April 1997
Technical Report #1063

General and Specific Models in Complex Robotics Systems – a critique and a proposal –

Uwe R. Zimmer

German National Research Institute
for Information Technology (GMD)
Schloß Birlinghoven – 53754 Sankt Augustin
Germany

This article focuses on the problem of balancing between general and specific modelling in complex system design. General models allow for more flexibility regarding the handling of unforeseen situations, while specific models can consider context conditions usually better. Based on the background of mobile robot applications, tasks like navigation, positioning, exploration, homing and others will be investigated regarding their preferences for an appropriate model. A frame called the LISA-model, considering an interaction of global and specific aspects, is proposed.

Due to the fact that the systems discussed here (adaptive, dynamic real-world interactions) are, as a whole, out of the range of analytical methods, real-world scenarios and empirical methods are set up and are discussed.

Keywords: **mobile robots, world modelling, dynamic indeterministic environments, neural networks, exploration, navigation, self-localization**

1. Introduction

In order to design a mobile robot system in a complex, dynamic environment forced to show some 'interesting' behaviour, some architectural concepts could be chosen out of the range of known methods, which can be classified either as *reflexive, local, and general* or *context-aware, global, and specific*. The attributes *reflexive* versus *context-aware* indicate the focus of information considered for decisions. *Locally* oriented modelling cannot distinguish easily between similar situations at different times or at different places. Thus such systems will model their environment and behaviours more *generally* than a *globally* oriented model, which can represent any *specific* situation in an individual instance. The remaining part of this section will increase this perspective and discusses some critical scenarios.

Reflexive, local, and general modelling

Examples for the first class of approaches can be found at multiple schools of robotic control architecture. The behaviour based, highly parallel approach presented e.g. by Maes in [12] is a classical representative of that class, but many have followed this idea. Thus at least references to behaviour based approaches or the subsumption architecture can be found in hundreds of mobile robot articles. Nehmzow emphasizes the aspect of plasticity on low levels (employing neural networks) in [15] producing general and highly adaptive models of the local environment. Nolfi discusses structures, which evolve local and general behaviour and furthermore adapt even the module structure of behaviours in [16]. A highly parallel behaviour oriented and genetic approach is suggested by Steels [22]. Employing the language of differential equations as a natural way to formulate behaviours together with their activation dynamics, Jaeger and Christaller propose a strict and simple implementation of behaviour based ideas in [10].

Common to all these concepts is that they are proven in real scenarios by physical mobile robots. The approaches mentioned above are representatives only, in order to stake out a whole research area. Finally there exists a lot of work on that field, which has (when seen historically) bridged the gap between abstract architectural concepts for complex behaviours and the physical world for the first time.

Essentially, these concepts produce general models of locally useful behaviours: decisions are based on local, immediate sensing, and the models reflect the general constraints of the actual working environment. In order to be useful in (partly) unknown environments the emphasis is on modelling general aspects than on modelling the known environment exactly. These approaches are problematic regarding the following scenarios:

- In case that local decisions cannot be based on local sensing only, these systems are determined to fail. These situations occur if either the sensor equipment is not able to detect the relevant static feature, or the time course of a feature is relevant. To give a simple example, the reader could imagine a mobile robot, which is well equipped with laser range finders and cameras, but is still not able to detect a common book board or a fractal structure like a plant reliably – not to talk about object recognition systems, even the simple detection as obstacles is not obvious. The laser range finder faces the problem that this structure looks like pink noise and the vision system has perhaps problems recognizing all the occurring textures. Although we are using sophisticated sensing equipment, a seemingly simple situation will be handled wrong. And, even worse, the experiment can be repeated over and over again, without any hope for improvement. This is not a point against cameras or laser range finders. It is easy to find critical scenarios for any other available (or known) sensor system as well.

One could argue that this robot is wrong configured regarding this environment. But the idea of adding new feature detectors is misleading in two ways. First we are far from being able to build a robot detecting all relevant features in any natural environment. Therefore it is not very likely to solve the problem by that means, but we can only hope to reduce its significance slightly. Second, even humans (or other biological species) are not able to detect all relevant features needed for local decisions. For example a high-voltage cable looks like a cable without voltage and a hot glass tube looks a cold glass tube. Moreover a lot of decisions are based on sophisticated knowledge about cultural and social constraints, which is not detectable by local sensing, but this problem class will not be addressed here. Nevertheless we are able to make up the right decisions in the case of the high voltage cable or the hot tube (hopefully) without adding new feature detection modules.

- The second class of scenarios is closely coupled with the first one, but is still different. Despite the first problem class, here, the relevant features can be detected, but not in a reliable way. Assuming an environment with many almost identical places or situations. In case that all these situations can be handled in the same manner, there is no problem. But assuming that one of these very similar places needs a specific treatment although the sensing looks very near to all the other ones, we are forced to model an exception. Unfortunately, modelling exceptions based on statistically weak significance does not fit the needs for reliability and robustness, which are essential here.

While the first class of scenarios highlights the problem of lacking context, the second one emphasizes the weakness of general models to react to exceptions. Both phenomena can be reduced to the need of context-aware and specific models, as introduced in the next section.

