Self-Supervised Neural System for Reactive Navigation

Artur Dubrawski
Laboratory of Adaptive Systems
Institute of Fundamental Technological Research
Polish Academy of Sciences
21 Świętokrzyska Str. 00-049 Warsaw, Poland

James L. Crowley
Laboratoire d'Informatique Fondamentale
et d'Intelligence Artificielle
Institut National Polytechnique de Grenoble
46 Av. Felix Viallet, 38031 Grenoble, France

Abstract

This paper deals with an artificial neural system for a mobile robot reactive navigation in an unknown, cluttered environment. A task of a presented system is to provide a steering angle signal letting a robot reach a goal while avoiding collisions with obstacles. Basic reactive navigation methods are briefly characterized, a special attention is paid to neural approaches. Then a qualitative description of a presented system is given. The main parts of the system are: the Fuzzy-ART classifier performing a perceptual space partitioning, and the neural associative memory, storing system's experience and superposing influences of different behaviors. Preliminary tests show that the learning by trial-and-error is efficient, as well in a case of beginning from scratch, as after some disturbances of either system's or environmental characteristics.

1 Introduction

Planned navigation of a mobile robot in a real world often fails due to unpredictable changes of the environment, noised readouts of sensors and imprecise execution of a path. Development of real time robotic applications [2] and analysis of animal behaviour [1] showed that a reflexive element in an autonomous robot control is necessary. Such methods of control, called either a *real-time obstacle avoidance* or a *reactive navigation*, choose appropriate locomotion actions in response to measured perceptual situations immediately, while no planning occurs.

1.1 Related work

In the most classical approach to the problem, called *potential field methods* [6] a robot travels in a field of imaginary forces generated by obstacles (repulsion) and by a goal (attraction). The explicit mapping function calculates a cummulative effect of all of those forces and, in accordance to a resultant, chooses a locomotion command. It was shown in [7] that the potential field methods have their inherent limitations as local minima and oscillations, partially resulting from a general form of a mapping function.

One of the more recent trends in the reactive navigation research is to use *fuzzy logic*, already successfully applied to more general control problems [8]. Basic idea of approximate reasoning in multivalued domains makes simple and intuitive synthesis of a controller. It is quite convenient to incorporate an obvious part of the environmental knowledge into a fuzzy rule base. But in a case of a real world problem, when a model of a controlled plant is known only partially, it is very difficult to set *a priori* a correct number of rules and values of their parameters [10].

Another modern approach to reactive navigation is based upon adaptive capabilities of artificial neural networks, employing learning to tune reactive controllers.

One of the most spectacular applications of neural techniques to a mobile robot control is the ALVINN system [9]. In ALVINN a multilayer perceptron is trained (using back-propagation rule) to map digital video images into steering commands. The training pairs are taken while the vehicle is driven by a human, then the network learns off-line. The aim of the trained system is to follow a road emulating human reactions. The system is successfully implemented on a real vehicle, but there are still several inherent weaknesses of the back-propagation learning method, limiting the flexibility of the system. Training is time consuming and off-line, a particular ALVINN module is dedicated to a specific class of a road. In case of a road type change, an arbitration between different modules responses is neccesary.

The problem of mapping perceptual situations into commands can be actually decomposed into two stages: a *classification* of a measured perceptual situation and an *association* a locomotion action with a perceptual class. In [11] the classification task is performed by a self-organizing Kohonen's map. Input vectors (composed of range-to-obstacle indicators' readouts and direction-to-goal indicator readouts) are partitioned into one of predefined perceptual situation classes. Then the classes are associated with locomotion commands by a neural associative memory.

The system presented in [11] can learn on-line, but keeping the number of perceptual classes constant while learning (as inherent in the Kohonen's algorithm) allows averaging and re-definitions of previously stored knowledge, which not always means refining. The learning process is thus non-incremental, and the perceptual space is, after training, partitioned almost uniformly. Moreover, a neighborhood paradigm offered by a Kohonen's mapping is rather useless in the case of a mobile robot control, because physically neighbouring locations of a robot rarely correspond to neighbouring state space regions.

