

IGMN: An incremental connectionist approach for concept formation, reinforcement learning and robotics¹

Milton Roberto Heinen, Paulo Martins Engel

Universidade Federal do Rio Grande do Sul. Av. Bento Gonçalves, 9500, Caixa Postal 15064
91501-970, Porto Alegre, RS, Brasil. mrheinen@inf.ufrgs.br, engel@inf.ufrgs.br

Abstract: This paper demonstrates the use of a new connectionist approach, called IGMN (standing for Incremental Gaussian Mixture Network) in some state-of-the-art research problems such as incremental concept formation, reinforcement learning and robotic mapping. IGMN is inspired on recent theories about the brain, especially the Memory-Prediction Framework and the Constructivist Artificial Intelligence, which endows it with some special features that are not present in most neural network models such as MLP, RBF and GRNN. Moreover, IGMN is based on strong statistical principles (Gaussian mixture models) and asymptotically converges to the optimal regression surface as more training data arrive. Through several experiments using the proposed model it is also demonstrated that IGMN learns incrementally from data flows (each data can be immediately used and discarded), it is not sensible to initialization conditions, does not require fine-tuning its configuration parameters and has a good computational performance, thus allowing its use in real time control applications. Therefore, IGMN is a very useful machine learning tool for concept formation and robotic tasks.

Key words: Artificial neural networks, Bayesian methods, concept formation, incremental learning, Gaussian mixture models, autonomous robots, reinforcement learning.

Introduction

Traditional artificial neural network (ANN) models, such as Multi-layer Perceptron (MLP) (Rumelhart *et al.*, 1986), Radial Basis Functions (RBF) network (Powell, 1987) and General Regression Neural Network (GRNN) (Specht, 1991), are based on Cybernetics, a science devoted to understand the phenomena and natural processes through the study of communication and control in living organisms, machines and social processes (Ashby, 1956). Cybernetics had its origins and evolution in the second-half of the 20th century, especially after the development of the McCulloch-Pitts neural model (McCulloch and Pitts, 1943). According to Cybernetics, the brain can be seen as an information system that receives information as input, performs some processing over this information and outcomes the com-

puted results as output. Therefore, in traditional connectionist models the information flow is unidirectional, from the input to the hidden layer (processing) and then to the output layer (Pfeifer and Scheier, 1994).

Although neural networks can be successfully used in several tasks, including signal processing, pattern recognition and robotics, most ANN models have some disadvantages that difficult their use in on-line tasks such as incremental concept formation and robotics. The Backpropagation learning algorithm (Rumelhart *et al.*, 1986), for instance, requires several scans over all training data, which must be complete and available at the beginning of the learning process, to converge for a good solution. Moreover, after the end of the training process the synaptic weights are “frozen”, i.e., the network loses its learning capabilities. These drawbacks highly contrast with

¹ Article based on Doctoral Thesis of Milton Roberto Heinen called *A Connectionist Approach for Incremental Function Approximation and On-line Tasks* (Heinen, 2011).

the human brain learning capabilities because: (i) we do not need to perform thousands of scans over the training data to learn something (in general we are able to learn using few examples and/or repetitions); (ii) we are always learning new concepts as new “training data” arrive, i.e., we are always improving our performance through experience; and (iii) we do not have to wait until sufficient information arrives to make a decision, i.e., we can use partial information as it becomes available. Besides being not biologically plausible, these drawbacks difficult the use of ANNs in online robotics, because in this kind of application the training data are just instantaneously available to the learning system and a decision must be made using the information available at the moment.

In Heinen (2011) and Heinen and Engel (2010a, 2010b, 2010c) a new artificial neural network model, called IGMN² (standing for Incremental Gaussian Mixture Network), was proposed to tackle great part of these problems. IGMN is based on parametric probabilistic models (Gaussian mixture models), that have nice features from the representational point of view, describing noisy environments in a very parsimonious way, with parameters that are readily understandable (Engel, 2009). Moreover, IGMN is inspired on recent theories about the brain, especially the Memory-Prediction Framework (MPF) (Hawkins, 2005) and the constructivist artificial intelligence (Drescher, 1991), which endows it with some unique features such as: (i) IGMN learns incrementally using a single scan over the training data; (ii) the learning process can proceed perpetually as new training data arrive; (iii) it can handle the stability-plasticity dilemma and does not suffer from catastrophic interference; (iv) the neural network topology is defined automatically and incrementally; and (v) IGMN is not sensible to initialization conditions.

The main goal of this paper is to present the application of IGMN in some practical tasks such as concept formation, reinforcement learning and robotic mapping, thus demonstrating that IGMN is a powerful machine learning tool that can be applied to many state-of-the-art computational and engineering problems. The remaining of this paper is organized as follows. Section *IGMN* presents

the main aspects of IGMN and its neural architecture. Section *Concept Formation* discusses the use of the proposed neural network model for incremental concept formation (Engel and Heinen, 2010a, 2010b; Heinen and Engel, 2010d), which is an important task in machine learning and robotics. Section *Reinforcement Learning* describes how IGMN can be used as a function approximator in reinforcement learning (RL) tasks (Heinen and Engel, 2010b). Section *Feature-based mapping* presents a new feature-based mapping algorithm (Heinen and Engel, 2010d, 2010e) that represents the environment using the multivariate Gaussian mixture models rather than grid cells or line segments. Finally, Section *Conclusion* provides some final remarks and perspectives.

IGMN

Figure 1 shows the general architecture of IGMN, which is inspired in the Memory-Prediction Framework (MPF) (Hawkins, 2005). It is composed by an association region P (in the top of this figure) and many cortical regions, $N^A, N^B \dots N^S$. All regions have the same size, i.e., the number of neurons, M , is always the same for all regions. Initially there is a single neuron in each region (i.e., $M = 1$), but more neurons are incrementally added when necessary using an error driven mechanism. Each cortical region N^K receives signals from the k th sensory/motor modality, \mathbf{k} (in IGMN there is no difference between sensory and motor modalities), and hence there is a cortical region for each sensory/motor modality.

An important feature of IGMN is that all cortical regions execute a common function, i.e., they have the same kind of neurons and use the same learning algorithm. Moreover, all cortical regions can run in parallel, which improves the performance especially in parallel architectures. More specifically, each neuron j of region N^K performs the following operation:

$$p(\mathbf{k}|j) = \frac{1}{(2\pi)^{\frac{D^K}{2}} \sqrt{|C_j^K|}} e^{-\frac{1}{2}(\mathbf{k} - \mu_j^K)^T C_j^K^{-1} (\mathbf{k} - \mu_j^K)}, \quad (1)$$

i.e., it uses a multivariate Gaussian activation function, where D^K is the dimensionality of \mathbf{k} (different sensory/motor modalities \mathbf{k} can

² Initially IGMN was called IPNN, but the name was changed to avoid misunderstandings with the Specht's PNN model (Specht, 1990).

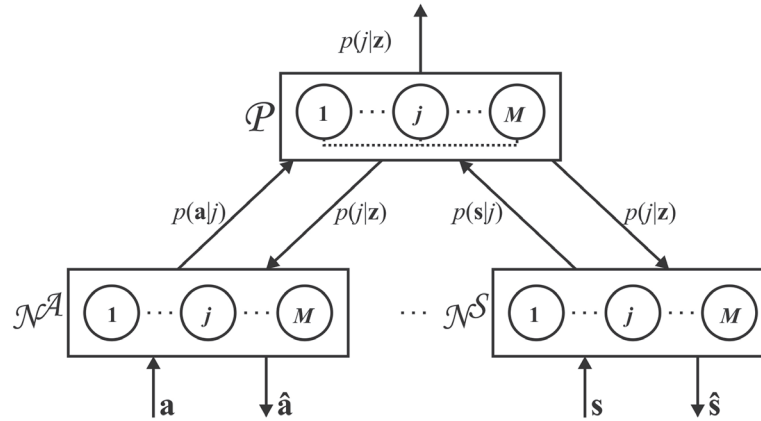


Figure 1. General architecture of IGMN.

have different dimensions D^K). Each neuron j of N^K maintains a mean vector μ_j^K and a covariance matrix C_j^K . These covariance matrices are initialized using a user defined fraction δ of the overall variance (e.g., $\delta = 1/100$) of the corresponding attributes, estimated from the range of these values according to:

$$\sigma_{ini}^K = \delta [\max(\mathbf{k}) - \min(\mathbf{k})], \quad (2)$$

where $[\min(\mathbf{k}), \max(\mathbf{k})]$ defines the domain of a sensory/motor modality \mathbf{k} . It is important to say that it is not necessary to know the exact minimum and maximum values along each dimension to compute σ_{ini}^K but instead just the approximate domain of each feature.