Context-aware, Global, and specific modelling

The second class of mobile robot approaches can be highlighted by a couple of different examples, too. For instance in [4], Elfes states a sophisticated model of sonar sensing and a grid-based mapping process, in order to construct a geometric, Cartesian representation of the space around the robot. This work has triggered dozens of others, which have tried to reduce algorithmic complexities by the inclusion of some heuristics, or to find special system parameters for specific environments. All these works are based on the idea of mapping an obstacle-probability distribution of sonar cones to a grid-map. A very well founded sensor and environment modelling can by found in [23], where Triggs describes a precise mathematical model of ultrasonic range measurement and its relation to localization tasks. The problems and chances of 'high precision'-approaches are discussed in some detail here. As far as known to the author, none of these 'high-precision' approaches has ever left the simulators or has been applied in an even slightly dynamic environment. A geometric mapping approach with an interesting low degree of sensor data interpretation is applied to a physical mobile robot equipped with high precision range finders by Edlinger and von Puttkamer in [3]. The resulting world model lacks the precision, which could be expected when looking at the employed equipment, but nevertheless (or even though) the exploration and navigation-behaviour of the whole system is stable. In [11], Kuipers and Byun introduce the idea of qualitative reasoning to this problem field and result in a qualitative, topological rather than in a precise, geometrical description of the world. In this model, instances are predefined situations, which have to be recognized in the actual environment, what makes the architecture rather hard to adapt to real world experiments. Kuipers's idea has inspired further work, even in real world scenarios, focusing on adaptability and lower levels of feature interpretation (e.g. [24]). In [5], Engels and Schöner employ dynamic neural fields for the representation of obstacles in the environment and (and that's the important new aspect) for the behaviours themselves as well. Thus behaviours and representations are modelled in one common structure (Amari's dynamic fields [1]) and can be correlated straightforward to each other. Nakamura et al. [14] utilize the sequences of sensor readings in order to build up a graph, where the nodes represent different typical behaviours (local movements), where the arcs state transition probabilities between these states.

A common feature of all these approaches is the construction or definition of a global world model, what does not mean a geometrical and Cartesian map necessarily, but just a structure, which the robot can correlate itself to, in order to get a meaning of a global time, place, or state (the following structures were applied in the examples: grid maps [4], geometric modelling [23], map of raw range data [3], qualitative symbols in a graph [11], qualitative, topologic maps [24], neural fields representing position and obstacles [5], and a behaviour oriented graph [14]). Therefore any experience in the actual working environment can be 'labelled' by a global or relative timestamp or position information. The structures are providing possibilities of concatenating sensor impressions at neighbouring places or in time-sequences, and the resulting robots have an understanding of time and/or place beyond reflexive behaviours. By employing this information, the problems from the first class of approaches can be overcome or reduced. On the other hand some other critical scenarios can be identified:

- Due to the fact that it is not possible in general to guarantee a consistent position information at any time during the robot's operation, the case where the correlation between the position and the current internal world model is getting weak or lost has to be considered carefully. Systems that do not provide a 'reflexive backup-system' for that case will fail here.
  This problem could be overcome by introducing a global observer (for example an observing camera, or even satellites) providing this information at any time. This is of course a legitimate solution for an actual, specific application, where this can be considered. On the other hand, one cannot expect to learn anything about the orientation and movement in space of all those creatures, which are not triggered and supported by an external observer (which is as far as known almost any creature on earth). For this scientific reason beyond actual applications the system is considered not having access to these possibilities.
- By concentrating on modelling specific places in space or specific moments in time-sequences, the aspect of generalization is getting critical. Considering a partly known and explored environment, the robot still needs a general answer to new situations. Generating general behaviours that can be applied to classes of situations which don't need to be neighboured in time or space, is not straightforward based on a global and specific modelling.

The need for a combination of local and general models is obvious, but the actual interaction or relation between them or their integration is not. The approach presented in this paper, is trying to overcome the gap between the general and the specific methodologies. It gives a common frame integrating these models in a harmonic way. The integration can also result in a selection of a specific approach for a class of tasks or situations, but in this case some reasons will be available for this decision.

2. *The considered system*

This section will give the frame for concepts and models presented in the sections to follow. The working environment, the mobile robot and the interactions between them (called behaviours in the following), are considered to be a closed system. Because of the complex interactions of the robot with its environment, the system has to be understood as a whole. Neither investigating the robot itself nor the behaviour isolated from the environment is seen as a promising possibility.

The design and analysis of such highly dynamic and complex systems which include 'real-world abilities', opens up a couple of traps and problems. For instance, the understanding of the scientist's integration in the life cycle of complex systems design and evaluation is getting more critical. Aspects of the awareness of implicit assumptions and especially the double role of the scientist as the designer and the evaluator of highly complex systems is discussed nicely in [18].

The environment

The working environment includes static and dynamic features. Static features are given by a couple of different objects, places, and light distributions, which are described in more detail in the experimental section 6. Important aspects of the static features concern their detectability:

- Most of the static features are detectable with at least one of the sensor systems employed by the mobile robot.
- Some of the features can be detected by a correlation of different sensor sources only.
- A limited amount of features cannot be detected by local sensing at all, but only by taking into account a global situation or a certain set of previous values.