The most natural approach to the perceptual situations space partitioning for a mobile robot reactive navigation purposes seems to be one of the adaptive resonance theory methods [12, 5]. Actually a member of the ART neural network family is used to partition a perceptual space in the system described in this paper.

1.2 The aim

The particular aim of our research is to develop a reactive navigation system capable of on-line learning reflexive locomotion behaviours, using a trial-and-error training method.

The system should learn incrementally in order to let the robot adapt to changing characteristics of non-stationary environment.

We cope with a large dimensionality of the robot's perceptual space, splitting the learning control problem into *partitioning of the perceptual space* and *association of control actions* subproblems.

The particular task, that the robot is expected to accomplish, is to *follow a goal* while *avoiding obstacles*, so the system is able to model simultaneously those two basic locomotion reflexes.

2 The system description

Mapping from the perceptual space into the commands space is decomposed into subsequent stages of *input data preprocessing*, *perceptual situation classification*, *action association*, and *action validation*. Such an approach to the problem decomposition leads to a hierarchical, layered control architecture, as shown in Fig. 1.

2.1 Input and output

The robot [1] is equipped with 24 ultrasonic range sensors and odometer, providing information about the spatial situation, i.e. relative locations of seen obstacles, and estimated position and orientation of the robot in an entire coordinate system. A goal position is set externally by a path planner or by a human supervisor.

A drive action is equivalent to a command setting a required steering angle. The command is chosen from a discrete set of available actions. We assume that linear movements of the robot are driven by another controller, however our system is able to stop and restart it.

2.2 Input data preprocessing

Since there are many ultrasonic rangefinders mounted round the vehicle, a dimensionality of the sonars' state space is large. However, practically certain groups of sonars are important for monitoring certain manoeuvres of the robot, and for certain scene sectors search. It is possible to decrease a dimensionality of the considered space, while preserving important information, by the way of grouping sonars and using the smallest readouts as representatives for the corresponding groups [5, 10]. For the described system purposes the sonars are divided into seven overlapping subsets, maintaining three sides (front, left and right) and four corners of the vehicle.

Processing only a static frame of the robot's scene configuration is insufficient for a safe navigation in a dynamic environment. Another task of preprocessor is to differentiate individual sonar readouts, and feed the system with a kinematic description of the scene, too. The obstacles' relative velocity data set is also partitioned into seven analogous groups. In such a way, after a scaling, on the sonars readouts preprocessor output there are 14 signals, valued in a range $[0, 1]$, providing an input vector to the *scene configuration partitioner* module.

The goal heading angle is calculated from the robot odometric position and orientation, and the externally set goal position. Then the angle is classified into one of 24 sectors of goal heading, distributed uniformly round the robot's center of rotation.

[1] *Robuter* laboratory vehicle used in *LIFIA*.

