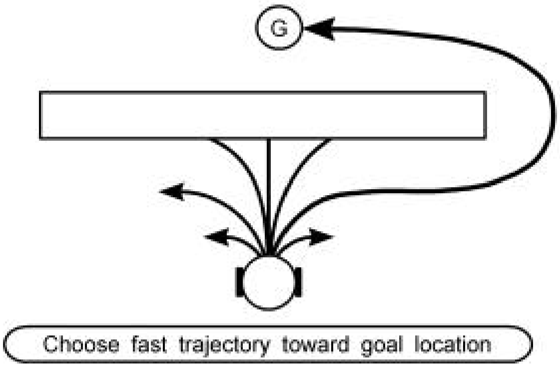


Figure 11 Goal seeking behaviour on a TVL robot.

3.4 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 12 (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 12 (b) & (d).

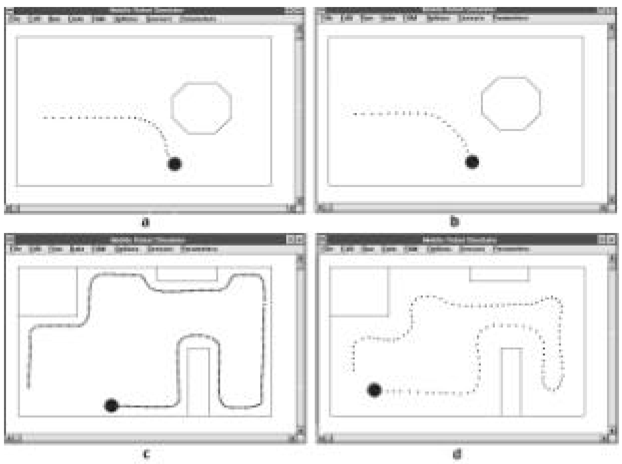


Figure 12 Simulator showing how object clearance distances can be controlled with a velocity threshold.

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

4 Learning Trajectory Velocities with Fuzzy Associative Memories

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. To learn appropriate velocities associated with all 7 trajectories, 7 FAMs are used, as shown in Figure 13. 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.

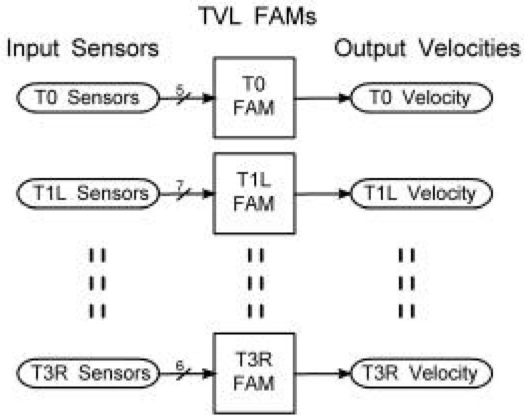
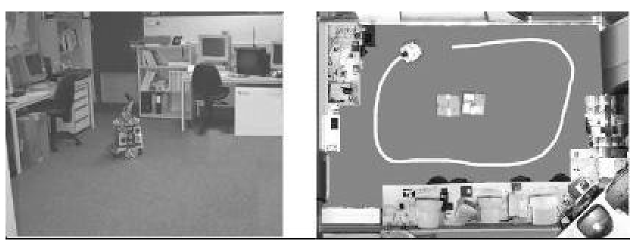


Figure 13 Learning trajectory velocities with 7 FAMs

5 Conclusion

When unstructured environments are interpreted via sonar sensors, like the lab shown in Figure.14(a), range readings can be difficult to predict. This is because a sonar signal will only be reflected back to a sonar sensor if the object's surface is almost normal to the sensor's beam. Consequently, objects such as tables, chairs, bookcases, cables, etc. can interrupt, delay or scatter sonar signals. Therefore hardwiring behaviours into mobile robots can be difficult and unreliable within unstructured environments. However, by learning to perceive the environment in terms of appropriate trajectory velocities a mobile robot can acquire a variety of adjustable behaviours quickly and simultaneously in addition to appropriate velocity control, as shown in Figure 14(b).



(a) (b)

Figure 14 Wall following behaviour exhibited by the Yamabico robot after 45 minutes of learning within an unstructured laboratory environment.

References

[Asada *et al*, 1995] M. Asada, E. Uchibe, S. Noda, S. Tawaratsumida and K.Hosoda. *Coordination of Multiple Behaviours Acquired by Vision-Based Reinforcement Learning*. Proceedings of IROS'94, pages 917-924, 1994.

[Colombetti and Dorigo, 1994] M. Colombetti and M. Dorigo, *Training Agents to Perform Sequential Behavior*, Adaptive Behavior Vol.2 No.3, pages 247-276, 1994.

[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.

[Floreano and Mondada 1996] D. Floreano and F. Mondada, *Evolution of Neural Control Structures: Some Experiments on Mobile Robots*, Robotics and Autonomous Systems, 1996.

[Kaelbling, 1996] L. P Kaelbling. *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research, Vol.4, pages 237-285, May 1996.

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

[Mataric and Cliff, 1996] M. J. Mataric and D Cliff, *Challenges in Evolving Controllers for Physical Robots*, Robotics and Autonomous Systems, Vol.19, No.1, pages 67-83, 1996.

[Pollack *et al*, 2000] J.B. Pollack, H. Lipson, S. Ficici, P. Funes, G. Hornby and R. Watson, *Evolutionary Techniques in Physical Robotics*. J. Miller (ed) Evolvable Systems: from biology to hardware; Proceedings of the Third International Conference (ICES 2000). pp. 175-186, 2000.

[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 pages 332-342, February 1994.

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