All these features are assumed to be relevant for some requested behaviours (introduced later in this section). The environment is considered dynamical in different aspects:

- The majority of detectable features in the environment will change systematically by rates significantly slower than given by the internal timing of the moving robot, especially the sampling rate of the sensor system.
- A minority of detected features changes at speeds that are equal or higher than given by the internal timing of the robot.

All features are subject to disturbances, most of which are considered to be Gaussian, sporadic, or systematic in nature.

The mobile robot

The mobile platform has the ability to move in two dimensions (x, y), where the orientation of the platform α is relevant for some sensors and actuators, leading to a total of three degrees of freedom. The platform is assumed to be able to accelerate in a given direction up to a given speed without further planning. By using an adequate kinematic, i.e. a symmetric platform and synchronous drives, the problem of local manoeuvring is ignored. Thus the mobile robot can drive towards any direction immediately, without further planning steps, and without considering a traktrix[1], steering limitations and other restrictions.

With energy and computational power carried on board, the platform is assumed to be autonomous. Energy needs to be loaded regularly, which is moreover a part of the requested behaviours of the robot. Besides its energy supplement, the robot knows some internal states representing its current curiosity, the correlation to the current environment, and a 'feel-good' value. All of them are triggered by the environment and employed by the behaviours described in the next section.

The sensor system is set up in a way to fulfil the demands of detecting the majority of relevant static and dynamic features according to the previous section. Some important information like receiving pain or getting rewards is delivered by dedicated sensor systems.

The expected behaviours

In the given context, the mobile platform has to show multiple different and concurrent behaviours that are interpreted as biologically motivated behaviours like 'survive', 'find food', 'avoid danger', 'find the nest', etc. and are determined by evaluation functions judging the success of the several efforts. Specifically, the discussed behaviours are:

- *Avoid dangers* – avoiding static as well as dynamic situations, where physical damage could be expected or experienced. This includes the classical *collision avoidance* task.
- *Localization* – moving in a way that enables or increases the correlation between the specific world models and the current sensor situations. This behaviour should especially increase the consistency of the internal world models.

1. *a traktrix describes the curve left by e.g. a wheel on the floor during manueuvring of the mobile platform. Usually the traktrix of the outer form of the robot is considered for security reasons.*

- *Homing* – accessing an fixed place in space in a regular manner.
- *Exploration* – finding new places in the environment.
- *Energy collection* – finding and localizing energy sources in the current environment.
- *Feel-good* – finding places, which are described by an arbitrary optimization function.
- *Navigation* – finding places which are given by an external observer.

The structure and the dynamics of the intended behaviours are considered as being far too complex to be solved by applying analytical methods. This is not a principle argument, but concerns the state of the art of analytical methods in control theory, dynamical systems and related fields.

3. The LISA-model

The previous section introduced the context and the attacked problem class. In this section a model is described which is suggested to address the problem of balancing between general and specific modelling.

After being aware of the constraints which were applied in constructing the *LISA*-model (section 3-1), the influence of the experimenter onto the system is discussed by explaining all the implicit and explicit parameters of the model (section 3-2). Based on these 'cornerstones' of design, all components of the *LISA*-model are described in detail (section 3-3).

3-1. Constraints

The philosophy of the *LISA* system is introduced by highlighting all major issues in the design space briefly. The evaluation procedures discussed in section 6 are based on these decisions.

3-1.1 Excluded features

In order to make the intentions of the designer very clear and obvious, the aspects that are *excluded* from the constraints of the *LISA*-model, are mentioned first.

Precision

Absolute precision is not an optimization parameter here. The tasks are regarded as being solved, when the robot shows a certain behaviour measured in qualitative terms. Precision is required in specific tasks like picking and placing applications in manufacturing environments, which can be isolated and solved individually. In addition, the meaning of precision is not very clear. This is especially true when combining contradictory tasks, in which the robot has to reach some goals, but the actual trajectory is usually of no interest. Assume a scenario where a robot has to reach the end of a corridor in time and without significant damage. All these goals are given as qualitative measurements. Whether the robot moves exactly in the middle, or more to the right of the corridor is usually of no relevance for this task.

Fail-safe

A 'fail-safe' or '1-secure' system is not assumed to be realistic in any natural environment. This includes any environment that hosts humans or any other species. This point is related to the next one, because even the definition of 'fail-safe', would request a complete knowledge of the whole system.

Complete knowledge

Any natural or slightly dynamical environment (not to talk about environments, which include other moving artifacts or even biological species), cannot be described fully and realistically by a complete and fixed model at any point in time. This statement cannot be proven in a sense, but it seems to be very obvious regarding any biological species.

3-1.2 LISA-constraints

The *LISA*-system is based on some central constraints, which are assumed to be important in many natural mobile robot scenarios.

Multiple models

Hosting more than one model for the same instance in the environment is a central idea of the proposed system. A critique leading to the need of such multi-model systems was given in section 1, where especially general and specific aspects of the same environment feature were found to be relevant and should be treated simultaneously.