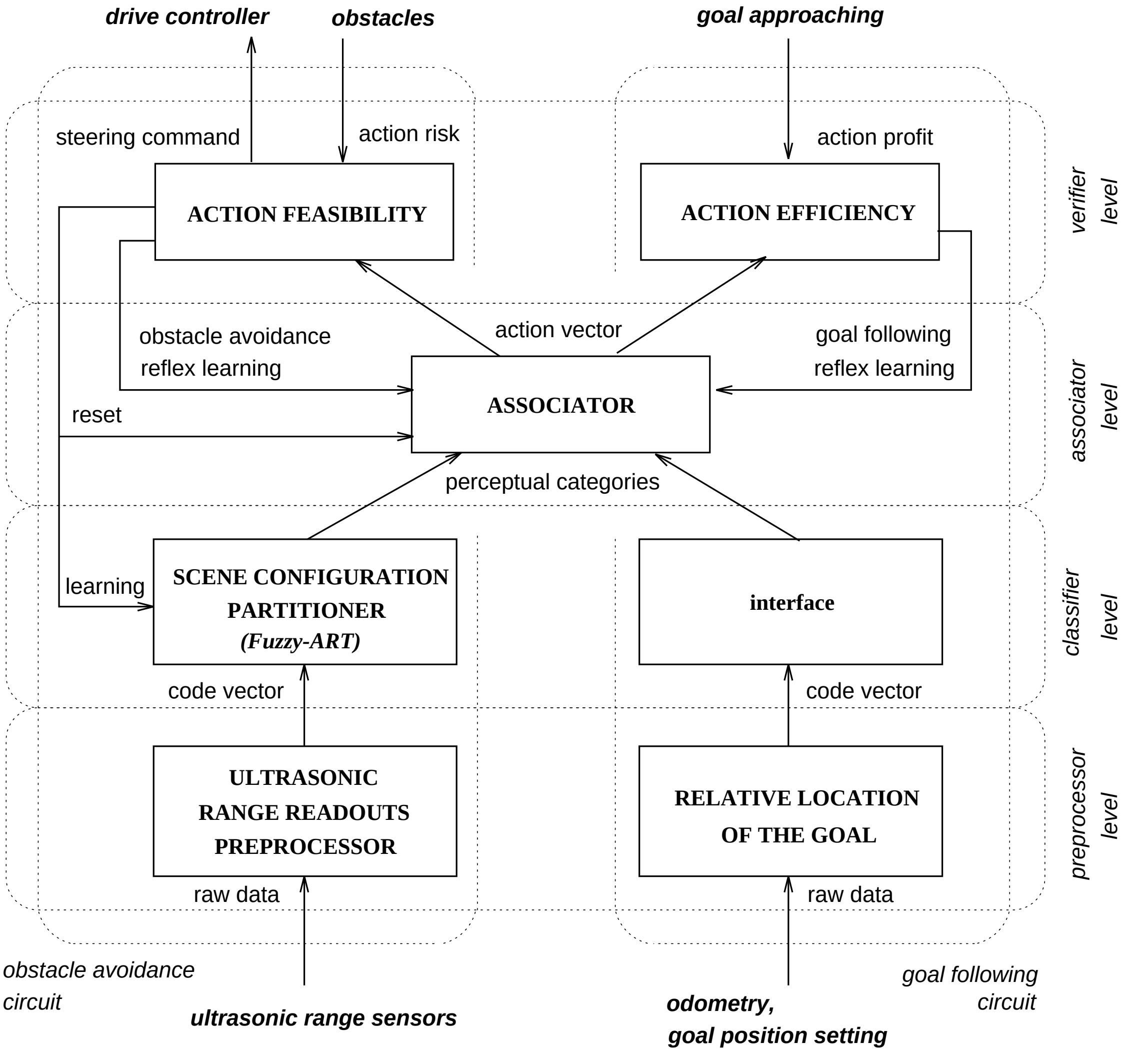

Figure 1: The system architecture

2.3 Partitioning of the perceptual space

As a partitioner of the scene configuration space there is implemented a self-organising adaptive resonance *Fuzzy-ART* [3] neural network module.

Neural networks belonging to the ART family have several features, important for the described system purpose. In particular the ART neural nets are capable of incremental learning, they memorize novelties without direct averaging as in Kohonen's maps. A learning process is performed *on-line* i.e. the net can learn in the same cycle while responding to an input stimulus. Moreover, in the ART-like systems a number of classes is not fixed, a network grows while learning, keeping a minimal configuration, enough in size to represent an input space categorization within a certain accuracy limits.

We implement the Fuzzy-ART neural network in a complementary coding form [3]. The input vector has 14 real-valued components, on the output there is a binary vector containing at most a single non-zero component.

In order to keep the number of scene configuration categories in reasonable boundaries, only impotrant perceptual situations may run a learning routine, and cause the network grow. Since there is plenty of *safe* regions in the perceptual space, they may be left uncoded. We call the situation important, if an action, just chosen by the system, cannot be safely executed.