Another important aspect of IGMN is that the neural regions are not fully connected, i.e., the neuron j of N^K is connected just to the j th neuron of P , but this connection is bidirectional. It is also important to notice that there are no *synaptic weights* in these connections, i.e., all IGMN parameters are stored in the neurons themselves. A bottom-up connection between N^K and P provides the component density function $p(\mathbf{k}|j)$ to the j th neuron in P . Therefore, a neuron j in the association region P is connected with the j th neuron of all cortical regions N via bottom-up connections and computes the a posteriori probability using the Bayes' rule:

$$p(j|\mathbf{z}) = \frac{p(\mathbf{a}|j) p(\mathbf{b}|j) \dots p(\mathbf{s}|j) p(j)}{\sum_{q=1}^M p(\mathbf{a}|q) p(\mathbf{b}|q) \dots p(\mathbf{s}|q) p(q)} \quad (3)$$

where it is considered that the neural network has an arbitrary number, s , of cortical regions and $\mathbf{z} = \{\mathbf{a}, \mathbf{b}, \dots, \mathbf{s}\}$. The dotted lines in

Figure 1 indicate the lateral interaction among the association units for computing the denominator of the Bayes' rule. The dotted lines in Figure 1 above indicate the lateral interaction among the association units for computing the denominator in (3). Each neuron j of the association region P maintains its a priori probability, $p(j)$, an accumulator of the a posteriori probabilities, sp_j , and an association matrix to store the correlations among each sensory/motor modality. The top-down connections between P and N^K , on the other hand, returns *expectations* to N^K that are used to estimate \mathbf{k} when it is missing.

IGMN has two operation modes, called *learning* and *recalling*. But unlike most ANN models, in IGMN these operations do not need to occur separately, i.e., the learning and recalling modes can be intercalated. In fact, even after the presentation of a single training pattern the neural network can already be used in the recalling mode (the acquired knowledge can be immediately used), and the estimates become more precise as more training data are presented. Moreover, the learning process can proceed perpetually, i.e., the neural network parameters can always be updated as new training data arrive.

As described before, IGMN adopts an error-driven mechanism to decide if it is necessary to add a neuron in each region for *explaining* a new data vector \mathbf{z}^t . This error-driven mechanism is inspired on the Constructivist IA (Drescher, 1991; Chaput, 2004), where the accommodation process occurs when it is necessary to change the neural network structure (i.e. to add a neuron in each region) to account for a new *experience* which is not explained for the current schemata (i.e., the current ANN

structure), and the assimilation process occurs when the new experience is well explained in terms of the existing schemata (Piaget, 1954). In mathematical terms, the ANN structure is changed if the instantaneous approximation error ε is larger than a user specified threshold ε_{max} . More details about the IGMN learning and recalling operation modes can be found at (Heinen, 2011). Next sections demonstrate the use of IGMN in many practical problems such as concept formation, reinforcement learning and robotics.

Concept formation

One of our primary motivations in developing IGMN was to tackle problems like those encountered in autonomous robotics. To be more specific, let us consider the so called perceptual learning, which allows an embodied agent to understand the world (Burfoot *et al.*, 2008). Here an important task is the detection of concepts such as “corners”, “walls” and “corridors” from the sequence of noisy sensor readings (e.g., sonar data) of a mobile robot. The detection of these regularities in data flow allows the robot to localize its position and to detect changes in the environment (Thrun *et al.*, 2006).

Although concept formation has a long tradition in machine learning literature, in the field of unsupervised learning, most methods assume some restrictions in the probabilistic modelling (Gennari *et al.*, 1989) which prevent their use in online tasks. The well known *k-means* algorithm (MacQueen, 1967; Tan *et al.*, 2006), for instance, represents a concept as a mean of a subset or cluster of data. In this case, each data point must deterministically belong to one concept. The membership of a data point to a concept is decided by the minimum distance to the means of the concepts. To compute the means, all data points belonging to every concept are averaged using a fixed number of concepts along all the learning process. For learning probabilistic models, a very used approach is the batch-mode EM algorithm (Dempster *et al.*, 1977), which follows a mixture distribution approach for probabilistic modelling. Like *k-means*, this algorithm requires that the number of concepts be fixed and known at the start of the learning process. Moreover, the parameters of each distribution are computed through the usual statistical point estimators, a batch-mode approach which considers that the complete

training set is previously known and fixed (Tan *et al.*, 2006).

These restrictions make the *k-means* and EM algorithms not suitable for on-line concept formation, because in this kind of task usually each data point is just instantaneously available, i.e., the learning system needs to build a model, seen as a set of concepts of the environment, incrementally from data flows. The IGMN model, on the other hand, is able to learn from data flows in an incremental (new concepts can be added by demand) and online (it does not require that the complete training set be previously known and fixed) way, which makes it a good solution for concept formation in on-line robotic tasks. Moreover, unlike the traditional neural network models (e.g., MLP and GRNN), the IGMN hidden neurons are not “black boxes”, and thus the Gaussian units can be interpreted as representations of the input space, i.e., high level concepts (Engel and Heinen, 2010a). The remaining of this section is organized as follows: Subsection *Related work* presents some related work about concept formation, and Subsection *Concept formation experiments* describes how IGMN can be used to build high-level concepts incrementally from data flows.

Related work

In the past different approaches were presented to create high level concepts from sonar data in robotic tasks. As a typical example of these approaches, Nolfi and Tani (1999) proposed a hierarchical architecture to extract regularities from time series, in which higher layers are trained to predict the internal state of lower layers when such states significantly change. In this approach, the segmentation was cast as a traditional error minimization problem (Haykin, 2008), which favours the most frequent inputs, filtering out less frequent input patterns as being “noise”. The result is that the system recognizes slightly differing walls that represent frequent input patterns, as distinct concepts, but is unable to detect corridors or corners that are occasionally (infrequently) encountered. Moreover, this algorithm has scarce means to handle the stability-plasticity dilemma and to appropriately model the data.

Focusing in change detection, Linåker and Niklasson (2000a, 2000b) proposed an adaptive resource allocating vector quantization (ARAVQ) network, which stores moving averages of segments of the data sequence as

vectors allocated to output nodes of the network. New model vectors are incorporated to the model if a mismatch between the moving average of the input signal and the existing model vectors is greater than a specified threshold, and a minimum stability criterion for the input signal is fulfilled. The main advantage of this approach over the Nolfi and Tany's model is that the ARAVQ network requires a single scan over the training data to converge. Moreover, it can add hidden neurons (i.e., to create new concepts) incrementally from data flows. However, like other distance-based clustering algorithms its induced model is equivalent to a set of equiprobable spherical distributions sharing the same variance, what badly fits to a data flow with temporal correlation, better described by elongated elliptical distributions. Next subsection describes some experiments in which IGMN is used to learn high-level concepts in an incremental and efficient way.

Concept formation experiments

This subsection describes some experiments in which IGMN is used to create high-level concepts from data flows. In these experiments, the data consist of 10 continuous values provided by the Pioneer 3-DX simulator software ARCOS (Advanced Robot Control & Operations Software). A Pioneer 3-DX robot has 8 sonar sensors, disposed in front of the robot at regular intervals, and a two-wheel differential, reversible drive system with a rear caster for balance. Figure 2 shows a Pioneer 3-DX robot and the disposition of its sonar sensors.

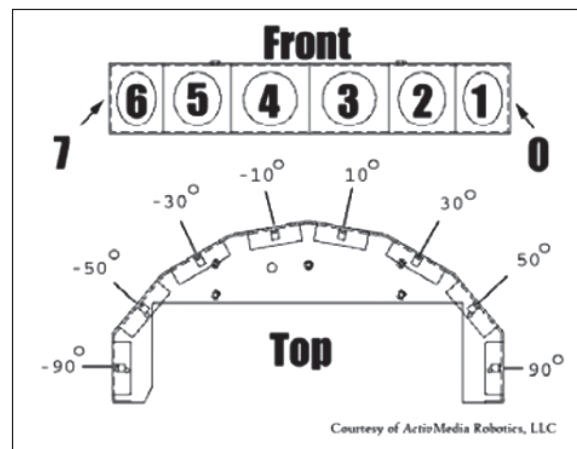
The IGMN network used in these experiments has two cortical regions, N^S and N^V . The cortical region N^S tackles the values of the sonar readings, i.e., $\mathbf{s} = \{s_1, s_2, \dots, s_8\}$, and the cortical region N^V receives the speeds applied at the robot wheels at time t , i.e., $\mathbf{v} = \{v_1, v_2\}$. To decide what is the most active



(a) Front view



(b) Side view



(c) Array of sonar sensors

Figure 2. Pioneer 3-DX mobile robot.

concept at time t , the maximum likelihood (ML) hypothesis $\ell = \arg \max_j [p(j|z)]$, where $z = \{s, v\}$, is used. It is important to note that IGMN computes and maintains the a posteriori probabilities of all concepts at each time, and hence it can be used in applications such as the so called multi-hypothesis tracking problem in robotic localization domains (Thrun *et al.*, 2006; Filliat and Meyer, 2003). The configuration parameters used in the following experiments are $\delta = 0.01$ and $\varepsilon_{max} = 0.1$. It is important to say that no exhaustive search was performed to optimize the configuration parameters.

The first experiment was accomplished in an environment composed of six corridors (four external and two internal), and the robot performed a complete cycle in the external corridors. Figure 3 shows the segmentation of the trajectory obtained by IGMN when the robot follows the corridors of this environment. IGMN created four probabilistic units, corresponding to the concepts "corridor" (1: plus sign), "wall at right" (2: circle), "corridor/obstacle front" (3: asterisk) and "curve at left" (4: cross). The symbols in the trajectory of Figure 3 represent the ML hypothesis in each robot position, and the black arrow represents the robot starting position and direction. More details about this experiment can be found at (Engel and Heinen, 2010a).

