

Natural inspiration for artificial adaptivity: some neurocomputing experiences in robotics^{*}

Carme Torras

Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
Llorens i Artigas 4-6, 08028-Barcelona, Spain.
torras@iri.upc.edu, <http://www-iri.upc.es/people/torras>

Abstract. The biological world offers a full range of adaptive mechanisms, from which technology researchers try to get inspiration. Among the several disciplines attempting to reproduce these mechanisms artificially, this paper concentrates on the field of Neural Networks and its contributions to attain sensorimotor adaptivity in robots. Essentially this type of adaptivity requires tuning nonlinear mappings on the basis of input-output information. Several experimental robotic systems are described, which rely on inverse kinematics and visuomotor mappings. Finally, the main trends in the evolution of neural computing are highlighted, followed by some remarks drawn from the surveyed robotic applications.

1 Introduction

Why the use of robots is not as widespread as some envisaged they would be by now? At the risk of oversimplification, I would say that it is due to their lack of adaptivity, at all levels. This capability is dispensable in well-engineered environments, and thus we have very performant robots in manufacturing lines, but it is a *sine qua non* when tasks are to be carried out in non-predefined worlds.

In this sense, the biological world –where adaptivity is crucial for survival– constitutes a very good source of inspiration for robotics researchers, since it provides existence proofs of many adaptive mechanisms that do function. However, caution must be taken, because the best natural solution may not be the best artificial one [40]. Wheels, wings and calculators have often been mentioned as examples of artificial solutions considerably different from their natural counterparts, and more performant according to certain criteria. The resources available to engineering design depart a lot from those in nature, and not just when it comes to materials, but also in the number of instances and spendable time.

With this note of caution in mind, i.e., accepting that biological plausibility in itself adds no special value from an engineering viewpoint, it is safe to look into natural adaptivity to get seed ideas that can be instantiated in a different way by artificial means.

^{*} A more detailed version of this review, although less up to date, can be found in [48].

2 Natural and artificial adaptivity

We refer to adaptivity as the capability of self-modification that some agents have, which allows them to maintain a level of performance when facing environmental changes, or to improve performance when confronted repeatedly with the same situation. The term ‘agent’ above stands for a single cell, an organ, an individual or even a whole society, because, in the biological world, adaptivity occurs at several levels, each having a possible counterpart in the design of autonomous robots [42, 14, 53].

At the cell level, several chemical and electrical mechanisms of **plasticity** have been discovered, some of which have been modelled and analysed within the Neural Modelling field, and later applied to adjust the parameters of robot sensors and actuators. See the chapters on ‘neural plasticity’ in [3].

When referring to individuals, adaptation is usually called **learning** and it takes two rather different forms depending on whether it occurs at the sensorimotor or cognitive levels. Sensorimotor adaptation consists in building relevant associations between stimuli and responses, while cognitive learning entails constructing symbolic representations to guide decision-making. Two disciplines have tried to mimick these two capacities. Neural Networks, closer to Biology, has proven adequate to handle the massively-parallel tasks of perception and control of action, while Artificial Intelligence, stemming from Computer Science and Cognitive Psychology, has developed the necessary data structures and procedures to tackle symbolic learning [46, 47]. Results in both disciplines have been applied to Robotics, the former to attain adaptive robot sensorimotor mappings [24], while the latter have led to so called learning robots [49, 9, 22, 23].

Finally, at the species level, adaptation is attained through **evolution**. Genetic algorithms [11, 17] and evolutionary computation [4, 5] are starting to be used to tailor robot genotypes to given tasks and environments [15].

Table 1 summarizes the different adaptation levels and the involved disciplines.

ADAPTATION LEVEL	TYPE OF ADAPTIVITY	“ARTIFICIAL” DISCIPLINE
Cell	Plasticity	Neural Modelling
Sensorimotor	Associative learning	Neural Networks
Cognitive	Symbolic learning	Artificial Intelligence
Species	Evolution	Evolutionary Algorithms

Table 1. Levels of adaptation and related disciplines

In this paper we concentrate on adaptivity at the individual sensorimotor level, i.e. both the robot morphology and its components are assumed to be fixed and what may change with experience is the functional relationship between sensors and actuators.