There is no need to classify the goal heading code. The *interface layer* purpose is just to transform the sector code into a binary vector containing a single non-zero value at the index corresponding to the goal heading sector class.

2.4 Action association

A role of action associator is performed by a single layer neural associative memory, using a nonmonotonic reinforcement learning rule. Its purpose is to learn to select proper actions in accordance to the recognised situation categories, and to superpose competitive influences of *goal following* and *obstacle avoidance* reflexes.

A number of neurons in the associator is equal to the number of steering commands available. In our case the action vocabulary consists of only 3 commands: *keep the steering angle, turn left a small (a couple of degrees) angle, turn right a small angle.*

Associator neurons are attracted, through adaptive connections, by signals coming from the output nodes of the *Fuzzy-ART* module and from the goal heading circuit *interface layer* cells. Excitations of the individual neurons depend on the stimuli and the connections strengths.

The most excited neuron of the associator layer indicates an action supposed to be the best at a given moment. Such a winning node is then validated, and it can be turned off by a *reset signal*, produced by the action feasibility verifier, if a just chosen action cannot be safely executed due to a collision risk. When an action reset occurs, the present scene category is considered as *important*, and it temporarily enables adaptive capabilities of the *Fuzzy-ART* scene configuration partitioner. Then an action corresponding to a subsequent most excited associator node is considered. If all the associator's neurons get reset an emergency stop routine is performed.

2.5 Learning reflexes

The selected action is tested for its feasibility by the verifier module, which is sending a reset signal if the closest obstacle detected by sonars is closer than a certain distance threshold value.

A reset of a chosen action causes punishment of a weight w_{ik} ($w_{ik} \in [0,1]$) of the connection between i-th neuron of the FuzzyART output layer, representing a current spatial state category, and the considered, k-th neuron of the associator:

$$w_{ik}^{(new)} = w_{ik}^{(old)} - \beta,$$

where β ($\beta \in [0,1]$) is a learning speed parameter.

Feasible action is passed to a drive controller, and its efficiency (in terms of goal approaching) is estimated. Reinforcement of a weight w_{jk} ($w_{jk} \in [0,1]$) of the connection between active j-th node of the goal heading circuit *interface layer* and the winning k-th neuron of the associator, may be positive or negative, in accordance to the action profit:

$$w_{jk}^{(new)} = w_{jk}^{(old)} - P \cdot \beta,$$

where P ($P \in [-1,1]$) is the action profit coefficient.

A $\vec{w}_i$ vector, as well as $\vec{w}_j$ vector, are normalized (in the euclidean sense) after each change of their components values.

On the beginning all the associator input connection weights are initialized uniformly, so the system has no particular preferences.

A punishment of a certain weight value makes a corresponding association less likely to happen in the future, what prevents the robot from taking wrong decissions, while a reinforcement of a connection strength encourages the system to make successful choices again.

Vector normalisation provides the system capability to cope with changing, nonstationary environments. Setting a certain association as *"wrong"* is not definite, due to a nonmonotonical learning law it can become *"favourable"* again. In such a way the system selforganises its reactions to received stimuli.

Due to the normalisation the considered vectors remain unitary in length, despite subsequent adaptations of their components. It makes easy to arbitrate multiple reflexes influences, and gives a very clear geometric interpretation: while learning the endpoints of the $\vec{w}_i$ and the $\vec{w}_j$ vectors slide on a unitary hypersphere surface towards favourable associations, and consequently, pointer the best action to be taken in a certain situation.

3 Experiment

The concept was tested so far using a software simulator of the mobile platform.

The aim of the experiment conducted was to investigate an efficiency of the *learning from zero* process. The test trial was formed by a return trip of the robot from a start point A to a given point B. An example of a training task is shown in Fig. 2.

