Handbook of Brain Theory and Neural Networks, Second edition
M.A. Arbib (ed.), Cambridge, MA: MIT Press, 2002
http://mitpress.mit.edu

Robot Navigation

José del R. Millán

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)

Rue du Simplon 4

CH-1920 Martigny

Switzerland

Tel: +41 – 27 – 7217770

Fax: +41 – 27 – 7217712

Email: jose.millan@idiap.ch

Running Head: Robot Navigation

Introduction

Mobile robots are gradually leaving the laboratories to undertake service tasks ranging from surveillance of buildings and supervision of plants, to transport patients and delivery items, to cleaning houses and guiding people. Independently of the assigned task, the basic capability of a mobile robot is to move to its destination —or sequence of targets— efficiently (e.g., along short trajectories) and safely (i.e., without colliding). *Navigation* refers to the capability of selecting and performing a path from a current position to a desired location. Implicit in this definition is the ability to adapt the goal-oriented behavior to the complexity of the task. If a target location is either visible or identified by a landmark (or sequence of landmarks), a simple stimulus-response strategy can be adopted (see REACTIVE ROBOTIC SYSTEMS). However, targets are often neither visible nor identified by any sequence of cues. In this case, for a robot to navigate it must first determine its position with respect to the target. This is the *localization* problem. Finally, to perform more flexible and sophisticated navigation (e.g., planning short-cuts), the robot needs a model of the environment encoding the spatial relationships between locations. Acquiring such a model is the *map-building* problem.

As any other robotic system (see ROBOT ARM CONTROL), mobile robots must rely upon on-line sensory information to take actions. But, contrarily to most arm robots, sensory information cannot only be proprioceptive (i.e., odometry process that gives the robot's coordinates based on internal encoders); it must also provide exteroperception (i.e., information about the external environment). Indeed, odometry alone accumulates errors due to slippage, what will make the robot get lost and crash sooner or later. Mobile robots mainly use three types of sensors to perceive their surroundings, namely tactile sensors that inform about contacts, range sensors (lasers, ultrasounds, or infrareds) that return distances after appropriate transformations, and vision. Given the opposite strengths and weaknesses of the different sensors, an orthogonal issue not covered by this article is that of SENSOR FUSION (q.v.).

A common property shared by all types of sensors is their noisy responses. This sensor uncertainty, together with the inaccuracy of the robot's actuators and the unpredictability of real environments, make the design of mobile robot controllers a difficult task. The complexity of the sensorimotor mapping underlying robot navigation yields two main consequences. First, simulations, though useful, are not enough to reproduce the actual agent-environment interaction. Second, robots must build their control strategies based on their own sensory perceptions of the real world (i.e., *embodiment*). Human-made controllers are, except for simple tasks and environments, inadequate because the designer must anticipate every possible situation the robot might face and must tune the controller's parameters to achieve efficient performance. An alternative is to endow robots with *learning capabilities* in order to acquire autonomously their control system and to adapt their behavior to never experienced situations (i.e., *generalization*).

Artificial neural networks (NN) offer a suitable learning framework to model the basis of adaptive behavior. Indeed, their noise robustness and generalization capabilities allow robots to cope with the nature of their interaction with the world and to build appropriate sensorimotor mappings. The next three sections discuss NN approaches to localization, map building, and navigation.

Localization

To solve complex navigation tasks, mobile robots must self-localize in the environment by relying upon their exteroceptive and proprioceptive sensory inputs. In general, localization calls upon *place recognition*. To localize itself, the robot can either simply memorize the sensory perceptions observed during exploration or learn a more complex representation (map) encoding spatial relationships between experienced local perceptions. How the robot acquires a map is discussed in the next section.

Strictly speaking, only Thrun (1998a; 1998b) uses a NN for localization. In the remaining approaches, the robot's perception is matched against the model, which has a NN organization, and its location is derived from that associated to the winning unit. Actually, the robot's perception can be the current sensory reading plus odometry information (Zimmer, 1996), sensory data averaged as the robot moves (Matarić, 1992), or egocentric views obtained from the sensory perceptions (Recce and Harris, 1996).

An alternative is to transform raw sensory data into a more reliable representation through NN (Thrun, 1998a). In particular, a feedforward NN is trained through *backpropagation* to generate a local *occupancy grid* from the current sensory perception. Such a grid is a discrete representation of the space around the robot, where each cell has a value that estimates the occupancy probability of the corresponding area of the world. After exploration, the localization algorithm searches for the previously stored grid which best matches the current local map. Two are the advantages of using a neural sensor interpretation to build local occupancy grids: the NN does not assume any noise distribution and it interprets sensor readings simultaneously. On the other hand, the shortcoming of this approach is its computational cost as building a $n*n$ grid requires $n*n$ calls to the NN.