The next experiment was performed in a more complex environment, composed by two different sized rooms connected by a short corridor. This environment was inspired in those used by Linåker and Niklasson (2000a, 2000b). Figure 4 shows the

segmentation performed by IGMN in this experiment. IGMN has created seven clusters, corresponding to the concepts "wall at right" (1: plus sign), "corridor" (2: circle), "wall at right/obstacle front" (3: asterisk), "curve at left" (4: cross), "bifurcation/obstacle front" (5: square), "bifurcation/curve at right" (6: five-pointed star) and "wall at left/curve at right" (7: hexagram).

Comparing these experiments, it can be noticed that some similar concepts, like "curve at left" and "obstacle front", were discovered in both experiments, although these environments are different (the environment shown in Figure 3 has many corridors whilst that one shown in Figure 4 has two large rooms and just one short corridor). This points out that concepts extracted from a data flow corresponding to a specific sensed environment are not restricted to this environment, but they form an alphabet that can be reused in other contexts. This is a useful aspect that can improve the learning process in more complex environments.

Reinforcement learning

This section presents a couple of experiments, published in (Heinen and Engel, 2010b), in which IGMN is used as a function approximator in reinforcement learning (RL) algorithms. Traditional reinforcement learning techniques (e.g., Q-learning and Sarsa) (Sutton and Barto, 1998) generally assume that states and actions are discrete, which seldom occurs in real mobile robot applications. To allow continuous states and actions directly in RL (i.e., without discretization) it is necessary to use

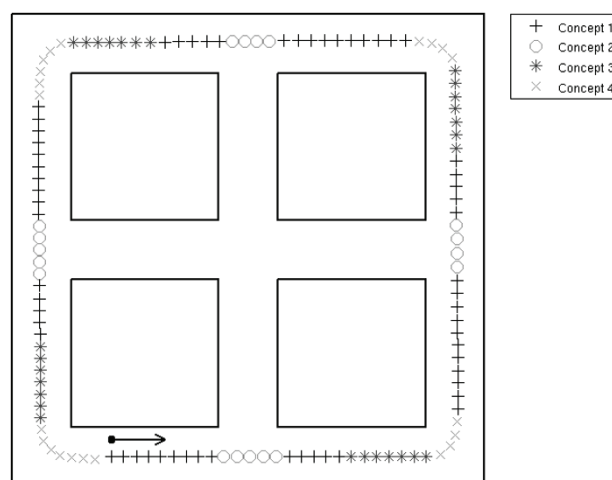


Figure 3. Segmentation of the trajectory.

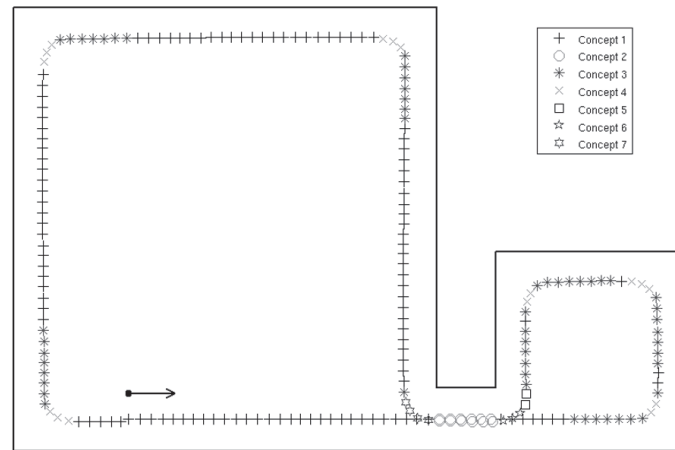


Figure 4. Segmentation of the trajectory.

function approximators like MLP (Utsunomiya and Shibata, 2009) or RBF (Doya, 2000; Basso and Engel, 2009) neural networks. According to Smart (2002), for a function approximator be successfully used in reinforcement learning tasks (i.e., for converging to a good solution) it must be: (i) incremental (it should not have to wait until a large batch of data points arrives to start the learning process); (ii) aggressive (it should be capable of producing reasonable predictions based on just a few training points); (iii) non-destructive (it should not be subject to destructive interference or “forgetting” past values); and (iv) must provide confidence estimates of its own predictions. Thus, according to these principles IGMN is very suitable for reinforcement tasks, i.e., it satisfies all the requirements described above. The rest of this section is organized as follows. Subsection *Related work* presents some related work in the field of reinforcement learning using continuous states and actions. Subsection *Selecting actions using IGMN* describes how IGMN can be used as a function approximator in a RL algorithm. Subsections *Pendulum with limited torque* and *Robot soccer task* describe some experiments performed to evaluate the proposed model in reinforcement learning tasks.

Related work

In the past, several approaches were proposed to allow continuous states and actions in RL algorithms. As an example of these approaches, in Doya (2000, 1996) a continuous formulation of the temporal difference $TD(\lambda)$ algorithm is presented (Sutton, 1988). This formulation uses normalized radial basis function (RBF)

networks to approximate the continuous state values and to learn the continuous actions. According to Doya (2000), RBF networks are more suitable for reinforcement learning tasks than MLP (Rumelhart *et al.*, 1986; Haykin, 2008) because they perform a local encoding of the input receptive fields, which avoids the catastrophic interference, i.e., the knowledge acquired in a region of the input space does not destroy the knowledge acquired previously in another region of the input space (Basso and Engel, 2009). However, in the algorithm described in Doya (2000, 1996) the radial basis functions are simply uniformly distributed among the input space and kept fixed during all the learning process, i.e., just the (linear) output layer is adjusted by the learning algorithm. Therefore, this algorithm does not adjust the network parameters of the hidden layers, which is a complex and nonlinear task. Moreover, it requires a priori knowledge to setup the neural units and wastes computational resources in unimportant regions of the input space.

Another interesting approach to allow continuous states and actions in RL is the Locally Weighted Regression (LWR) algorithm proposed by Smart and Kaelbling (2000). Although at a first glance LWR seems very promising, it has a strong drawback: it requires that all data points received so far to be stored and analyzed at each decision making (i.e., it is a “lazy learning” algorithm). Thus, this algorithm is not suitable for on-line robotic tasks, because in this kind of task the sensory data are very abundant, which makes the algorithm very slow and requires large amount of memory to store all previous data points. The IGMN learning algorithm, on the other hand, does

not require that any previous data to be stored or revisited, i.e., each training data can be immediately used and discarded. This makes the proposed model more suitable to be used in on-line robotic tasks, especially when the learning process must occur perpetually (i.e., when there are no separate phases for learning and use). The next subsection describes how IGMN can be used to select continuous actions in a RL algorithm.

Selecting actions using IGMN

Implementing a reinforcement learning algorithm using IGMN can be straightforward – we just need to use three cortical regions, N^S , N^A and N^Q , to represent the states, \mathbf{s} , actions, \mathbf{a} , and the $Q(\mathbf{s}, \mathbf{a})$ values, respectively. If the actions are discrete, then it is very easy to select the best action at each time: we just need to propagate the current state and all possible actions in the corresponding cortical regions and select the action which has the highest Q value, i.e.:

$$\mathbf{a}^* = \arg \max_{\mathbf{a}} [Q(\mathbf{s}, \mathbf{a})]. \quad (4)$$

Moreover, the exploration x exploitation dilemma can be tackled using an action selection mechanism such as *softmax* and ϵ -greedy (Sutton and Barto, 1998). On the other hand, if the actions are continuous, the action selection process becomes a general optimization problem far from trivial (Smart, 2002).

In this paper a new strategy, proposed in (Heinen, 2011), is used for selecting continuous actions in reinforcement learning algorithms. This strategy consists in first propagating through the IGMN network the current state, \mathbf{s} , and the maximum value, Q_{max} , currently stored in the corresponding Gaussian units, i.e.:

$$Q_{max} = \max_{j \in M} (\mu_j^Q). \quad (5)$$

Then the Q_{max} value is propagated through the cortical region N^Q , the associative region P is activated and the *greedy* action $\hat{\mathbf{a}}$ is computed in the cortical region N^A .

To tackle the exploration x exploitation dilemma, instead of simply choosing the greedy action $\hat{\mathbf{a}}$ at each moment we can randomly select the actions using the estimated covariance matrix $\hat{\mathbf{C}}^A$, i.e., the actions can be randomly selected using a Gaussian distribution of mean $\hat{\mathbf{a}}$ and covariance matrix $\hat{\mathbf{C}}^A$. In the beginning

of the learning process, when $M = 0$, the initial action can be randomly chosen. The main advantage of this action selection mechanism is that it enables high exploration rates in the beginning of the learning process, when the Gaussian distributions are larger, and this exploration is reduced as the confidence estimates become stronger. Moreover this mechanism does not require any optimization technique (just the IGMN itself), which makes the proposed RL algorithm very fast. Hence, this mechanism allows an exploration strategy based on statistical principles which do not require *ad-hoc* parameters.