During training there were several trials on the same path performed, for each of them there were counted numbers of executed locomotion commands. We call the system more efficient if it needs less control

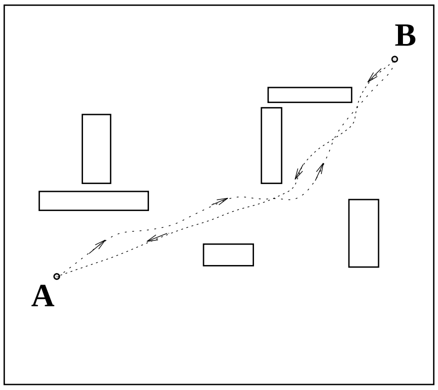

Figure 2: An example of a training task: a return path from A to B composes a training trial.

cycles to complete a trial. So, as a system's efficiency measure the *relative increment of experience* was chosen, calculated for an n-th training cycle as follows:

$$\Delta_i = 100\% \cdot \left(1 - \frac{N_n}{N_0}\right),$$

where N_0 is a number of actions executed during the first trial, and N_n is a number of actions needed to complete the n-th trial.

The efficiency dynamics i.e. the relationship between the experience measure values and the training time (trial number) is shown in Fig. 3. The results achieved for the different tasks (different training paths) are qualitatively very similar (refer solid, dashed and dotted curves in Fig. 3.). Size of the unstable part (X in fig. 3.) is strongly dependend on the *learning speed parameter* β. The larger β the larger X range.

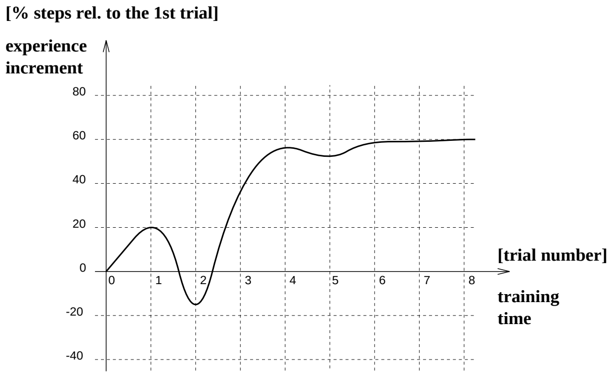

Figure 3: Experience dynamics observed while repeating typical training trials.

Generally similar character of the efficiency dynamics was observed during training a *virgin* system, and after major changes of parameters of the already experienced system. In both cases knowledge aquired by a *trial-and-error* method establishes a better steady state efficiency level (Y in fig. 3.), than it was either on the beginning of learning, or just after a disturbance of system's parameters.

Learning after a disturbance results in an increase of perceptual categories number, which embodies system's capabilities to deal with non-stationarities of the environment, or of its own structure. In both cases, an importance of the Fuzzy-ART feature of preserving already stored knowledge while incorporating new data, can be easily seen.

Experiments revealed expected system's failure, when facing a classical problem of reactive navigation (Fig. 4.). If the goal is just behind the wall and the the steering angle controller is symmetrical, i.e. left and right turns have equvalent priority, the robot falls in oscillations.

4 Conclusions

In reactive locomotion control system sensory data are directly mapped into drive actions. A chosen drive action is a reflexive, immediate reaction to the sensory data stimulus, there is no reasoning based on the world model or on the planned path.

The system presented in this article realises such a mapping using a four-level hierarchy in which the main blocks are the neural network modules: *Fuzzy-ART* scene configuration classifier and the certain model of a neural associative memory.

The system models and superposes two main reflexive behaviours of a mobile robot: *goal following* and *obstacle avoidance.*

The complete system explores its environment in an unsupervised manner, there is no need of preparing training data sets (as in [9]). It learns locomotion reflexes while responsing (*on-line*) using trial-and-error method.

Perceptual space partitioning method allows the system store newly recognised spatial situation classes incrementally (unlike [11]), and only when they are important, i.e. if they require reflexive avoidance reactions. Such an approach keeps a number of stored *Fuzzy-ART* categories in reasonable boundaries.