Adaptation

The working environment of the mobile robot is seen as the most important and especially most reliable knowledge source available. Thus the robot's world model should be generated through continuous interaction. On the other hand, some a-priori knowledge is required for quickly gathering first relevant information about the environment. In order to be useful in significantly differing opening-scenarios, the a-priori knowledge has to be kept on a general level, i.e. for instance, any kind of a-priori given environment 'map' is not considered here.

Continuous and life-long adaptation is a central assumption in the designed model.

Failure exploitation

In contrast to building a fail-safe system, the intention here is to build a failure-exploiting system. Failures are regarded as a normal part of operation and they are included in the adaptation process as any other experience. Furthermore they are welcome as explicit negative examples in the learning process, which can be speed up the adaptation significantly.

Cooperative action selection

The control structure is not intended to include a rigid scheduling of tasks or priorities. Instead of using such a scheduler, a competitive self-organizing system is considered to give better adaptability under changing conditions. This also includes the chance to do the wrong thing in a critical decision. But for the sake of flexibility, such failures are tolerated and seen as a part of the adaptation process (see above).

Realtime

Participating in a scenario, which includes any dynamics that cannot be controlled by the mobile vehicle fully (e.g. the speed of moving objects is given by masses, gravity), results in the necessity of keeping track with the speeds given by the environment or the interactions. Thus the central parts of any system taking part have to limit their computation times between the stimulus and the response by a constant (i.e. O(1)), which is given by the occurring speeds and accelerations. Usually the actual reaction time depends on the current scene complexity (typically O (log (n)), O (n·log (n)), or O (n)), but even so a maximal reaction time needs to be assumed.

On the other hand the speeds of the mobile robot should not be significantly faster than anything else around it, otherwise the environment could be treated as being static for some time-span. Of course, this is not the major problem for state-of-the-art mobile artifacts.

These real-time problems have to be attacked in multiple areas in computer science (operating and communication systems, search algorithms, planning, etc.). What matters finally is that the whole control loop including all the extraction, planning, mapping, correlating, integrating, fusing, action selection and other modules has to show the demanded (constant) timing behaviour.

3-1.3 Other aspects

The final couple of aspects are not completely excluded but not used as principles in constructing the *LISA*-system. In order to complete the picture of constraints they are just mentioned.

Sophisticated planning and reasoning methods are not a part of the *LISA*-system. Furthermore some critical real-time aspects of planners should have to be considered. When regarded as an 'off-line part' of a robotic control structure, i.e. when the robot is allowed to stop for some deeper reasoning, it can be considered as a meta level employing the *LISA*-system, and thus it is not excluded. Nevertheless it is not part of the inner design.

Explicit cooperation with similar or other robots is not discussed in this article, although a correlation between several behaviour-modules on different robots could be considered. Otherwise any other robot is included in the models as a part of the operating environment, without exploiting the possibility of cooperation.

A high-level (symbolic) man-machine interface is not in the inner focus of the system. No kind of communication is included in the spectrum of investigated behaviours yet, but an expansion to some communication oriented behaviours seems to be a field of discussion.

3-2. Parameters

In any robotic real-world problem, the set of behaviour relevant features is not fixed or known in advance. Therefore, it is not possible to equip a robot with precise feature detectors delivering all the instances relevant for the task. The idea applied here is to use either 'raw' sensor data, or to categorise the input space by applying self-organized clustering methods, which are controlled by a fixed or an adaptive **metric**. The choice of this metric is the first main parameter to control the cognition capabilities of the robot.

The next class of parameters is the **set of appraisal functions** (the **appraisal system**) determining the presetting and optimization of behaviours. The degree of adaptivity depends strongly on the designer's ability to formulate the appraisal functions in a simplistic way and on a general basis.

Finally there are a couple of parameters regarding the adaptation processes. Depending on the structures chosen, adaptation speeds, growing and shrinking rates, adaptation delays, etc. are to be controlled. The relevant parameters can be classified into **local adaptation parameters**, which influence the plasticity of the individual modules, and **cooperative adaptation parameters**, which control the coupling between the modules and their collective behaviour.

3-3. Components

The components of the *LISA*-system are shown in figure 1 and discussed individually and in detail in the sections following this introduction.

At the right hand side of figure 1, raw sensor data is introduced to the architecture, followed by a first *preprocessing and compressing* (P&C) stage. The remaining five grey areas mark the basic *LISA*-modules. The *categorization* employing the results from the P&C module, abstract the sensor data on different levels, in order to prepare them for their integration in the general and specific behaviour modules. In the *specific behaviour module* all information at a given place in space is collected and processed in order to generate an optimal behavioural answer to any specific spatial situation. Whereas the *general behaviour modules* generalized the experiences and the structures in the data stream without any respect to the global spatial situation. The *appraisal system* influences both in order to optimize the overall behaviour by means of a-priori given evaluation functions. Finally the *action selection* shown at the bottom of figure 1 collects and coordinates the different suggestions from the specific and global behavioural modules, producing the final output given to the actuators.