The following subsections describe two experiments performed to evaluate the proposed model in reinforcement learning tasks using continuous states and actions: a pendulum with limited torque and a robot soccer task in a simulated environment. The configuration parameters used in these experiments are $\delta = 0.01$ and $\epsilon_{max} = 0.1$. More details about these experiments are found at (Heinen and Engel, 2010b).

Pendulum with limited torque

This experiment, also performed by Doya (1996, 2000; Heinen and Osório, 2006b; Sutton, 1988) to evaluate the Doya's continuous actor-critic, consists in learning the control policy of a pendulum with limited torque using reinforcement learning (Figure 5). The dynamics of the pendulum are given by (Doya, 2000):

$$\begin{aligned} \dot{\theta} &= \omega \\ ml^2 \dot{\omega} &= -\mu\omega + mgl \sin \theta + \mu, \end{aligned} \quad (6)$$

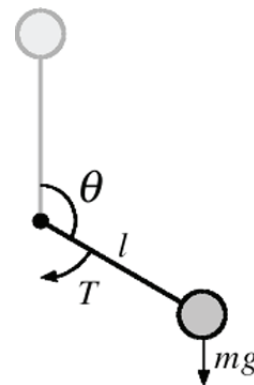


Figure 5. Pendulum swing up task.

where θ is the pendulum angle and θ' is the angular velocity. The physical parameters are mass $m = 1$, pendulum length $l = 1$, gravity constant $g = 9.81$, time step $\Delta t = 0.02$ and

maximum torque $T_{max} = 5.0$. The reward is given by the height of the tip of the pendulum, $R(x) = \cos(\theta)$, and the discount factor is $\gamma = 0.9$. Each episode starts from an initial state $x(0) = (\theta(0), 0)$, where $\theta(0)$ is selected randomly in $[-\pi, \pi]$. An episode lasted for 20 seconds unless the pendulum is over-rotated ($|\theta| > 5\pi$). These parameters are the same used in the continuous actor-critic (Doya, 2000). Due to the stochastic nature of RL, this experiment was repeated 50 times using different random seeds, and the average of the obtained results is shown in Figure 6a.

In Figure 6a the x axis represents the learning episode, and the y axis represents the time in which the pendulum stayed up (t_{up}), i.e., when $|\theta| < \pi/4$ (this is the same evaluation criteria used by Doya (2000)). The thick line in Figure 6a represents the mean and the thin lines represent the 95% confidence interval of the obtained results. Comparing these results with those presented in Doya (2000), reproduced here in Figure 6b, we can notice that the proposed model has a superior performance compared to the Doya's continuous actor-critic (specially in the first episodes), is more stable and does not require any previous configuration of the Gaussian units. The average number of units added during learning is IGMN was 109.41.

Robot soccer task

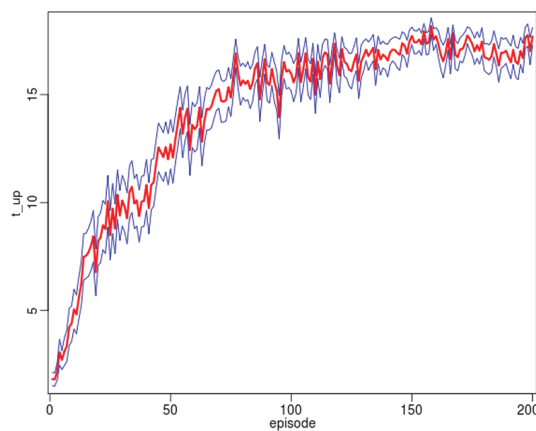
The next experiment, originally proposed in Asada *et al.* (1996, 2003), consists in learning to shoot a ball into the goal of a simulated robot soccer environment. To perform this experiment a robot soccer simulator was developed using

the Open Dynamics Engine (ODE - <http://www.ode.org>) physics simulation library. A previous version of this simulator, described in Heinen and Osório (2006a, 2006b, 2007), was used to evolve gaits of legged robots. The simulated environment follows the rules of the RoboCup (http://www.robocup.org/) Soccer Middle Size League. The soccer field has 18 meters of length by 12 meters of width, the goal has 1 meter of height and 2 meters of width, the goal posts have 12.5cm of diameter, and the ball has 70cm of circumference and 450 grams of weight. Moreover, walls of 1 meter of height were installed 1 meter apart from the field limits allowing the robot to perceive the environment using sonar sensors.

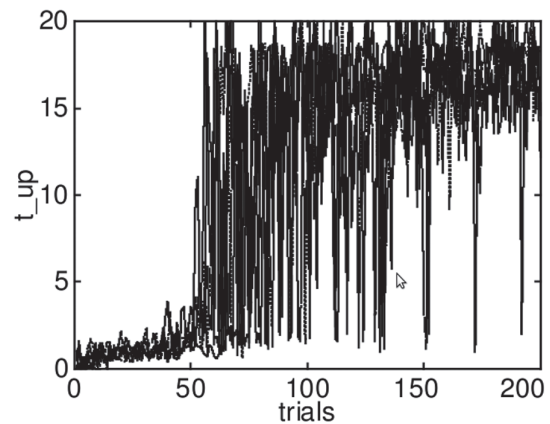
The simulated robot is similar to the Pioneer 3-DX robot used in the previous experiments. It has a box shape of 44.5cm of length, 39.3cm of width and 23.7cm of height. Its weight is 9kg and it has two wheels with 19.53cm of diameter and differential kinematics. The time interval Δt used in the simulations is 0.05 seconds. The IGMN network used in this experiment has two cortical regions, N^s and N^v . The cortical region N^s receives the values of the sonar readings, i.e., $\mathbf{s} = \{s_1, s_2, \dots, s_8\}$, and the cortical region N^v receives the speeds applied at the robot wheels at time t , i.e., $\mathbf{v} = \{v_1, v_2\}$. The reward function $r(t)$ used in this experiment is:

$$\begin{aligned} r(t) &= a(-d_{rb}(t)) + b(-d_{bg}(t)) & \text{if } d_{bg}(t) > 0 \\ r(t) &= 10 & \text{if } d_{bg}(t) > 0 \\ r(t) &= -10 & \text{if the ball exits the field} \end{aligned} \quad (7)$$

where $d_{rb}(t)$ is the distance from the robot to the ball, and $d_{bg}(t)$ is the distance from the ball



(a) Proposed RL algorithm



(b) Continuous actor-critic

Figure 6. Results in the pendulum task.

to the goal in the time t . The parameters $a = 1/4L$ and $b = 2/L$ (where L is the field length) are used to modulate the influence of the terms in the reward function. If the ball hits the goal the episode ends with a reward $r(t) = 10$ for a second, and if the ball exits the field the episode ends with a reward $r(t) = -10$ for a second. Moreover, if the simulation time exceeds $t_{max} = 300$ seconds the episode ends with no reward.

The learning process occurs in 1000 episodes. The robot starts an episode always in the same position, but the ball is randomly positioned (but in the range view of the sonar sensors). Thus, to obtain success in this task the robot needs: (i) to identify the ball using just sensory information; (ii) to move in the direction of the ball; and (iii) to “shoot” (or to lead) the ball into the goal without losing it. To evaluate the results two estimators were used: (i) the distance of the ball to the goal at the end of the episode (zero when the ball hits the goal) and (ii) the time required to the ball hit the goal (t_{max} in case of failing). Due to the stochastic nature of the task the whole experiment was repeated 30 times using different random numbers. Figure 7 shows the mean of the results obtained in this experiment.

Observing the graph of Figure 7a it can be noticed that in the first episodes the robot was not able to reach the ball (the variations are due to the random initial ball position), but after the 100th episode the distances are strongly reduced. After the 750th episode the mean distances had stabilized near 0.6 meters, which indicates that the robot was able to lead the ball into the goal in great part of the episodes.

The graph of Figure 7b, on the other hand, shows that the simulation time was practically constant (near t_{max}) until the 80th episode, where it starts to reduce strongly until

the 200th episode. Beyond this point the time reduces more slowly and stabilizes near 65 seconds after the 600th episode. These results show that the robot was able to accomplish the task at the end of the training process, because 60 seconds is the minimum time required to perform this task (i.e., to shoot a ball into the goal) using the simulation conditions described above.

Figure 8 shows an example of robot trajectory during the task. The number of probabilistic neurons added by IGMN during the learning process was 138.32 in average, and the time required to execute each experiment (i.e., to perform 1000 episodes) was approximately 2.5 hours.