3 Natural inspiration for artificial neural adaptivity

Neural networks are essentially procedures for approximating nonlinear mappings given a set of inputs and some information on the corresponding outputs. The approximation is attained by iteratively tuning the weights of the connections between neurons. This iterative process is referred to as neural adaptivity, leading to the desired input-output behaviour of the network.

The approaches to adaptivity pursued within the Neural Networks field have their roots in the learning paradigms developed in the domain of Behavioural Psychology (refer to Figure 1). This is the reason why the rules to attain neural adaptivity are usually called learning rules. The role of the animal in the behavioural learning experiments is played here by the neuron. It is worth noting that, although inspiration comes from the biological world, the artificial techniques are here applied not only to a different physical substrate, but also at a different level (neuron instead of animal).


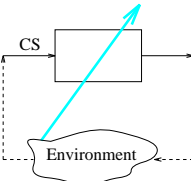
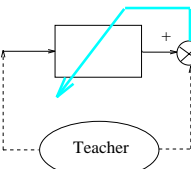
<u>TYPE OF LEARNING</u>	<u>DEGREE OF FEEDBACK</u>	<u>DIAGRAM</u>	<u>KEYWORDS</u>	<u>NEURAL RULES</u>
CLASSICAL CONDITIONING	none		Pavlov Unsupervised Open-loop	CORRELATIONAL
INSTRUMENTAL CONDITIONING	qualitative		Skinner Trial-and-error Optimal control	REINFORCEMENT
INPUT-OUTPUT TEACHING	quantitative		Supervised Reference-model control Closed-loop	ERROR- MINIMIZATION

Fig. 1. Learning paradigms inspiring neural adaptivity. In the diagrams, US stands for unconditioned stimulus, CS for conditioned stimulus and R for response.

The most basic learning paradigm is **classical conditioning**, as introduced by Pavlov [26], which consists of repeatedly presenting to an animal (e.g. a dog) an initially meaningless stimulus (e.g. the sound of a bell) together with an unconditioned stimulus (e.g. food) that triggers a reflex response (e.g. salivation). As a result of such paired presentations, the animal builds up an association so that, when presented with the conditioned stimulus (e.g. the sound of a bell)

alone, it produces the same response as before (e.g. salivation). This type of learning is completely open-loop, in the sense that it entails no feedback. The neural learning rules mimicking this type of conditioning come from the classical Hebbian rule [13], in which a connection weight is adjusted according to the correlation between the activation of the two neurons connected, this being the reason why they are called *correlational rules*. Neural models incorporating these type of rules are Self-Organizing feature Maps (SOM) [16], the Cerebellar Model Articulation Controller (CMAC) [1], and Adaptive Resonance Theory (ART) [12].

Instrumental conditioning was introduced by Skinner [41] and requires that the animal under experimentation performs an arbitrary action (e.g. pressing a lever, walking around) in the presence of an initially meaningless stimulus (e.g. a flickering light). If the action is “appropriate” to the given stimular situation, the animal receives a reward (e.g. food). Otherwise, it receives nothing or punishment, depending on the particular experimental design being applied. Thus, this learning paradigm strongly relies on providing the animal with a reinforcement signal dependent on the action performed. This can be conceptualized as a qualitative feedback. The neural learning rules implementing this type of conditioning at the neuron level are known as *reinforcement-based rules* [43].

Note that the natural progression in the degree of feedback supplied suggests the use of a quantitative error signal to guide learning. This is represented in the third row of Figure 1 under the name of **input-output teaching**. Here, after presenting an input to the system and observing the emitted response, a teacher supplies the desired output whose difference with the emitted one provides the error signal which is fed back to the system. This is an entirely closed-loop learning process that requires perfect knowledge of the input-output pairs to be associated. The most widely used neural learning rules follow this scheme and we call them *error-minimisation rules*. They are mostly variants of the well-known backpropagation procedure [19, 39] aimed at palliating its main drawbacks, namely catastrophic forgetting, overfitting, and a slow convergence rate.