3-3.1 Categorization

Due to the high bandwidth of some sensor systems, preprocessing and compression (P&C) preceding the categorization modules becomes necessary. Any kind of compressor or feature extraction can be considered. The methods range from simple normalization and smoothing (e.g. in the case of whiskers, simple range measurements, or photo resistors) to complex image compression methods. Following the idea of keeping the system as adaptive as reasonable, general compression methods should be preferred to specialized feature extraction methods. The criteria for the selection of such methods is the preference to keep the general structures rather than the specific details. Due to the fact that these compression methods are usually an elimination of information, it has to be kept in mind that the ability to define metrics in order to recognize 'similar' sensor impressions has to be assured. Thus there is a trade-off between generality and precise feature extraction at this early stage of sensor data processing already.

In order to reduce the bandwidth of the sensor data stream even further, unsupervised clustering methods are employed. Metrics are defined separately for each kind of sensor data thus keeping it simple and avoiding preferences regarding one or multiple sensor modes. In systems with a limited number of sensor modes, a metric can be defined including the complete sensor data stream [24], but the model presented here is designed to be more flexible by applying new or removal of sensor modes. Thus the clustering is being done in separate self-organized maps. First results with this concept are presented in [25].

The employed method follows the dynamic topologic clustering presented in [24]. There, the number of clusters is adapted according to the current working environment and does not need to be given in advance. The neighbourhood topology is not limited to an a-priorily chosen structure, but is continuously adapted by the data stream. The method is discussed in the remaining part of this section. Each sensor mode is classified independently. Their fusion into a global representation is done at the stage of global topologic mapping producing the specific behaviours described in section 3-3.4.

Metric

Assuming an input space of $\Re^n$ in each sensor mode after preprocessing and compressing, a metric for 'similarity' has to be defined. The basis for causality – the ability to act similar under similar conditions – is set here (what is a definition if 'intelligence' given in former times). Thus the performance of the whole system depends critically on these metrics. The metric for a sensor mode a will be denoted as $\|\bullet\|_a$ in the following. The influence of the preprocessing stage (e.g. a shift to the frequency space) should be kept in mind.

Adaptation

A set of cells c_i together with weighted connections $[c_i;c_j] \in [0, 1]$ is considered as the network N. A sensor sample prototype or a 'sensor-situation' $S(c_i) \in \Re^n$ is attached to each cell c_i. The input set the network is adapted to, is considered as an infinite stream of data. Thus the input data can be stored temporally only, i.e. procedures assuming a fixed set of input data cannot be applied here. This does not imply that the input data lacks any structure. It is assumed that the characteristics of an input data stream in a closed (nevertheless dynamic and rich) environment can be described by a limited amount of prototypes and their relations.

A sensor sample S_x will change the network structure in the following way. A cell c_{opt} with the smallest distance to the input signal S_x is determined according to:

$$\forall c_i \in N: \quad \|S(c_{opt}) - S_x\| \le \|S(c_i) - S_x\| \tag{1}$$

The optimal matching cell c_{opt} together with neighbours defined by its connections are adapted by:

$$S(c_{opt}^{new}) = S(c_{opt}) - \varepsilon_o \cdot (S(c_{opt}) - S_x) \tag{2}$$

$$\forall c_j | [c_{opt};c_j] > 0: \quad S(c_j^{new}) = S(c_j) - \varepsilon_n \cdot (S(c_j) - S_x) \tag{3}$$