Feature-based mapping

Map building is a fundamental problem in mobile robotics, in which a robot must memorize the perceived objects and features, merging corresponding objects in consecutive scans of the local environment (Thrun, 2002). There are several approaches to solve the map building problem. Among those are occupancy grid and feature-based maps. The occupancy grid maps are generated from stochastic estimates of the occupancy state of an object in a given grid cell (Thrun *et al.*, 2006). They are relatively easy to construct and maintain, but in large environments the discretization errors, storage space and time requirements become matters of concern. Feature-based maps, on the other hand, model the environment by a set of geometric primitives such as lines, points and arcs (Meyer and Filliat, 2003). Segment-based maps, which are the most common type of feature-based maps, have been advocated as

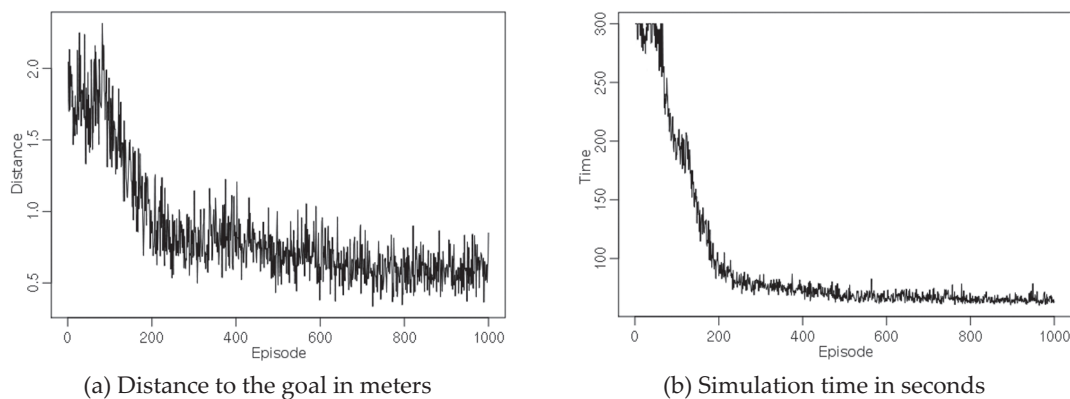


Figure 7. Robot soccer experiment.

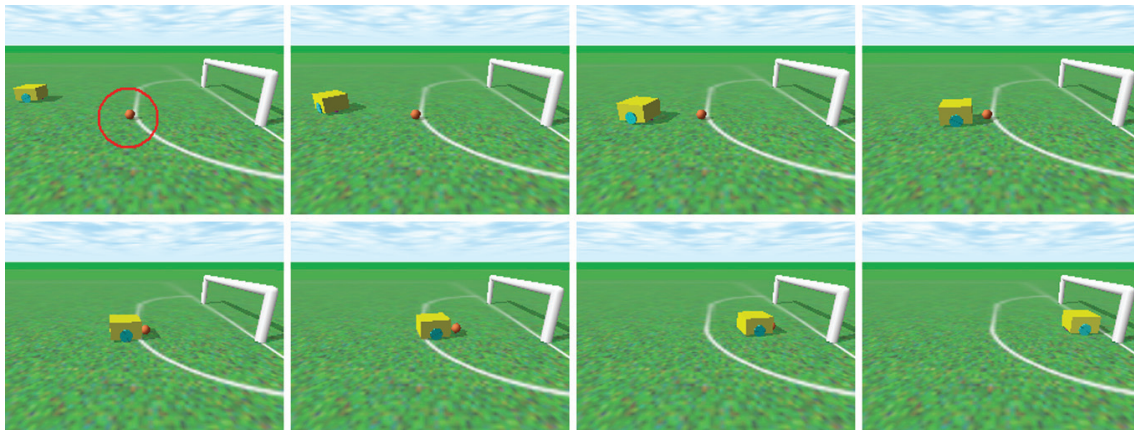


Figure 8. Robot trajectory during the task.

a way to reduce the dimensions of the data structures storing the representation of the environment (Amigoni *et al.*, 2006a). Its main advantage over occupancy grid maps is that line segments can be represented with few variables, thus requiring less storage space. Moreover, line segments are also easy to extract automatically from range data.

However, segment-based maps are not able to give closed and connected regions like occupancy grid maps because some objects do not provide line segments. Moreover, the number of extracted line segments is very high if the environment is irregular (not composed only by straight walls) and/or the range data is quite noisy. Another disadvantage of segment-based maps is the absence of probabilistic information in the generated map (although some researchers (Gasós and Martín, 1997; Ip *et al.*, 2002) have used fuzzy sets to deal with uncertainty in the mapping process). In fact, according to Thrun (2006), probabilistic approaches are typically more robust in face of sensor limitations, sensor noise, environment dynamics, and so on. Other localization and mapping techniques, such as particle filters and potential fields, generally use grid maps to represent the environment, and therefore have the same restrictions pointed out above.

This section presents a new feature-based mapping algorithm, proposed in (Heinen and Engel, 2011; Heinen and Engel, 2010e), which uses the IGMN probabilistic units to represent the features perceived in the environment. This kind of representation, which is inherently probabilistic, is more effective than segment-based maps because it has an arbitrary accuracy (it does not require discretization) and can even model objects that do not provide line

segments. Moreover, the proposed mapping algorithm does not require an exclusive kind of sensor (it can be used either with laser scanners or sonar sensors), requires low storage space and is very fast, which allows it to be used in real time. The remaining of this section is organized as follows. Subsection *Related work* describes some previous feature-based mapping techniques. Subsection *Mapping using IGMN* describes how IGMN can be used to create feature-based maps. Finally, Subsection *Experiments* describes some experiments performed to evaluate the proposed mapping algorithm using real sonar and simulated laser range data.

Related work

In the last decade several feature-based mapping algorithms have been proposed to solve the map building problem. In Zhang and Ghosh (2000) a segment-based mapping algorithm is proposed that describes a line segment using the center of gravity of its points and the direction θ of its supporting line. This algorithm groups laser points in clusters, and for each cluster, a line segment is generated. In Lee *et al.* (2005) a feature based mapping algorithm is presented which uses an association model to extract lines, points and arc features from sparse sonar data. In Puente *et al.* (2009) a mapping algorithm is presented which uses a segmentation algorithm derived from computer vision techniques to extract geometrical features from laser range data. In Latecki *et al.* (2004) a mapping algorithm is proposed which represents the environment by polygonal curves (polylines).

In (Amigoni *et al.*, 2006a, 2006b) a method derived from the Lu and Milos' algorithm

(Lu and Milos, 1998) is presented for building segment-based maps that contain a small number of line segments. In Lu and Milos (1998) an algorithm is proposed to build a global geometric map by integrating scans collected by laser range scanners. This method, which considers scans as collections of line segments, works without any knowledge about the robot pose. In Luo *et al.* (2008) an indoor localization method based on segment-based maps is proposed. It works in four steps: clustering scan data; feature extraction from laser data; line-based matching; and pose prediction. But this method assumes that the environment map already exists, i.e., it neither creates nor updates the map.

In Lorenzo *et al.* (2004) a method is proposed to solve the SLAM (Simultaneous Localization and Map Building) problem based on segments extracted from local occupancy maps, in which line segments are categorized as new obstacle boundaries of a simultaneously built global segment-based map or as prolongations of previously extracted boundaries. In Delius and Bugard (2010) a point-based representation is described, in which the data points gathered by the robot are directly used as a non-parametric representation. To reduce the large memory requirements necessary to store all data points, an off-line algorithm based on the fuzzy *k*-means is used to select the maximum-likelihood subsets of data points.

As described above, the main limitation of all these feature-based mapping techniques is the absence of probabilistic information in the generated map. To avoid this limitation, Gasós

and Martín (1997), Gasós and Rosseti (1999) use fuzzy-segments to represent uncertainty in the feature positions. This model extracts segments from points provided by sonar sensors, which are modelled on the map using fuzzy-sets. In the model proposed by Ip *et al.* (2002), on the other hand, an adaptive fuzzy clustering algorithm is used to extract and classify line segments in order to build a complete map for an unknown environment.

Although these mapping techniques are able to create good representations in simple environments composed by straight walls, most of them are not able to build the map in real-time while the robot navigates in the environment (i.e., they are off-line solutions). The mapping algorithm proposed in this subsection, on the other hand, is able to build environment representations in real time and incrementally. Moreover, it is inherently probabilistic and does not assume a specific environment structure. The next subsection describes how IGMN can be used in a feature-based mapping algorithm.

Mapping using IGMN

This subsection describes a geometric-based mapping algorithm which uses the IGMN units (also called mixture model components) to represent the features (objects, walls, etc) of the environment. Figure 9 shows the general architecture of the mapping algorithm. It consists mainly of two IGMN's, the local model, which represents the local perception of the robot, and the global model,

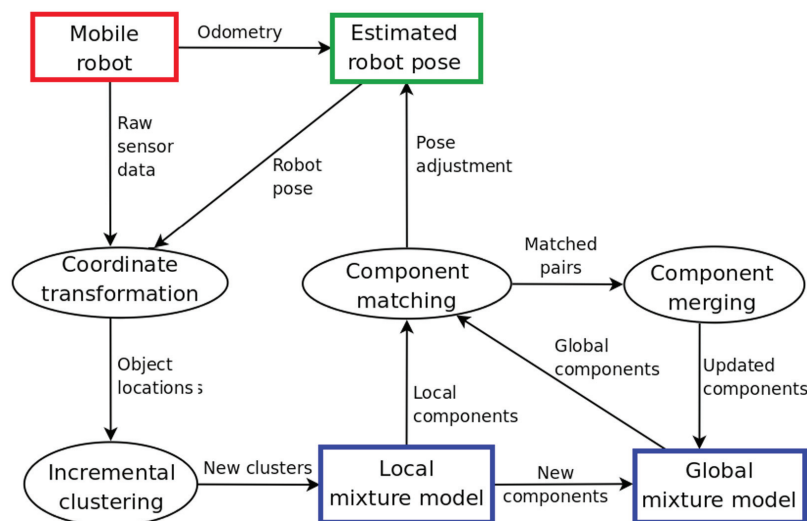


Figure 9. General mapping architecture.

which represents the global environment map. Each IGMN has two cortical regions, N^A and N^B . The cortical region N^A tackles the values of the sensor readings and the cortical region N^B receives the odometric information provided by the Pioneer simulator.