Several techniques to prevent catastrophic forgetting [30] by explicitly minimising degradation of the previously learned patterns while encoding a new one [25, 28] and by introducing noise [2, 35] have been devised. Overfitting, i.e., the problem of learning a function too tailored to the samples and thus yielding a high generalisation error, is usually addressed by using methods for model complexity control [6, 7] and, in particular, regularisation. An interesting observation is that many such methods lead to functional invariance [31, 32, 36], i.e., they converge to the same function irrespective of network size for fixed regularisation parameters.

4 Robot sensorimotor mappings

Motion control, both in biological and technological systems, relies strongly on sensorimotor mappings. These mappings vary considerably [45], depending not

only on the nature of the involved sensors and actuators, but also on the goal pursued.

Tasks to be carried out by robots are usually specified in world coordinates (or, alternatively, in terms of sensor readings), while robot moves are governed by their actuator's variables. For instance, a sealing task may be specified as a given curve in 3D space or as following a prespecified visual pattern, but it has ultimately to be translated into currents sent to the motors governing the different joints. Therefore, robot control critically depends on the availability of accurate mappings from physical space or sensor space to joint space or motor space. The discussion in what follows is centered on mappings required for robot arms to work, but similar arguments apply to the case of mobile robots [20, 21].

For a gripper to reach a desired position and orientation in space, the robot controller must access a mapping relating workspace coordinates to joint coordinates. This is called *inverse kinematics mapping*, because the natural (direct) map is that relating the values of the joint coordinates defining an arm configuration to the position and orientation of its end-effector (hand, gripper,...) in the workspace.

If a desired end-effector trajectory is specified instead, then the controller should resort to an *inverse dynamics mapping* relating such trajectory to the forces and torques that need to be exerted at the different joints to realize it. Note that this mapping, which is again called inverse for the same reason above, cannot be characterized uniquely in terms of inputs and outputs, it being instead dependent on state variables (or the short-term history of inputs) as it is usually the case with dynamic systems.

For tasks entailing the achievement of a goal using sensory feedback, programming even in terms of the coordinates of the workspace can be very complex. An example of this is the visual inspection of large objects that cannot be precisely placed (e.g., aircraft wings), since devising a detailed vision-based control strategy that moves the camera to the same relative position with respect to the object in all possible situations, and subject to real-world conditions of uncertainty and noise, is extremely difficult. What is needed to accomplish this type of tasks is an appropriate *sensorimotor mapping* relating sensory patterns to motor commands.

The diversity of the aforementioned mappings sometimes hides what they have in common: an underlying highly nonlinear relation between a continuous (often hard to interpret) input domain and a continuous motor domain; a relation that is very difficult (when not impossible) to derive analytically. Furthermore, because of environmental changes or robot tear-and-wear, the mappings may vary in time and then one would like the controller to adapt to these variations, without any human intervention if possible. Therefore, a way of learning (or tuning) these mappings automatically while robots move is highly desirable. Since, as we have mentioned, neural networks are essentially procedures for approximating nonlinear mappings, they constitute a good tool to attain the desired adaptivity.

In what follows we will describe some experiences related to the learning of two of the mappings mentioned above, namely inverse kinematics and visuomotor mappings.

5 Adaptive inverse kinematics

Making robots adaptive to changes in their own geometry (e.g., link bending, encoder shifting and other wear-and-tear deformations occasioned by regular functioning) would certainly widen their range of application. Since these geometric changes affect the robot inverse kinematics, the interest of using neural networks to approximate such mapping has been widely recognized. Especially when the operation conditions of the robot (in space, underwater, etc.) make it very hard to recalibrate it.

Along this line DASA (Daimler-Benz Aerospace S.A.), in the framework of the Advanced Servicing Robot project targeted at unmanned space stations, proposed an application of maintenance of electronic equipment that required the automatic recalibration of a 6-dof robot in-situ, since recalibration through teleoperation from earth is a very time-consuming task due to communication delays. After reviewing previous approaches to the learning of inverse kinematics, we will present the solution implemented in the REIS robot included in the space-station mock-up located at DASA's R&D laboratory, in Bremen, Germany (see Figure 2).