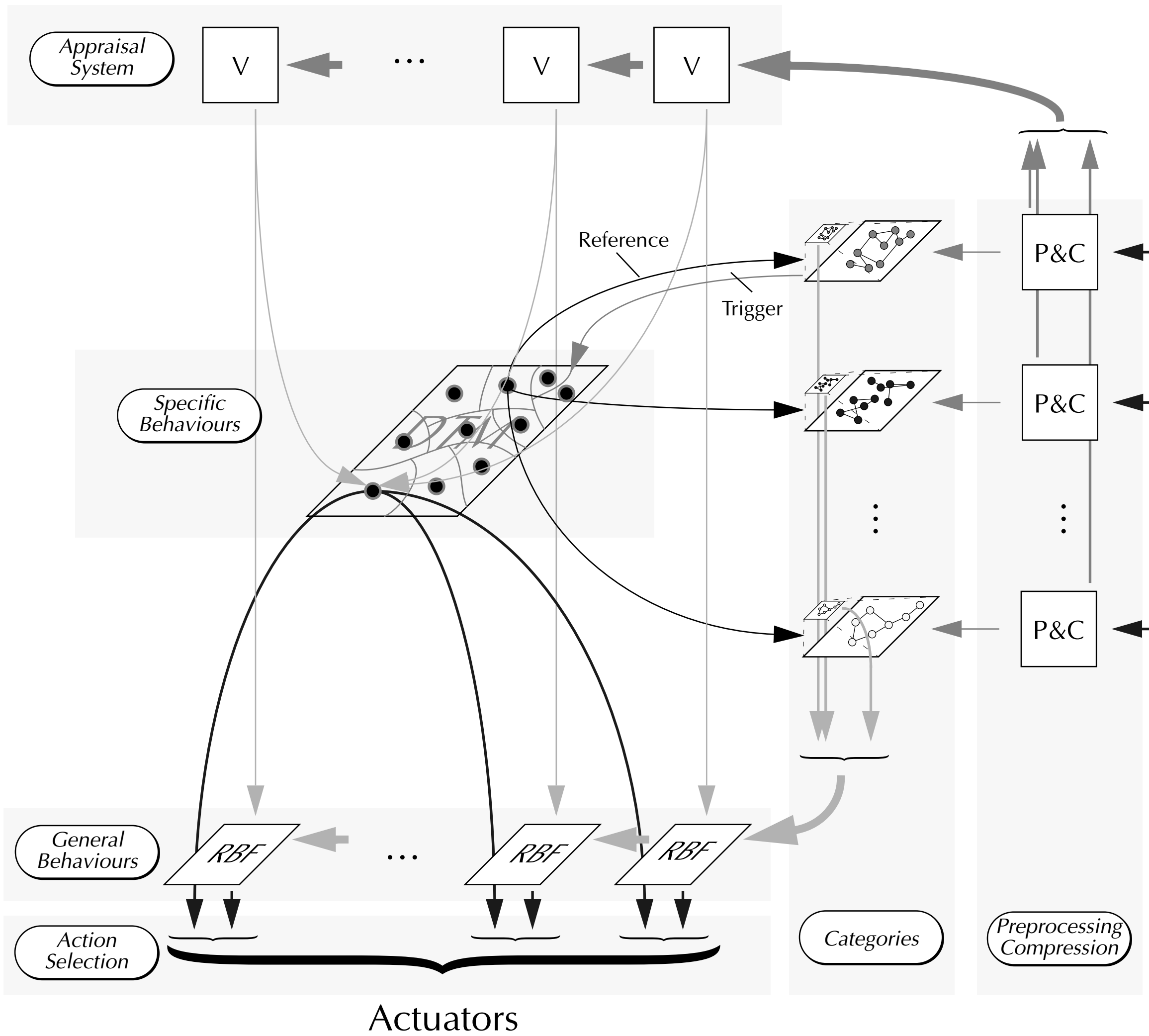

figure 1 : Components

Applying a gaussian kernel on the neighbourhood radius and a sigmoid function on the adaptation rate, an alternative version for (2) and (3) can be formulated:

$$S(c_{opt}^{new}) = S(c_{opt}) - \varepsilon_o \cdot \left(\frac{2\alpha}{1 + e^{-\frac{2\beta}{\alpha}\|S(c_{opt}) - S_x\|}} - \alpha \right) \cdot (S(c_{opt}) - S_x) \quad (4)$$

$$\forall c_j | [c_{opt};c_j] > 0 \colon S(c_j^{new}) = S(c_j) - \varepsilon_n \cdot e^{-\frac{\|S(c_j) - S_x\|^2}{2\sigma^2}} \cdot \left(\frac{2\alpha}{1 + e^{-\frac{2\beta}{\alpha}\|S(c_j) - S_x\|}} - \alpha \right) \cdot (S(c_j) - S_x) \quad (5)$$

where α, β, σ scaling the sigmoid function and the gaussian kernel respectively; ε_o and ε_n are parameters for the adaptation speed adjusted according to:

$$\varepsilon_o^{new} = \begin{cases} \varepsilon_o \cdot \varepsilon_\Delta & ;\|S(c_{opt}) - S_x\| \le d_{acc} \\ \varepsilon_o^{init} & ;d_{acc} < \|S(c_{opt}) - S_x\| \end{cases} \quad (6)$$

$$\varepsilon_n^{new} = \begin{cases} 0 & ;\|S(c_{opt}) - S_x\| \le d_{sat} \\ \varepsilon_n \cdot \varepsilon_\Delta & ;d_{sat} < \|S(c_{opt}) - S_x\| \le d_{acc} \\ \varepsilon_n^{init} & ;d_{acc} < \|S(c_{opt}) - S_x\| \end{cases} \quad (7)$$

where d_{acc} is an accuracy limit, which triggers the full adaptation speeds ε_o^{init}, ε_n^{init}; d_{sat} is a saturation limit stopping the adaptation of the neighbours.

$$0 < \varepsilon_\Delta < 1 \,;\, 0 < \varepsilon_n < \varepsilon_o < 1 \,;\, d_{sat} < d_{acc} \quad (8)$$

The total adaptation movement $M(c_i)$ for each cell c_i is accumulated over all adaptation steps following

$$M(c_{opt}^{new}) = M(c_{opt}) + |\|S(c_{opt}) - S_x\|| \quad (9)$$

$$\forall c_j | [c_{opt};c_j] > 0 \colon$$

$$M(c_j^{new}) = M(c_j) + |\|S(c_j) - S_x\|| \quad (10)$$

and indicates the need for a change in the network structure. Finally an update counter $U(c_i)$ is incremented for the cell c_{opt} in each adaptation step. In connection with $M(c_i)$, it describes the reliability $R(c_i)$ of the prototype c_i:

$$R(c_i) \sim \frac{U(c_i)/U_t}{M(c_i)/M_t} \quad (11)$$

where U_t and M_t are the total number of network updates, and the total amount of movements, respectively:

$$M_t = \sum_{t_i = 1}^{k} |\|S(c_{opt}(t_i)) - S_x(t_i)\|| \quad (12)$$

with k is the total number of updates on the network. In order to achieve a better stability each situation S_x is presented to the network multiple times. Moreover a delay between the correlation process (described later) and the actual adaptation in the individual sensor modes has been found useful. This delay leads to a learning queue, which is more frequently sampled for the adaptation process than being shifted by inserting a new sensor reading.