Initially both models are empty. When a sensor reading arrives (which can be a laser scan or a sonar reading) it is transformed into object locations on a global coordinate system based on the robot position estimated by the dead reckoning system. These object locations are grouped in clusters using the IGMN algorithm, thus composing the local model. When a specified number N of sensor readings arrives (e.g., 10 laser scans or 100 sonar readings), the local model is matched against the global model. If the global model is still empty, all local units are added to it and deleted from the local model. Otherwise the robot pose is adjusted to minimize the differences between the local and global models using a component matching process described in Heinen and Engel (2011). Both IGMN models are then merged into the global model and the local model is emptied. When a new sensor reading arrives, all these steps are repeated, and so the global model is updated at each N readings.

All this process occurs in real time at normal sensor arriving speeds (e.g., one laser

scan at each 100 milliseconds) even with more than 500 Gaussian units in the IGMN models. In fact, the prototype was able to perform all these operations (including the matching and merging processes) in less than 30 milliseconds on the same typical computer described before. In relation to memory requirements, the proposed mapping algorithm is very parsimonious, requiring just eight floating point numbers to store each two-dimensional Gaussian distribution (D^2+D+2 floating-point variables, where $D = 2$ is the dimension of the map). More details about the proposed feature-based algorithm can be found at Heinen and Engel (2011, 2010e).

Experiments

This subsection describes some experiments performed to evaluate the proposed mapping algorithm using two kinds of sensory information: (i) data provided by a simulated laser scanner; and (ii) data provided by sonar sensors. The robot used in these experiments is a Pioneer 3-DX, shown in Figure 2. This robot has a Sick LMS-200 laser scanner installed on it, which in ideal conditions is capable of measuring out to 80m over a 180° arc. Figure 10 shows the real environment used in the simulation. It is composed by two long corridors of

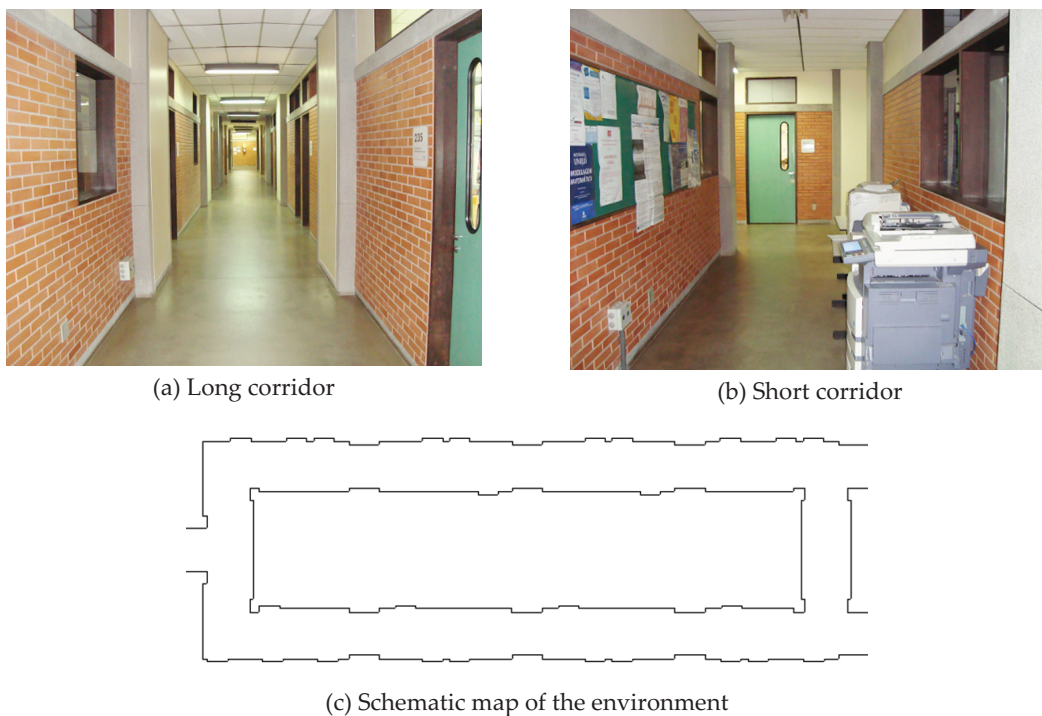


Figure 10. Simulated environment.

2.3x30 meters linked by two short corridors of 2.3x10 meters, as shown in the schematic map presented in Figure 10c. This environment has several irregularities (e.g. doors, saliencies and printers) which difficult the mapping process.

Experiments using laser data

This subsection describes two experiments performed using sensor data provided by a simulated laser scanner that is equivalent to the Sick LMS-200 installed on the real Pioneer 3-DX robot. In these experiments, the robot was manually controlled to perform one loop in the simulated environment shown in Figure 10c. A complete laser scan is received at each 100 milliseconds, and the mapping process is performed at each second (i.e., 10 scans). The first experiment was conducted using $\delta = 0.01$ and $\varepsilon_{max} = 0.1$, and this configuration produced 9 large clusters, as can be seen in Figure 11.

In this figure, each cluster is represented by an ellipse whose width is equivalent to a Mahalanobis distance of two. The occupancy

probabilities of this map are graphically shown in Figure 12, where darker regions represent higher occupancy probabilities and lighter regions correspond to probabilities close to 0. It is important to highlight that the proposed mapping algorithm does not have any random initialization and/or decision, and thus the obtained results are always identical for the same dataset and configuration parameters.

The next experiment was performed using the same conditions described above, but using $\varepsilon_{max} = 0.01$, which makes the system more sensible to small variations in the laser data. The results obtained in this experiment are shown in Figures 13 and 14. It can be noticed from Figure 13 that much more clusters were generated in this experiment (76 Gaussian components were generated). Nevertheless, these clusters fit very well the environment features, existing almost one cluster for each feature (doors entrances, saliencies in the walls, etc.). Moreover, each wall is modeled by a thin, long cluster which closely represents the center of the wall.

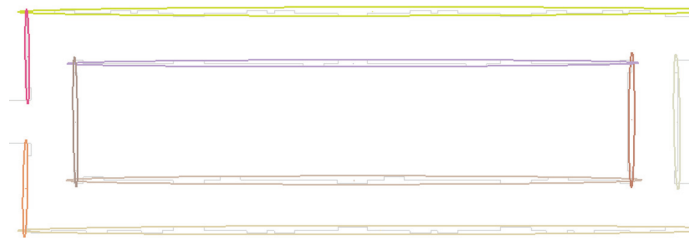


Figure 11. Results obtained using laser data.

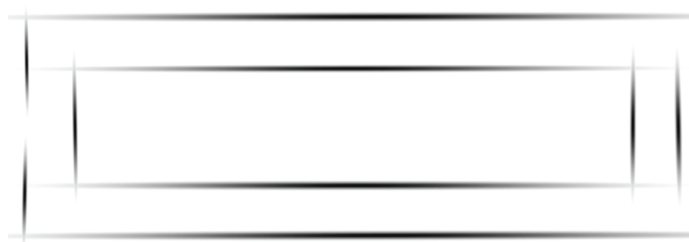


Figure 12. Occupancy probabilities.

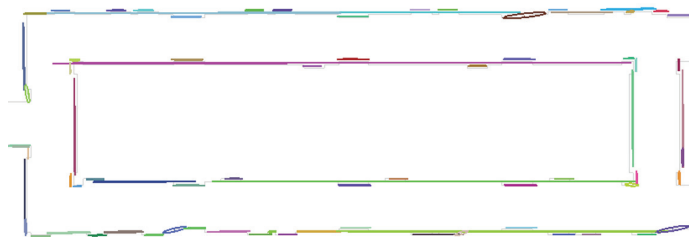


Figure 13. Results obtained using laser data.



Figure 14. Occupancy probabilities.

These experiments show that the proposed mapping algorithm is able to create useful representations of the environment, and these representations can be coarse (Figure 11) or fine (Figure 13) depending on the ε_{max} configuration value. Although these experiments were performed using high quality simulated laser range data, the proposed mapping algorithm is not restricted to this kind of sensory data. In the next subsection we describe an experiment performed using this same environment, but with data provided by real sonars, which are less accurate and noisier than laser scanners.