The conclusion reached after extensive experimentation with feedforward networks using backpropagation [18, 46] is that a coarse mapping can be obtained quickly, but an accurate representation of the true mapping is often not feasible or extremely difficult. The reason for this seems to be the global character of the approximation obtained with this type of networks using sigmoid units: every connection weight has a *global* effect on the final approximation that is obtained [18].

An obvious way to avoid this global effect is of course using local representations, so that every part of the network is responsible for a small subspace of the total input space. Thus, Ritter et al. [27] have used a self-organizing map (SOM) together with an error-minimisation rule to learn the visuomotor mapping of a robot arm with three degrees of freedom (dof) in 3D space. The target position of the end-effector is defined as a spot registered by two cameras looking at the workspace from two different vantage points. Neurons are arranged in a 3D lattice to match the dimensionality of physical space. It is expected that learning will make this lattice converge to a discrete representation of the workspace.

Extensive experimentation by Ritter et al. [27] and other groups has shown that the network self-organizes into a reasonable representation of the workspace in about 30.000 learning cycles. This should be taken as an experimental demonstration of the powerful learning capabilities of this approach, because the conditions in which it is made to operate are the worst possible ones: no a priori knowledge of the robot model, random weight initialization, and random sampling of the workspace during training.



Fig. 2. Space station mock-up at Daimler-Benz Aerospace, Bremen.

This basic model has been extended in three directions to cope with higher-dof robots. First, a hierarchical version, consisting of a 3D SOM whose nodes have associated a 2D SOM each, was applied to a 5-dof robot. The 3D net encodes the workspace as before, while each 2D subnet approximates the end-effector orientation space at the corresponding position [27].

Ruiz de Angulo and Torras [29] adapted this hierarchical model to suit a practical setting. Thus, instead of learning the kinematics from scratch, only the deviations from the nominal kinematics embedded in the original robot controller are learnt. This, together with informed initialization and sampling, as well as several modifications in the learning algorithm aimed at improving the cooperation between neurons, led to a speed-up of two orders of magnitude with respect to the original model.

The resulting algorithm constitutes the core of the recalibration system that was installed in the REIS robot included in the space-station mock-up located at DASA, as mentioned above. Figure 2 is a photograph of such a set-up, where the different racks containing the electronic cards that the robot should maintain are shown. The robot must reach the handles of the racks with enough precision to be able to pull them out and, afterwards, extract a faulty card in order to replace it by another one. Although testing in this set-up has been constrained by the need to preserve robot integrity, the system has proven able to correct large miscalibrations of the robot: 95% of the decalibration was corrected with

the first 25 movements, this percentage raising to 98% after 100 movements [29]. Moreover, other desirable features in stand-alone applications, such as parameter stability, are guaranteed.

The third extension of the basic model relies on the generalisation of SOMs to parameterized SOMs (called PSOMs). The idea is to turn the discrete representation into a continuous one by associating a basis function to each neuron, so that a parameterized mapping manifold is obtained. Moreover, PSOMs make no distinction between inputs and outputs, thus encoding bidirectional mappings. The PSOM reduces considerably the number of training samples required to attain a given precision as compared to the SOM [50], allowing the learning of the full inverse kinematics of a 6-dof robot with less than 800 movements.

The main drawback of using neural networks to approximate the inverse kinematics of robot arms is precisely the high number of training samples (i.e., robot movements) required to attain an acceptable precision. A trick has been proposed [33, 37], valid for most industrial robots, that greatly reduces the number of movements needed to learn or relearn the mapping to a given accuracy. This trick consists in expressing inverse kinematics as a composition of learnable functions, each having half the dimensionality of the original mapping. A training scheme to learn these component functions has also been proposed. Experimental results obtained by using PSOMs, with and without the decomposition, show that the time savings granted by the proposed scheme grow polynomially with the precision required.

The aforementioned trick assumes that the last three robot joints cross at a point, a condition satisfied by some classic robot architectures, but not by other more innovative ones. Therefore, a more general decomposition technique applicable to any serial robot has recently been developed [38], which still retains the main advantage of the trick above: The input dimensionality of each of the tasks resulting from the decomposition is half that of the original one. Thus, for a given desired accuracy, if the number of training samples required to learn inverse kinematics directly is $O(n^d)$, through the decomposition it reduces to $O(n^{d/2})$. This yields an enormous reduction in the number of samples required for high-precision applications.