The adaptation discussed up to now requires a well defined network. In general, an appropriate initialization is hard to be determined in advance. Growing and shrinking techniques starting from an empty network can construct adequate network structures.

Growing and Shrinking

Two kinds of insertion and two kinds of deletion of cells are introduced. They are characterized by the following terms / situations: insertion due to a miss-classification (**spontaneous insertion**), insertion due to a long term under-representation (**statistical insertion**), removal due to a multiple mismatch of local expectations (**local removal**), removal due to limited resources (**overflow removal**).

For the discussed kinds of insertion, two cells are important. The best matching cell $c_{opt,prev} \ne c_{opt}$ of the previous adaptations and the second best matching cell $c_{opt,2} \ne c_{opt}$ from the current adaptation step:

$$\forall c_i \in N \colon$$

$$\|S(c_{opt}) - S_x\| \le \|S(c_{opt,2}) - S_x\| \le \|S(c_i) - S_x\| \quad (13)$$

Spontaneous insertion: Regarding the degree of miss-classification indicated by $\|S(c_{opt}) - S_x\|$ exceeding a certain limit ϑ, the cell c_{opt} and their neighbours are not adapted to the new situation but a new cell c_{si} is introduced representing the new input value (in case of an empty network, this routine will generate the first cell). Thus equations (2), (3) and (6) will be replaced by

$$S(c_{si}) = S_x \,;\, \text{if } \|S(c_{opt}) - S_x\| > \vartheta \quad (14)$$

Furthermore the cell c_{si} is treated as c_{opt} in the following. The connection weights are adjusted now, in order to use the coincidence of activities or the order of appearance as an indicator for neighbourhood or similarity. If activation coincidence is a measurement for similarity (Hebbian style adaptation), the connection to the second best matching unit is emphasized:

$$[c_{opt};c_{opt,2}] = 1 \quad (15)$$

(see [8] for an introduction concerning systems based on this neighbourhood definition).

Alternatively, if the time-sequence of activations is typical for a specific sensor mode (e.g. position measuring devices), the connection between two consecutive best matching cells is strengthened:

$$[c_{opt};c_{opt,prev}] = 1 \quad (16)$$

Local Removal: All other m connections diverting from c_{opt} are reduced accordingly. Assuming that the chance of an neighbouring cell to become the second best or the best matching cell is reduced by the number of neighbours , the reduction term w_r is scaled by m.

$$\forall c_j | [c_{opt};c_j] > 0 :$$

$$[c_{opt};c_j]^{new} = \begin{cases} [c_{opt};c_j] - \frac{w_r}{m} & ;[c_{opt};c_j] > \frac{w_r}{m} \\ 0 & ;[c_{opt};c_j] \le \frac{w_r}{m} \end{cases} \quad (17)$$

In case that a cell loses its only connection in such a step, it is removed. This occurs, if the environment changes more quickly than is the adaptation procedures, or the feature represented by that cell has disappeared completely.

Statistical insertion: Observing the amount of movement of the cells in the network (measured in terms of $M(c_i)$), areas in the input space which are not represented adequately by prototype cells can be identified. Thus the total adaptation movement $M(c_i)$ is checked against a limit ∂ for c_{opt} and all its direct neighbours. If a cell c_r is found where

$$M(c_r) > \partial \quad (18)$$

a second cell $c_{r,2} \ne c_r$ with $[c_r;c_{r,2}] \ge 0$ and

$$M(c_r) \ge M(c_{r,2}) \ge M(c_j), \forall c_j | [c_r;c_j] \ge 0 \quad (19)$$

is determined. A new cell c_{sti} is generated and equipped with connections to c_r and $c_{r,2}$:

$$[c_r;c_{sti}] = 1 ; [c_{sti};c_{r,2}] = 1 \quad (20)$$

The attached sensor situation $S(c_{sti})$ is assigned to a value between c_r and $c_{r,2}$:

$$S(c_{sti}) \in [S(c_r), S(c_{r,2})] \quad (21)$$

The actual value for $S(c_{sti})$ depends on the a-priori knowledge about the characteristics of the sensor mode as well as the metric chosen.

Overflow removal: In case that the search on the cells in the network cannot be limited to any local area around the cell $c_{opt,prev}$, the total number of cells must be limited in order to keep real-time constraints. Thus if the number of cells exceeds a number n_{max}, the cell c_u with the smallest reliability $R(c_u)$ will be deleted.

In the other case that a correlation between the time sequence of sensor readings and their neighbourhood in the sensor space can be found, the search area can be limited by means of the topological neighbourhood around the last active cell $c_{opt,prev}$. Thus the total number of cells in the network need not to be restricted.