Experiments using real sonar data

In these experiments, the proposed mapping algorithm is evaluated using data provided by the sonar sensor array of the real Pioneer 3-DX robot. The time interval of each scan is 100 milliseconds, and the local model is generated using 100 complete scans (i.e., at each

10 seconds). Figure 15 shows the global model generated after one loop in the environment and Figure 16 illustrates the occupancy probabilities of this map, where darker regions represent higher probabilities. The configuration parameters used in this experiment are the same of the previous one (i.e., $\varepsilon_{max} = 0.01$).

We can notice that even with many noise sources present in the environment, the proposed mapping algorithm was able to create a reasonable map of the environment using this very noisy sensory data. The results obtained in Figure 15 show that the proposed mapping algorithm does not require just laser scanner sensors, but of course the quality of the maps is superior when high quality sensor data is used.

Conclusion

This paper has shown the use of IGMN, proposed at Heinen (2011) and Heinen and Engel

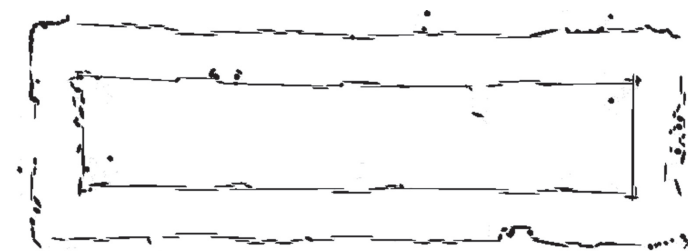


Figure 15. Results obtained using sonars.

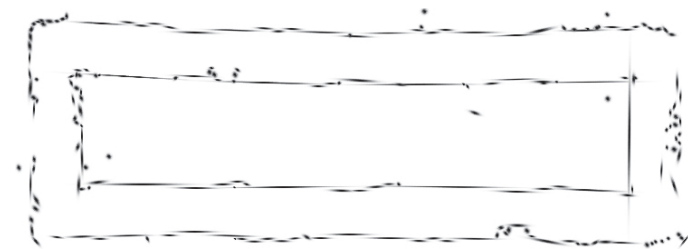


Figure 16. Occupancy probabilities.

(2010c) in many applications such as reinforcement learning, robotic mapping and concept formation, thus demonstrating that IGMN is a very powerful machine learning tool that can be applied in many state-of-the-art problems of computer science, robotic and control research areas. The future perspectives include expanding the feature-based mapping algorithm, presented in Section 5, into a complete SLAM solution, thus controlling the action of the mobile robot as it navigates through the environment.

References

- AMIGONI, F.; FONTANA, G.; GARIGIOLA, F. 2006a. A Method for Building Small-Size Segment-Based Maps In: M. GINIS; R. VOYLES, *Distributed Autonomous Robotic Systems*, Berlin, Springer, vol. 7, p. 11-20.
http://dx.doi.org/10.1007/4-431-35881-1_2
- AMIGONI, F.; GASPARINI, S.; GINI, M. 2006b. Building segment-based maps without pose information. *Proceedings of the IEEE*, **94**(7):1340-1359.
<http://dx.doi.org/10.1109/JPROC.2006.876925>
- ASADA, M.; KATOH, Y.; OGINO, M.; HOSADA, K. 2003. A humanoid approaches to the goal – reinforcement learning based on rhythmic walking parameters. In: D. POLANI; B. BROWNING; A. BONARINI; K. YOSHIDA, *RoboCup 2003: Robot Soccer World Cup VII*. Heidelberg, Springer, p. 344-354.
http://dx.doi.org/10.1007/978-3-540-25940-4_30
- ASADA, M.; NODA, S.; TAWARATSUMIDA, S.; HOSODA, K. 1996. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, **23**:279-303.
<http://dx.doi.org/10.1007/BF00117447>
- ASHBY, W.R. 1956. *An Introduction to Cybernetics*. London, Chapman & Hall, 295p.
- BASSO, E.W.; ENGEL, P. M. 2009. Reinforcement learning in non-stationary continuous time and space scenarios. In: ARTIFICIAL INTELLIGENCE NATIONAL MEETING (ENIA), 7, Bento Gonçalves, 2009. *Proceedings...* Bento Gonçalves, p. 1-8.
- BURFOOT, D.; LUNGARELLA, M.; KUNIYOSHI, Y. 2008. Toward a theory of embodied statistical learning. In: M. ASADA; J.C.T. HALLAM; J.A. MEYER; J. TANI, *From Animals to Animats 10*. Berlin, Springer, p. 270-279.
http://dx.doi.org/10.1007/978-3-540-69134-1_27
- CHAPUT, H. 2004. *The Constructivist Learning Architecture: A Model of Cognitive Development for Robust Autonomous Robots*. Austin, TX, Ph.D. Thesis. University of Texas, 91 p.
- DELIUS, D.M.; BUGARD, W. 2010. Maximum likelihood sample-based maps for mobile robots. *Robotics and Autonomous Systems*, **58**:133-139.
<http://dx.doi.org/10.1016/j.robot.2009.09.008>
- DEMPSTER, A.P.; LAIRD, N.M.; RUBIN, D.B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**(1):1-38.
<http://dx.doi.org/10.2307/2984875>
- DOYA, K. 1996. Temporal difference learning in continuous time and space. *Advances in Neural Information Processing Systems*, **8**:1073-1079.
- DOYA, K. 2000. Reinforcement learning in continuous time and space. *Neural Computation*, **12**(1):219-245.
<http://dx.doi.org/10.1162/089976600300015961>
- DRESCHER, G.L. 1991. *Made-up Minds: A Constructivist Approach to Artificial Intelligence*. Cambridge, The MIT Press, 220 p.
- ENGEL, P.M. 2009. *INBC: An incremental algorithm for dataflow segmentation based on a probabilistic approach*. Technical Report RP-360. Porto Alegre, UFRGS, 30 p.
- ENGEL, P.M.; HEINEN, M.R. 2010a. Concept formation using incremental Gaussian mixture models. In: I. BLOCH; R.M. CESAR, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Berlin, Springer-Verlag, p. 128-135.
http://dx.doi.org/10.1007/978-3-642-16687-7_21
- ENGEL, P.M.; HEINEN, M.R. 2010b. Incremental learning of multivariate Gaussian mixture models. In: A.C. da R. COSTA; R.M. VICARI; F. TONIDANDEL, *Advances in Artificial Intelligence – SBIA 2010*. Berlin, Springer, p. 82-91.
http://dx.doi.org/10.1007/978-3-642-16138-4_9
- FILLIAT, D.; MEYER, J. 2003. Map-based navigation in mobile robots: I. A review of localization strategies. *Cognitive Systems Research*, **4**(4):243-282.
[http://dx.doi.org/10.1016/S1389-0417\(03\)00008-1](http://dx.doi.org/10.1016/S1389-0417(03)00008-1)
- GASÓS, J.; MARTÍN, A. 1997. Mobile robot localization using fuzzy maps. In: T.P. MARTIN; A.L. RALESCU, *Fuzzy Logic in Artificial Intelligence Towards Intelligent Systems*. Berlin, Springer-Verlag, p. 207-224.
http://dx.doi.org/10.1007/3-540-62474-0_16
- GASÓS, J.; ROSSETI, A. 1999. Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments. *Fuzzy Sets and Systems*, **107**(1):1-24.
[http://dx.doi.org/10.1016/S0165-0114\(97\)00321-7](http://dx.doi.org/10.1016/S0165-0114(97)00321-7)
- GENNARI, J.; LANGLEY, P.; FISHER, D. 1989. Models of incremental concept formation. *Artificial Intelligence*, **40**(1-3):11-61.
[http://dx.doi.org/10.1016/0004-3702\(89\)90046-5](http://dx.doi.org/10.1016/0004-3702(89)90046-5)
- HAWKINS, J. 2005. *On Intelligence*. New York, Owl Books, 268 p.
- HAYKIN, S. 2008. *Neural Networks and Learning Machines*. 3rd ed., Upper Saddle River, Prentice-Hall, 906 p.
- HEINEN, M.R. 2011. *A Connectionist Approach for Incremental Function Approximation and On-line Tasks*. Porto Alegre, RS. Ph.D. Thesis. Universidade Federal do Rio Grande do Sul – UFRGS, 172 p.