The development of humanoid robots has recently raised the interest in learning inverse kinematics. Due to the many dof's involved, the aim is no longer learning the mapping for the whole workspace, but focussed on a specific trajectory. Following the trend of using localized representations, D'Souza et al. [8] have applied a supervised algorithm –locally weighted projection regression– in this context, with promising results.

6 Adaptive visuomotor mappings

Depending on the task to be performed and the camera-robot arrangement, visuomotor mappings take different forms. Thus, in eye-hand coordination, where cameras external to the robot are used to monitor the pose (position and orientation) of its end-effector, a mapping from the camera coordinates of a desired

end-effector pose to the joint angles that permit attaining that pose is sought. This mapping is closely related to the inverse kinematics one, especially if the camera coordinates of selected points in the end-effector uniquely characterize its pose. Therefore, the same models used to learn inverse kinematics have been applied to the learning of the visuomotor mapping underlying eye-hand coordination [27].

A camera mounted on a robot arm is used in tasks such as visual positioning and object tracking. The goal of these tasks is to move the camera so that the image captured matches a given reference pattern. The target is thus no longer a position of the robot in space but a desired image pattern, and the desired visuomotor mapping needs to relate offsets w.r.t. that pattern with appropriate movements to cancel them. In visual positioning, the scene is assumed to be static and the main issue is to attain high precision. Applications include inspection and grasping of parts that cannot be precisely placed (e.g., in underwater or space settings). The aim of object tracking is to maintain a moving object within the field of view, speed being here the critical parameter instead of precision.

The classical way of tackling these tasks consists of defining a set of image features (corners, holes, etc.) and then deriving an interaction matrix relating 2D shifts of these features in the image to 3D movements of the camera [10]. The advantages of applying neural networks to this task are the direct learning of the interaction matrix, as well as avoiding the costly matching of features in the current and reference images.

The latter approach has been used in an application developed for Thomson Broadcast Systems [52] for the inspection of large objects (e.g., ship hulls, airplane wings, etc.). Since these objects are difficult to move, it is the camera that has to travel to attain a pre-specified position and orientation with respect to the object. The developed camera control system consists of a feedforward network trained with backpropagation. The training procedure consists of moving the camera from the reference position to random positions and then using the displacement in image features together with the motion performed as input-output pairs. In operation, the robot is commanded to execute the inverse of the motion that the network has associated to the given input.

The key option in this work is the use of global image descriptors, which permits avoiding the costly matching of local geometric features in the current and reference images. By using a statistical measure of variable interdependence (the mutual information criterion), sets of global descriptors as variant as possible with each robot dof are selected from a battery of features, including geometric moments, eigenvectors, pose-image covariance vectors and local feature analysis vectors [51]. The results obtained with a 6-dof show that, after 10.000 learning cycles, translation and rotation errors are lower than $2mm$ and 0.1 degrees, respectively. Figure 3 shows the robot-mounted camera and the reference image of an object to be inspected (a water valve), together with several snapshots along the visual positioning process. In this case, the silhouette of the object could be readily extracted and 32 Fourier descriptors coding it were used as



a) Robot and camera



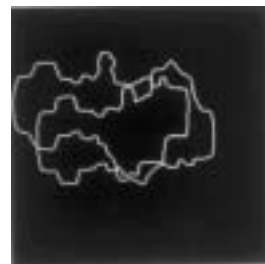
b) Reference image



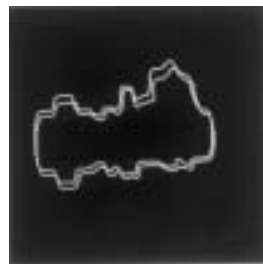
c) Initial image



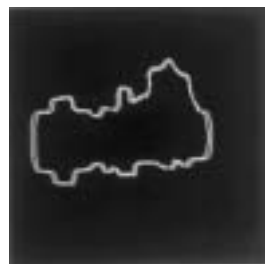
d) Initial contours



e) After 1 movement



f) After 5 movements



g) After 7 movements



h) Final image

Fig. 3. Visual positioning system developed in collaboration with Thomson Broadcast Systems.

image features. It can be observed that, after 7 movements, the captured image is practically registered with the reference one.