A totally different approach is to learn what environmental features are the most relevant landmarks for localization. Thrun (1998b) trains a feedforward NN to optimize a Bayesian measure of probabilistic localization. Training is done on samples collected during an exploration phase where each sample consists of a sensory perception and its location. During operation, the robot averages the NN response for the k nearest neighbor samples to its estimated location. This approach has demonstrated its superiority to hand-coded localization methods based on using doors and ceiling lights as landmarks.

Map Building

The representations a robot may learn are of two main types, namely *metric* and *topological*. In the former, maps quantitatively reproduce the geometric and spatial features of the environment. This is computationally expensive and vulnerable to errors that affect the metric information. Topological maps are more qualitative and consist of a graph, where nodes represent perceptually distinct places (landmarks) and arcs indicate spatial relations between them. They are less vulnerable to sensory errors, and enable fast planning since the latter reduces to a simple search process in a graph. However, topological representations rely upon the existence of ever-recognizable landmarks.

One of the most popular approaches to build metric maps is based on the use of occupancy grids. Thrun (1998a) trains a feedforward NN to create a local occupancy grid modeling the space surrounding the robot. Successive local grids generated as the robot explores its environment are then combined to produce an accurate global metric map. Once the global metric map is available, a topological graph can be abstracted off-line, what greatly reduces the cost of planning paths between different locations in the environment. This approach (in conjunction with the localization process discussed before) is implemented in museum tour-guided robots. An alternative is to use the same neural sensor interpretation but only for deriving coarse geometrical features from which to build up on-line a variable-resolution partitioning of the environment (Arleo, Millán, and Floreano, 1999). The environment is discretized in cells of different sizes, with a high resolution only on critical areas (i.e., around obstacles). The resulting map combines geometrical and topological aspects that are learned simultaneously.

Among topological approaches, Matarić's model (1992) builds a sparse graph in which each node represents a unique predefined landmark. Spatial relationships between landmarks are encoded by neighbor links in the graph which produces a structure isomorphic to the topology of the environment. Disambiguation between similar sensory patterns is done by spreading expectations from the currently active unit to its neighbors (contextual discrimination) or by attaching metric information to the units. SELF-ORGANIZING FEATURE MAPS; KOHONEN MAPS (q.v.) provide an alternative way of acquiring topological maps (Kurz, 1996; Zimmer, 1996). A self-organizing, or *Kohonen*, map clusters the sensory perceptions gathered during the exploration phase of the robot (Kurz, 1996). The dimensionality of the Kohonen map matches the robot's degrees of freedom, either two or three depending on whether or not the robot moves with a constant orientation. As a result, neighboring units in the learned Kohonen map correspond to neighboring areas in the sensory space. The problem is that there is no guarantee that neighbor areas in sensory space are also close in metric space. Still worse, due to the limitations of its sensors a robot may have similar sensory perceptions from two different metric locations. A possible solution is to include odometry information in the self-organizing process (Zimmer, 1996), what presumes a reliable localization process. Another solution is to use the temporal sequence of sensory perceptions and not just the current one. This can be achieved by means of a recurrent Kohonen map. Finally, instead of using a self-organizing map with a fixed structure (dimensionality and number of units), it is also possible to learn the topological map of the environment by means of a *dynamic self-organizing map*. This method adds a new unit whenever the current sensory perception is sufficiently different from any existing unit (Millán, 1997; Zimmer, 1996), or using statistical measures (Zimmer, 1996). In this kind of network the topology of the environment is kept in the links between units. It is worth noting that the ADAPTIVE RESONANCE THEORY (q.v.) can yield quantizations of the environment similar to a dynamic self-organizing map, but the resulting map does not exhibit topological relationships between the units.

Map learning systems engineered so far are not as robust, flexible, and adaptable as biological spatial-learning solutions. Neurophysiological findings suggest that spatial memory of mammals is supported by location-sensitive neurons (*place cells*) in the *Hippocampus* (see HIPPOCAMPUS: SPATIAL MODELS). Recent research in robot navigation has moved towards biologically inspired approaches to develop autonomous systems that mimic mammalian spatial learning capabilities. For example, Recce and Harris (1996) put forward a map building model that ascribes the spatial memory function to the hippocampus. The authors assume that a place cell in the robot's hippocampus memorizes a complete egocentric map of the environment. This is a strong requirement for both robots and animals, especially if operating in middle- and large-scale environments. Arleo and Gerstner (2000) propose a hippocampal model in which unsupervised *Hebbian* learning is applied to acquire a spatial map incrementally and on-line. The representation consists of a population of localized overlapping place fields that provide a stable coarse space code. The robot establishes place fields by extracting spatio-temporal properties of the environment from visual inputs and solves visual ambiguities by taking into account proprioceptive self-motion signals.