- HEINEN, M.R.; ENGEL, P.M. 2010a. IPNN: An incremental probabilistic neural network for function approximation and regression tasks. *In: INTERNATIONAL JOINT CONFERENCE 2010, BRAZILIAN NEURAL NETWORKS SYMPOSIUM (SBRN)*, 11, São Bernardo do Campo, 2010. *Proceedings...* São Bernardo do Campo, p. 39-44.
- HEINEN, M. R.; ENGEL, P. M. 2010b. An incremental probabilistic neural network for regression and reinforcement learning tasks. *In: K. DIAMANTARAS; W. DUCH; L.S. ILIADIS, Artificial Neural Networks – ICANN 2010*. Berlin, Springer-Verlag, p. 170-179.
http://dx.doi.org/10.1007/978-3-642-15822-3_22
- HEINEN, M.R.; ENGEL, P.M. 2010c. IGMN: An incremental neural network model for on-line tasks. *In: WORKSHOP ON MSC DISSERTATION AND PHD THESIS IN ARTIFICIAL INTELLIGENCE (WTDIA)*, V, São Bernardo do Campo, 2010. *Proceedings...* São Bernardo do Campo, p. 732-741.
- HEINEN, M. R.; ENGEL, P. M. 2010d. Incremental probabilistic neural networks for concept formation, robotic localization and mapping. *In: DYNAMICS DAYS SOUTH AMERICA 2010 – INTERNATIONAL CONFERENCE OF CHAOS AND NONLINEAR DYNAMICS*, São José dos Campos, 2010. *Proceedings...* São José dos Campos, Institute of Physics Publishing, p. 75-76.
- HEINEN, M.R.; ENGEL, P.M. 2010e. Feature-based mapping using incremental gaussian mixture models. *In: LATIN AMERICAN ROBOTICS SYMPOSIUM AND VI INTELLIGENT ROBOTIC MEETING (LARS)*, VII, São Bernardo do Campo, 2010. *Proceedings...* São Bernardo do Campo, p. 67-72.
<http://dx.doi.org/10.1109/LARS.2010.13>
- HEINEN, M.R.; ENGEL, P. M. 2011. Incremental feature-based mapping from sonar data using gaussian mixture models. *In: ACM SYMPOSIUM ON APPLIED COMPUTING (SAC) – SPECIAL TRACK ON INTELLIGENT ROBOTIC SYSTEMS*, 26, TaiChung, 2011. *Proceedings...* New York, ACM Press, p. 1370-1375.
<http://dx.doi.org/10.1145/1982185.1982484>
- HEINEN, M.R.; OSÓRIO, F.S. 2006a. Applying genetic algorithms to control gait of physically based simulated robots. *In: IEEE CONGRESS OF EVOLUTIONARY COMPUTATION (CEC)*, Vancouver, 2006. *Proceedings...* Vancouver, p. 1823-1830.
<http://dx.doi.org/10.1109/CEC.2006.1688528>
- HEINEN, M.R.; OSÓRIO, F.S. 2006b. Gait control generation for physically based simulated robots using genetic algorithms. *In: J.S. SICHMAN; H. COELHO; S.O. REZENDE, Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*. Berlin, Springer-Verlag, p. 562-571.
http://dx.doi.org/10.1007/11874850_60
- HEINEN, M.R.; OSÓRIO, F.S. 2007. Evolving gait control of physically based simulated robots. *Journal of Theoretical and Applied Informatics (RITA)*, 14(1):119-134.
<http://dx.doi.org/10.1109/SBRN.2010.13>
- IP, Y.; RAD, A.; CHOW, K.; WONG, Y. 2002. Segment-based map building using enhanced adaptive fuzzy clustering algorithm for mobile robot applications. *Journal of Intelligent and Robotic Systems*, 35(3):221-245.
<http://dx.doi.org/10.1023/A:1021163807498>
- LATECKI, L.J.; LAKAEMPER, R.; SUN, X.; WOLTER, D. 2004. Building polygonal maps from laser range data. *In: INTERNATIONAL COGNITIVE ROBOTICS WORKSHOP (COGROB)*, 4, Valencia, 2004. *Proceedings...* Valencia, p. 1-7.
- LEE, S.J.; CHO, D.; CHUNG, W.K.; LIM, J.H.; KANG, C.U. 2005. Feature based map building using sparse sonar data. *In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS)*, Alberta, 2005. *Proceedings...* Alberta, p. 492-496.
<http://dx.doi.org/10.1109/IROS.2005.1545029>
- LINAKER, F.; NIKLASSON, L. 2000a. Time series segmentation using an adaptive resource allocating vector quantization network based on change detection. *In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN)*, Los Alamitos, 2000. *Proceedings...* Los Alamitos, p. 323-328.
<http://dx.doi.org/10.1109/IJCNN.2000.859416>
- LINAKER, F.; NIKLASSON, L. 2000b. Sensory flow segmentation using a resource allocating vector quantizer. *In: JOINT IAPR INTERNATIONAL WORKSHOPS ON ADVANCES IN PATTERN RECOGNITION*, London, 2000. *Proceedings...* Springer, p. 853-862.
- LORENZO, J.M.; MARTÍN, R.; NÚÑEZ, P.; PÉREZ, E.J.; HERNÁNDEZ, F.S. 2004. A hough-based method for concurrent mapping and localization in indoor environments. *In: CONFERENCE ON ROBOTICS, AUTOMATION AND MECHATRONICS (RAM)*, Singapore, 2004. *Proceedings...* Singapore, 2:840-845.
<http://dx.doi.org/10.1109/RAMECH.2004.1438027>
- LU, F.; MILOS, E. 1998. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249-275.
<http://dx.doi.org/10.1023/A:1007957421070>
- LUO, R.C.; LI, J.; CHEN, C. 2008. Indoor localization using line based map for autonomous mobile robot. *In: WORKSHOP ON ADVANCED ROBOTICS AND ITS SOCIAL IMPACTS*, Taipei, 2008. *Proceedings...* Taipei, p. 1-6.
<http://dx.doi.org/10.1109/ARSO.2008.4653586>
- MACQUEEN, J. 1967. Some methods for classification and analysis of multivariate observations. *In: BERKELEY SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY*, 5, Berkeley, 1967. *Proceedings...* Berkeley, 1:281-297.
- MCCULLOCH, W.S.; PITTS, W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133.
<http://dx.doi.org/10.1007/BF02478259>

- MEYER, J.; FILLIAT, D. 2003. Map-based navigation in mobile robots: II. A review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283-317.
[http://dx.doi.org/10.1016/S1389-0417\(03\)00007-X](http://dx.doi.org/10.1016/S1389-0417(03)00007-X)
- NOLFI, S.; TANI, J. 1999. Extracting regularities in space and time through a cascade of prediction networks: The case of a mobile robot navigating in a structured environment. *Connection Science*, 11(2):125-148.
<http://dx.doi.org/10.1080/095400999116313>
- PFEIFER, R.; SCHEIER, C. 1994. From perception to action: The right direction? In: FROM PERCEPTION TO ACTION CONFERENCE, Los Alamitos, 1994. *Proceedings...* Los Alamitos, p. 1-11.
<http://dx.doi.org/10.1109/FPA.1994.636076>
- PIAGET, J. 1954. *The construction of Reality in the Child*. New York, Basic Books, 386 p.
- POWELL, J.D. 1987. Radial basis function for multivariable interpolation: A review. In: IMA CONFERENCE ON ALGORITHMS FOR THE APPROXIMATION OF FUNCTIONS AND DATA, Shrivenham, UK, 1987. *Proceedings...* Shrivenham, p. 143-167.
- PUNTE, P.; VALERO, D.R.A.; MATIA, F. 2009. 3D feature based mapping towards mobile robots' enhanced performance in rescue missions. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), St. Louis, 2009. *Proceedings...* St Louis, p. 1138-1143.
<http://dx.doi.org/10.1109/IROS.2009.5354363>
- RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J. 1986. *Learning Internal Representations by Error Propagation*. Cambridge, The MIT Press.
- SMART, W.D. 2002. *Making Reinforcement Learning Work on Real Robots*. Providence, USA. Ph.D. Thesis. Brown University, 165 p.
- SMART, W.D.; KAEHLING, L.P. 2000. Practical reinforcement learning in continuous spaces. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 17, San Francisco, 2000. *Proceedings...* San Francisco, Morgan Kaufmann, p. 903-910.
- SPECHT, D.F. 1990. Probabilistic neural networks. *Neural Networks*, 3:109-118.
[http://dx.doi.org/10.1016/0893-6080\(90\)90049-Q](http://dx.doi.org/10.1016/0893-6080(90)90049-Q)
- SPECHT, D.F. 1991. A general regression neural network. *IEEE Trans. Neural Networks*, 2(6):568-576.
<http://dx.doi.org/10.1109/72.97934>
- SUTTON, R.S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9-44.
<http://dx.doi.org/10.1007/BF00115009>
- SUTTON, R.S.; BARTO, A.G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, The MIT Press, 322 p.
- TAN, P.; STEINBACH, M.; KUMAR, V. 2006. *Introduction to Data Mining*. Boston, Addison-Wesley, 900 p.
- THRUN, S. 2002. Robotic mapping: A survey. In: G. LAKEMEYER; B. NEBEL, *Exploring Artificial Intelligence in the New Millenium*. San Diego, Morgan Kaufmann Publishers.
- THRUN, S.; BURGARD, W. FOX, D. 2006. *Probabilistic Robotics. Intelligent Robotics and Autonomous Agents*. Cambridge, MIT Press, 652 p.
- UTSUNOMIYA, H.; SHIBATA, K. 2009. Learning with a recurrent neural network in a continuous state and action space task. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING OF THE ASIA-PACIFIC NEURAL NETWORK ASSEMBLY, New Zealand, 2009. *Proceedings...* New Zealand, p. 970-978.
- ZHANG, L.; GHOSH, B.K. 2000. Line segment based map building and localization using 2D laser rangefinder. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), San Francisco, 2000. *Proceedings...* San Francisco, 3:2538-2543.
<http://dx.doi.org/10.1109/ROBOT.2000.846410>

Submitted on April 19, 2011.

Accepted on May 19, 2011