7 Conclusions

This paper has reviewed the ways in which neural computing may help to increase sensorimotor adaptivity in robots. The mechanisms of neural adaptivity have been inspired in the learning paradigms of Behavioural Psychology (classical and instrumental conditioning), and fall into three classes that require progressively more feedback: correlational, reinforcement and error-minimisation rules.

Some trends in the development of these learning rules deserve notice, since they have parallels in other disciplines dealing with adaptivity at different scales, such as Evolutionary Computation and Artificial Intelligence. The first trend is that of progressing from binary variables to continuous ones. This entails moving from discrete search spaces and classification tasks to manifold representations and function approximation. Then, issues such as model complexity control [7] and functional invariance [36] become very important.

A second trend is that of progressively replacing local feedback for more global one, this globalisation taking place both spatially and temporally. The first learning rules proposed required feedback to be supplied to each single neuron. Backpropagation made a big step forward in allowing feedback to be supplied at the overall network level (spatial globalisation). Reinforcement learning has greatly contributed to dealing with deferred feedback (temporal globalisation).

The dichotomy between locality and globality appears also in the state space representation. Correlational rules are often incorporated into network models that build localised representations (such as SOM, ART and CMAC), while the strength of most models based on error-minimisation and reinforcement rules lies precisely in the distributed (global) way in which information is represented across all the network weights. In the localised representations, appropriately tuning the neighbourhood size is a key issue.

Moving from off-line to on-line learning is another trend observed in neural computing. Initial learning procedures were designed to work in a batch mode (with all training samples supplied at the same time), while a later challenge was to incorporate new samples into an already trained network. Sequential learning addresses this challenge by explicitly seeking to avoid catastrophic forgetting [34].

Finally, let us mention the important role that randomness plays in learning. This has been widely acknowledged in many contexts, but specifically in neural computing noisy inputs and weights have proven useful for regularisation (a complexity control method), and randomness is of course a key ingredient of reinforcement learning.

After the overview of neural adaptivity, the paper has focused on its application to robot control. This basically entails the learning of nonlinear mappings relating stimuli to responses. Several robotics applications have been surveyed, classified according to the underlying mapping that needs to be approximated: inverse kinematics and visuomotor mappings.

The learning of inverse kinematics makes robot arms adaptive to changes in their own geometry (e.g., link bendings, encoder shiftings, etc.), while learning of visuomotor mappings allows robots to cope with changing environments (e.g., different loads, moving objects, etc.).

A first remark stemming from the survey of robotic applications is that in the case of mappings that can be easily sampled, it seems sufficient to apply a plain error-minimisation procedure. Some simple inverse kinematics mappings and visuomotor mappings used for visual positioning have been learned in this way. If the input space is complex, then many researchers have resorted to a combination of correlational rules for the efficient coding of that space, with error-minimisation rules to build the appropriate association with the outputs. The use of SOMs to encode the robot workspace or the sensor space falls into this category. Then, an error-minimisation rule is used to build the appropriate input-output mapping: inverse kinematics in this case. Finally, when the task is specified as a goal to be reached using sensory feedback, without making explicit the movements necessary to reach it, then the only possibility is to resort to reinforcement learning schemes, which depend just on the availability of a measure of success rather than an error measure.

The number of learning cycles required ranges widely in the applications described, depending on the complexity of the mapping to be learned as well as on the accuracy required. Note that learning time is directly related to the number of training samples, each of which entails at least one robot movement. And robots are slow as compared to computers. Therefore, minimising the number of training samples is of paramount importance in the application of neural networks to robotics, and many efforts are currently oriented in this direction (e.g., adaptive sampling, function decomposition) [37, 38].