Navigation

If the robot has acquired (or is given) a topological map, then whenever it is requested to navigate to a given destination it simply searches an optimal route in the graph and uses elemental *behaviors* to move from a node to the next along that route. However, building and maintaining consistent global maps of the environment is far from trivial since noisy sensory data may introduce errors into the maps. Also, unless the robot is equipped with good exploration strategies it may fail to model the whole environment and topological relationships. For example, most map building approaches rely on a wall following (and obstacle avoidance) behavior that prevents the robot to visit open as well as cluttered areas. Thus, while in operation, the robot will never take shortcuts. Alternatively, the robot can directly use behaviors to reach its destination without resorting to any map. In this section we discuss how the necessary behaviors can be learned by means of NN.

A behavior is a set of perception-action rules that provide the robot with a given functionality such as obstacle avoidance (see REACTIVE ROBOTIC SYSTEMS). Perception-action mappings can be learned off-line from representative training sets, mainly through *supervised* techniques (Pomerleau, 1993; Sharkey, 1998). Pomerleau's system (1993) is a paradigmatic example of the potentiality of the supervised approach. He trains a feedforward NN to drive a car in a variety of roads. Training data are gathered by observing a human expert while driving the vehicle. In particular, inputs correspond to images of the road in front of the car and desired outputs to the driver's steering direction. After learning, the NN controller makes the vehicle follow the road by keeping it in the center of the lane. Instead of requiring a human for data collection, an alternative is to use a preprogrammed controller as an initial teacher. Sharkey (1998) makes feedforward NN learn from an initial behavior-based controller to approach a target while avoiding obstacles. The final neural controller, obtained through a bootstrapping process, performs better than the original controller does. Nevertheless, the resulting network also inherits limitations of the initial controller. This illustrates one of the fundamental limitations of supervised learning and leads to the necessity of *autonomous* robot learning.

Autonomous robots must train themselves on-line in order to cope with weak and incomplete training examples. REINFORCEMENT LEARNING (q.v.) is an appropriate paradigm to achieve this. A reinforcement-based robot can improve its performance over time without needing extensive previous knowledge about the task. This is quite appealing but, on the other hand, makes the learning process very slow. In the sequel we will describe several extensions to the basic reinforcement learning framework that speed up considerably the convergence to suitable sensorimotor mappings (or policies as customarily called in the control and reinforcement learning fields), so making possible to build practical learning mobile robots.

Lin (1991) combines *Q-learning* (a widely studied reinforcement learning technique) and teaching. The controller has one feedforward network for each discrete action the robot can perform. The input to each NN is the current sensory perception and the output is a prediction of the *Q-value* of that perception-action pair. The robot normally takes the action with the highest Q-value (see Q-LEARNING FOR ROBOTS). Now, a human teacher brings the robot to its goal several times along efficient paths. Then, the robot learns the appropriate Q-values, and hence good policies, from these examples. The taught navigation sequences help reinforcement learning by biasing the search for suitable actions toward promising parts of the action space.

Thrun (1995) also uses Q-learning, but integrates it with explanation-based learning (EBL). EBL requires a domain theory that is previously learned by a set of feedforward NN (action models) in a supervised manner, one network for each discrete action the robot can perform. Each network receives the current sensory perception and predicts the next perception and Q-value. In addition to action models, the robot has also a Q network per action similar to that of Lin (1991). As before, the Q networks encode the control policies. Finally, for each actual sequence of actions taking the robot either to the goal or to a failure, the robot explains the observed example in terms of its domain theory by computing the derivatives of the policy with respect to the action model networks. These derivatives are used to bias the supervised learning of the policy (i.e., the Q networks). It is worth noting that since the action models are task-independent, they are learned once and can be used across the different tasks faced by the robot.

