Rapid Simultaneous Learning of Multiple Behaviours with a Mobile Robot

Koren Ward

School of Information Technology and Computer Science

University of Wollongong

koren@uow.edu.au
www.uow.edu.au/~koren
Abstract

Figure 1  Yamabico Mobile Robot

Although various robot behaviour learning methods have been available for some time [1] [2] they generally are too slow to be of much practical use. The following slides provide a brief introduction to a novel robot behaviour learning method called Trajectory Velocity Learning (TVL) [3]. The advantage of TVL is that it enables a mobile robot equipped with range sensing devices to automatically acquire multiple adjustable behaviours quickly and simultaneously.
Trajectory Velocity Learning

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

Trajectory Velocity Learning involves learning a map between the robot's input sensors and appropriate velocities for negotiating the robot's predefined output trajectory commands as shown in Fig. 2.

Figure 3 The robot's predefined trajectory commands

Each output trajectory command has a predefined radius and maximum velocity for moving in free space. Although various combinations of output trajectory commands can be used, the 7 shown in Fig. 3 have been shown to be adequate for medium sized robots while providing relatively smooth motion when commands are switched.
Defining Appropriate Trajectory Velocities

Before implementing a robot that learns appropriate trajectory velocities, it is necessary to firstly define what appropriate trajectory velocities are. Ideally, a robot following a trajectory that is on a collision course with an object, should slow down and stop just before coming into contact with the object. Also, the deceleration rate should be low enough (i.e. around -0.5m/s²) so as to prevent abrupt changes to the robot's motion. Therefore, if a robot is to appropriately set its velocity during each time step, it should set its velocity according to its predefined deceleration rate and the collision distance with the obstacle so that the robot becomes stationary just before a collision occurs, as shown in Fig 4(a). Alternatively, if the engaged trajectory is not on a collision course with an obstacle, as shown in Fig 4(b), the robot should set its velocity at the trajectory's predefined maximum velocity defined in Fig.3.

Figure 4  Defining appropriate velocities for negotiating the robot's trajectory commands.
Perceiving Appropriate Trajectory Velocities from Sensor Data

Unfortunately, due to the highly specular nature of sonar signals, trajectory collision points are very difficult to calculate directly from sonar data, particularly within unstructured environments. However, it is relatively easy for the robot to learn associations between sensor data and appropriate trajectory velocities with an associative map (as explained on page 10). By having a learnt associative map that can indicate appropriate trajectory velocities directly from sensor data, as shown in Fig. 5, the robot can not only appropriately control its velocity but can perform a variety of behaviours by making trajectory choices based on this information.
Object Avoidance Behaviour

By being able to perceive appropriate trajectory velocities directly from sensor data, obstacle avoidance behaviour can be performed by giving the robot a single instruction to *follow fast*\(^1\) trajectories nearest to the forward direction, as shown in Fig. 6. This occurs simply because trajectories which lead into free space are perceived (from the learnt map) as having faster appropriate velocities than trajectories that collide with nearby objects.

---

\(^1\) A "fast" trajectory is one that is perceived as having an appropriate trajectory velocity that is a high proportion of its maximum velocity.
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 Fig.7.
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 Fig. 8. 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.
Adjusting the Object Clearance Distances

Figure 9  Simulation showing how object clearance distances can be controlled with a velocity threshold.

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 Fig. 9 (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 Fig. 9 (b) & (d).

\(^2\) The velocity threshold determines if a perceived trajectory velocity is above a specified proportion of its maximum velocity.
TVL Mobile Robot Controller

Figure 10  A TVL controller for producing multiple adjustable mobile robot behaviours simultaneously.

Figure 10 shows a schematic of a typical TVL controller which summarizes how a mobile robot can become capable of appropriately controlling its velocity and performing a variety of adjustable behaviours by making simple choices based on learnt appropriate trajectory velocities.

But how are appropriate trajectory velocities learnt? . . . .
Learning Appropriate Trajectory Velocities

One way appropriate trajectory velocities can be learnt is by randomly selecting trajectories and slowly following each until a collision or full circle occurs, as shown in Fig 11. When a collision occurs, a preset constant deceleration rate (eg. -0.5m/s²) is used to calculate appropriate trajectory velocities back along the path that lead 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 using compositional rule inference [5]. If a full circle occurs, the trajectory's maximum velocity is used to associate with the sensor data for training the map.
Learning Trajectory Velocities with Fuzzy Associative Memories

![Diagram of FAMs](image)

**Figure 12** Learning trajectory velocities with 7 FAMs

Fuzzy Associative Maps (FAMs) [4] are very suitable for learning associations between sensor data and trajectory velocities because they can be incrementally trained via compositional rule inference [5] and 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 Fig. 12.
TVL FAMs

Sonar Sensors

Fuzzy Membership Functions

- VN Very Near
- N Near
- M Medium
- F Far
- VF Very Far

Figure 13 Inputs and membership functions of FAMs

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. Fig. 13 shows how the sensors and fuzzy membership functions of the FAM matrices belonging to trajectories T0 through to T3L were allocated.
Conclusion

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

When unstructured environments are interpreted via sonar sensors, like the lab shown in Fig. 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 Fig. 14 (b).
References