Acknowledgments

This work has been partially supported by the Catalan Research Commission, through the “Robotics and Control” group. The author thanks Bernd Maediger, from Daimler-Benz Aerospace, and Christophe Venaille, from Thomson Broadcast Systems, for providing the specifications and the robot set-ups for the applications described in Sections 5 and 6 in the framework of the former Esprit project CONNY, and also for supplying the photographs in Figures 2 and 3, respectively.

References

1. Albus JS (1975) A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control* 97: 220-227
2. An G (1996) The effects of adding noise during backpropagation training on generalization performance. *Neural Computation* 8: 643-674
3. Arbib MA (1995) *Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA (An updated second edition will appear in November 2002)

4. Bäck T, Fogel DB and Michalewicz Z (eds) (1997) Handbook of Evolutionary Computation. Oxford University Press, New York, and Institute of Physics Publishing, Bristol
5. Beyer H-G and Schwefel H-P (2002) Evolution strategies - A comprehensive introduction. *Natural Computing* 1(1): 3-52
6. Bishop C (1995) Neural Networks for Pattern Recognition. Oxford University Press
7. Cherkassky V (2002) Model complexity control and statistical learning theory. *Natural Computing* 1(1): 109-133
8. D'Souza A, Vijayakumar S and Schaal S (2001) Learning inverse kinematics. Proc. IEEE/RSJ Conf. on Intel. Robots and Systems, Maui, Hawaii, USA, pp. 298-303
9. Dorigo M (ed) (1996) Special issue on 'Learning Autonomous Robots'. IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics 26(3)
10. Espiau B, Chaumette F and Rives P (1992) A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation* 8(3): 313-326
11. Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA
12. Grossberg S (1987) Competitive learning: from interactive activation to adaptive resonance. *Cognitive Science* 11: 23-63
13. Hebb DO (1949) The Organization of Behavior. Wiley, New York
14. Higuchi T, Iwata M and Liu W (eds) (1997) Evolvable Systems: From Biology to Hardware. Springer-Verlag, Berlin Heidelberg New York
15. Husbands P and Meyer JA (1998) Proc. 1st European Workshop on Evolutionary Robotics (EvoRobot'98). Springer-Verlag, Berlin Heidelberg New York
16. Kohonen T (2001) Self-Organizing Maps (third edition). Series in Information Sciences 30, Springer-Verlag, Berlin Heidelberg New York Tokyo
17. Koza JR (1992) Genetic Programming: On Programming Computers by means of Natural Selection. MIT Press, Cambridge, MA
18. Kröse BJA and van der Smagt PP (1993) An Introduction to Neural Networks (5th edition). University of Amsterdam
19. LeCun Y (1985) Une procedure d'apprentissage pour reseau au seuil assymetrique. Proc. of COGNITIVA, pp. 599-604
20. Millán JR and Torras C (1992) A reinforcement learning connectionist approach to robot path finding in non-maze-like environments. *Machine Learning* 8(3/4): 363-395
21. Millán JR and Torras C (1999) Learning sensor-based navigation. In: Morik K, Kaiser M and Klingspor V (eds) Making Robots Smarter: Combining Sensing and Action through Robot Learning. Kluwer Academic Publishers, Boston, MA
22. Mitchell T, Franklin J and Thrun S (1996) Recent Advances in Robot Learning. Kluwer Academic Publishers, Boston, MA
23. Morik K, Kaiser M and Klingspor V (1999) Making Robots Smarter: Combining Sensing and Action through Robot Learning. Kluwer Academic Publishers, Boston, MA
24. Omidvar O and van der Smagt P (1997) Neural Systems for Robotics. Academic Press, San Diego, CA
25. Park DC, El-Sharkawi MA and Marks II RJ (1991) An adaptively trained neural network. *IEEE Trans. on Neural Networks* 2(3): 334-345
26. Pavlov IP (1927) Conditioned Reflexes. Oxford University Press
27. Ritter H, Martinetz T and Schulten K (1992) Neural Computation and Self-Organizing Maps. Addison Wesley, New York
28. Ruiz de Angulo V and Torras C (1995) On-line learning with minimum degradation in feedforward networks. *IEEE Trans. on Neural Networks* 6(3): 657-668
29. Ruiz de Angulo V and Torras C (1997) Self-calibration of a space robot. *IEEE Trans. on Neural Networks* 8(4): 951-963
30. Ruiz de Angulo V and Torras C (2000) A framework to deal with interference in connectionist systems. *AI Communications* 13(4): 259-274
31. Ruiz de Angulo V and Torras C (2001) Architecture-independent approximation of functions. *Neural Computation* 13(5): 1119-1135