It was observed that the learning reflexes process becomes unstable in a case when a certain scene configuration category subsequently causes resets of the associated action and, in the following steps, allows the execution formerly reseted actions. That is the case of very narrow perceptual space regions, where it is rather difficult (a feature of a discrete representation) to tune corresponding parameters of classifiers and verifiers. However, after a sufficiently long learning time, the risk of unstable behaviours gets really low, as long as changes of the environment are insignificant. In an opposite case a technique of forcing more accurate partitioning of weak stability regions might help a lot.

The experiment shows that the method fails in some cases which are the classical problems of reactive navigation, as oscillations near local minima of control surface. For example a position shown in Fig. 4. leads to an infinite number of training trial steps. Failures like that were easy to predict, since the described system is unable to deal with sequences of situation-action-result, processing only subsequent static views of its environment.

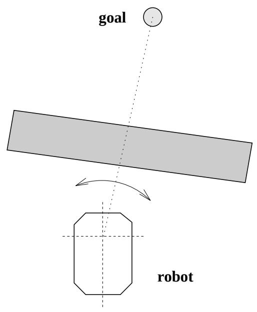

Figure 4: A position leading to infinite oscillations.

However, the particular aim of the experiment was to investigate an efficiency of the *learning from zero* process, and system's behaviour after some disturbances of its structure or the parameters of the environment. As promissing results authors found that the system provides the robot ability to remain effective after changes and to self-supervise the training.

To take into account time-sequential aspects of mobile robot control, several techniques may be considered. Future research directions include those of temporal action sequence analysis and temporal difference reinforcement learning.

Acknowledgement

Authors would like to thank Mr. Patrick Reignier of LIFIA INP Grenoble, for his valuable participation in discussions and indispensible assistance while implementing the presented system.

References

[1] Anderson T.L., Donath M. Animal Behavior as a Paradigm for Developing Robot Autonomy. *Robotics and Autonomous Systems*, 6(1990), pp. 145-168.

[2] Brooks R.A. A Robust Layered Control System for a Mobile Robot. *IEEE Journal on Robotics and Automation*, vol. 1., 1986.

[3] Carpenter G.A., Grossberg S., Rosen D. Fuzzy ART: Fast Stable Learning of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, Vol. 4, 1991.

[4] Crowley J.L. Coordination of Action and Perception for a Mobile Surveillance Robot. *IEEE Expert*, Winter 1987.

[5] Dubrawski A., Crowley J.L. Learning Locomotion Reflexes: A Self-Supervised Neural System for a Mobile Robot. *Robotics and Autonomous Systems*, 12(1994).

[6] Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. Journal of Robotic Research*, Vol. 5. no. 1., pp. 90-98, 1986.

[7] Koren Y., Borenstein J. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. *Proc. 1991 IEEE Int. Conf. on Robotics and Automation*, pp. 1938-1404, April 1991.

[8] Kosko B. *Neural Networks and Fuzzy Systems.* Prentice-Hall, 1992.

[9] Pomerleau D.A. Efficient training of an artificial neural networks for autonomous navigation. *Neural Computation*, Vol. 3, 1991.

[10] Reignier P. Fuzzy Logic Techniques for Mobile Robot Obstacles Avoidance. in: *Crowley J.L., Dubrawski A. (Eds.) Proc. Int. Workshop on Intelligent Robotic Systems, Zakopane, Poland, July 1993*, IPPT PAN and LIFIA IMAG, 1993.

[11] Sorouchyari E. Self-organizing neural network for trajectory control and task coordination of a mobile robot. *Connection Science*, Vol. 2, 1990.

[12] Zhang B., Grant E. Using Competitive Learning for State-Space Partitioning. *Proc. 1992 Int. Symposium on Intelligent Control Glasgow, Scotland,Aug. 1992*, pp. 391-396, IEEE Control Systems Society 1992.