The previous reinforcement-based robots perform discrete actions, while for smooth operation they should take continuous actions. Millán (1996; 1997) has implemented *Actor-Critic* architectures instead of Q-learning for this purpose. Key components of his learning architecture are the use of *local networks* and the incorporation of *bias* into the network. Local networks make the robot learn incrementally new sensorimotor rules (or tune existing ones) without degrading the performance of other rules. The robots use built-in reflexes (basic domain knowledge) as bias. Two are the benefits of bias. First, it accelerates the learning process since it focuses the search process on promising parts of the action space immediately. Second, it makes the robot operational from the very beginning and increments the safety of the learning process. The NN is trained on-line by means of a combination of reinforcement learning and self-organizing rules. Every time the robot fails to generalize its previous experience to the current sensory perception, it uses the reflexes and adds a new unit to the network. The robot may also add a new unit whenever it receives an advice from humans. This unit is integrated into a dynamic self-organizing map and associates a region around the current perception to either the computed reflex or the advice. The resulting sensorimotor rule is then tuned by means of reinforcement learning and self-organizing rules. Experimental results show that a few minutes suffice for the robot to navigate efficiently in office environments of moderate complexity where the robot can easily get trapped inside concave areas.

Recently, reinforcement learning has been shown to be a suitable framework to model reward-based navigation in animals. For example, Arleo and Gerstner (2000) employ Q-learning in continuous space to drive action units one synapse downstream from their hippocampal place cells. Due to the coarse space code provided by the localized overlapping place fields, Q-learning converges in few trials, which is consistent with the rapid acquisition of goal-oriented behavior of animals. Several action modules share the same space representation and guide the robot to multiple targets.

Discussion

For mobile robots to undertake real-world tasks with unreliable sensors and actuators, whose response greatly depend on the specific working environment, they must exhibit adaptive capabilities. NN naturally cope with the learning task of analyzing the perception-action interactions for navigation, localization and map building. Even though different NN approaches have solved some instances of these three aspects of robot navigation, it does not exist a complete navigation system that is purely made of neural components. From an engineering standpoint, such a complete mobile robot must incorporate other types of learning techniques to generate more abstract models of its perceptions, actions and sensorimotor rules (Kaiser et al., 1995). In addition, the engineering perspective calls for combining learning capabilities with alternative techniques to build successful mobile robots such as those of Thrun and coworkers (Thrun, 1998a). On the other hand, a different perspective looks at animals for inspiration to develop the necessary neural components of a complete robot navigation system. There are numerous efforts along this *bio-inspired* direction (see BIOLOGICALLY INSPIRED ROBOTICS and NEUROETHOLOGY, COMPUTATIONAL) in which reinforcement-based learning (see REINFORCEMENT LEARNING and Q-LEARNING FOR ROBOTS) and hippocampal models (see COGNITIVE MAPS and HIPPOCAMPUS: SPATIAL MODELS) are keystones of this type of future intelligent (because adaptive) mobile robots.

References

Arleo, A., Millán, J. del R., and Floreano, D., 1999, Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation, IEEE Trans. Robot. Automat., 15:990–1000.

Arleo, A, and Gerstner, W., 2000, Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity, Biol. Cybern., 83:287–299.

* Kaiser, M., Klingspor, V., Millán, J. del R., Accame, M., Wallner, F., and Dillman, R., 1995, Using machine learning techniques in real-world mobile robots, IEEE Expert, 10:37–45.

Kurz, A., 1996, Constructing maps for mobile robot navigation based on ultrasonic range data, IEEE Trans. Syst. Man Cybern.—Part B, 26:233–242.

Lin, L.-J., 1991, Programming robots using reinforcement learning and teaching, in Proceedings of the Ninth National Conference on Artificial Intelligence, Menlo Park: AAAI Press/MIT Press, pp. 781–786.

Matarić, M.J., 1992, Integration of representation into goal-driven behavior-based robots, IEEE Trans. Robot. Automat., 8:304–312.

Millán, J. del R., 1996, Rapid, safe, and incremental learning of navigation strategies, IEEE Trans. Syst. Man Cybern.—Part B, 26:408–420.

Millán, J. del R., 1997, Incremental acquisition of local networks for the control of autonomous robots, in Proceedings of the Seventh International Conference on Artificial Neural Networks, Heidelberg: Springer-Verlag, pp. 739–744.

Pomerleau, D.A., 1993, Neural Network Perception for Mobile Robot Guidance, Boston: Kluwer.

Recce, M., and Harris, K.D., 1996, Memory for places: A navigational model in support of Marr's theory of hippocampal function, Hippocampus, 6:735–148.

Sharkey, N.E., 1998, Learning from innate behaviors: A quantitative evaluation of neural network controllers, Auton. Robots, 5:317–334.

Thrun, S., 1995, An approach to learning mobile robot navigation, Robot. Auton. Syst., 15:301–319.

* Thrun, S., 1998a, Learning maps for indoor mobile robot navigation, Artif. Intell., 99:21–71.

Thrun, S., 1998b, Bayesian landmark learning for mobile robot localization, Machine

Learn., 33:41–76.

Zimmer, U.W., 1996, Robust world-modelling and navigation in a real world, Neurocomputing, 13:247–260.