32. Ruiz de Angulo V and Torras C (2001) Neural learning invariant to network size changes. Proc. Intl. Conf. on Artificial Neural Networks (ICANN'01), Vienna, Austria, Lecture Notes in Computer Science 2130: 33-40
33. Ruiz de Angulo V and Torras C (2002) Learning inverse kinematics via cross-point function decomposition. Proc. Intl. Conf. on Artificial Neural Networks (ICANN-2002), Madrid, Spain, Lecture Notes in Computer Science 2415: 856-861
34. Ruiz de Angulo V and Torras C (2002) Sequential learning in feedforward networks: proactive and retroactive interference minimization. Proc. Intl. Conf. on Artificial Neural Networks (ICANN-2002), Madrid, Spain, Lecture Notes in Computer Science 2415: 1339-1344
35. Ruiz de Angulo V and Torras C (2002) A deterministic algorithm that emulates learning with random weights. Neurocomputing 48(1-4): 975-1002
36. Ruiz de Angulo V and Torras C (2004) Neural learning methods yielding functional invariance. Theoretical Computer Science 320(1): 111-121
37. Ruiz de Angulo V and Torras C (2005) Speeding up the learning of robot kinematics through function decomposition. IEEE Trans. on Neural Networks, Nov.
38. Ruiz de Angulo V and Torras C (2005) Using PSOMs to learn inverse kinematics through virtual decomposition of the robot. Proc. 8th Intl. Work-Conf. on Artificial Neural Networks (IWANN'05), Vilanova i la Geltrú, Spain, Lecture Notes in Computer Science 3512: 701-708
39. Rumelhart DE, Hinton GE and Williams RJ (1986) Learning representations by back-propagating errors. Letters to Nature 323: 533-535
40. Simon HA (1969) The Sciences of the Artificial. MIT Press, Cambridge, MA
41. Skinner BF (1938) The Behavior of Organisms: An Experimental Analysis. Appleton Century
42. Steels L (1995) The Biology and Technology of Intelligent Autonomous Agents. NATO ASI Series F, Springer-Verlag, Berlin Heidelberg New York
43. Sutton RS and Barto AG (1998) Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA
44. Torras C (1985) Temporal-Pattern Learning in Neural Models. Lecture Notes in Biomathematics 63, Springer-Verlag, Berlin Heidelberg New York
45. Torras C (1989) Sensorimotor integration in robots. In: Ewert P and Arbib MA (eds) Visuomotor Coordination: Experiments, Comparisons, Models and Robots, Plenum Press, pp. 673-689
46. Torras C (1993) Symbolic planning versus neural control in robots. In: Rudomín P, Arbib MA, Cervantes-Pérez F and Romo R (eds.) Neuroscience: From Neural Networks to Artificial Intelligence, Research Notes in Neural Computing 4: 509-523, Springer-Verlag: Berlin Heidelberg New-York
47. Torras C (ed) (2001) Special issue on 'Neural Networks at IJCAI'01'. Intl. Journal of Computational Intelligence and Applications 1(4)
48. Torras C (2002) Neural computing increases robot adaptivity. Natural Computing 1(4): 391-425
49. van de Velde W (ed) (1993) Special issue on 'Towards Learning Robots'. Robotics and Autonomous Systems 8(1-2)
50. Walter J and Ritter H (1996) Rapid learning with parametrized self-organizing maps. Neurocomputing 12: 131-153
51. Wells G and Torras C (2001) Assessing image features for vision-based robot positioning. Journal of Intelligent and Robotic Systems 30(1): 95-118
52. Wells G, Venaille Ch and Torras C (1996) Vision-based robot positioning using neural networks. Image and Vision Computing 14: 715-732
53. Ziemke T and Sharkey N (eds) (1998) Special issue on 'Biorobotics'. Connection Science 10(3-4)