Principal Component Analysis

The prototypes $S(c_i)$ are applied furtheron as primitives for the specific world model discussed in section 3-3.4. The underlying sensor space $\Re^n$ is still the same as at the end of the P&C stage. In order to get a higher level of abstraction (and a further bandwidth reduction) as needed for the general behaviour modules, a further reduction of dimensions is required. Due to the fact that the general behaviour modules will fuse the different sensor modes once again and thus add up all the dimensions from the different sensor categorization systems, the bandwidth reduction becomes an important factor, regarding realtime abilities.

Assuming that the actual structure and topology in the data of a specific sensor mode can be represented in a subspace $\Re^{n_p}$, the following extension of the presented structure will lead to a principal component analysis of the data stream (concerning the n_p most important eigenvectors) and deliver the topology preserving projection of the prototypes $S(c_i)$ to $\Re^{n_p} \subset \Re^n$. The idea is to keep the topographical relationships in $\Re^{n_p}$ consistent to the one $\Re^n$. Thus the relative distances between cells in $\Re^n$ should be equivalent to the distances in $\Re^{n_p}$.

An additional prototype $S_p(c_i)$ is introduced for each cell c_i. $S_p(c_i)$ is generated and updated in parallel to $S(c_i)$. Thus extensions are needed for the statistical (21) and spontaneous insertion techniques (14) and the adaptation step (2) and (3).

In each *statistical insertion* step, $S_p(c_{sti})$ is initialized equivalently to $S(c_{sti})$:

$$S_p(c_{sti}) \in [S_p(c_r), S_p(c_{r,2})] \quad (22)$$

Assuming that an explicit projection function between $S(c_i)$ and $S_p(c_i)$ is not available (on-line), (22) is implemented by employing a mean value:

$$S_p(c_{sti}) = (S_p(c_r) + S_p(c_{r,2}))/2 \quad (23)$$

In case of an *spontaneous insertion*, $S_p(c_{si})$ is initialized as:

$$S_p(c_{si}) = S_p(c_{opt}) \quad (24)$$

and is adapted by determining the immediate neighbours of c_{opt} and employing them for the adaptation of $S_p(c_{si})$:

$$\forall c_j | [c_{opt};c_j] > 0 :$$

$$S_p(c_{si}) = S_p(c_{si}) - \varepsilon_p \cdot (S_p(c_{si}) - S_p(c_j)) \cdot \left(1 - \frac{\| S_x - S(c_j) \|}{\| S_p(c_{si}) - S_p(c_j) \|}\right) \quad (25)$$

Finally $S_p(c_{si})$ is adjusted by c_{opt} itself:

$$S_p(c_{si}) = S_p(c_{si}) - \varepsilon_p \cdot (S_p(c_{si}) - S_p(c_{opt})) \cdot \left(1 - \frac{\| S_x - S(c_{opt}) \|}{\| S_p(c_{si}) - S_p(c_{opt}) \|}\right) \quad (26)$$

where ε_p is defined analoguous to ε_o in (6).

The additional prototypes introduced above are *adapted* furtheron in parallel to the procedures described in (2) and (3). For any adapted cell c_u, the low dimensional prototypes are adjusted according to the distances in $\Re^n$:

$$\forall c_u \in \{c_{opt}\} \cup \{c_k | [c_{opt};c_k] > 0\}:$$
$$\forall c_j | [c_u;c_j] > 0:$$

$$S_p(c_u) = S_p(c_u) - \varepsilon_p \cdot (S_p(c_u) - S_p(c_j)) \cdot \left(1 - \frac{\|S_x - S(c_j)\|}{\|S_p(c_u) - S_p(c_j)\|}\right) \tag{27}$$

Routines for deleting and manipulating connection weights are not needed, because the criteria for deletion and insertion are delivered by the main network, i.e. from cells in $\Re^n$.

The low dimensional prototypes $S_p(c_i)$ serve as a base for the topology-preserving reconstruction of the original data-space in $\Re^{n_p}$. Including neighbours of c_{opt} in the reconstruction process and applying Gaussian kernels lead to:

$$S_x^{n_p} = \frac{\sum_{\{c_i\}} \frac{S_p(c_i)}{e^{-\frac{\|S(c_i) - S_x\|^2}{2\sigma_i^2}}}}{|\{c_i\}|} \tag{28}$$

where $\{c_i\} = \{c_{opt}\} \cup \{c_k | [c_{opt};c_k] > 0\}$

and:

$$\sigma_i = \frac{\sum_{\{c_j | [c_i;c_j] > 0\}} \|S(c_i) - S(c_j)\|}{|\{c_j\}|} \tag{29}$$

i.e. the mean distance of c_i to all its neighbours.

The compression function

$$g_{PCA}: \Re^n \to \Re^{n_p} \tag{30}$$

defined piecewise with (28) and (29) is employed by the general behaviour modules, assuming that the reduced bandwidth allows for short reaction times and keeping the characteristics of the current sensor situation.

g_{PCA} can also be generated directly, employing for instance single layer networks with Hebbian style learning applying the adaptation rules suggested by Oja [17] (symmetrical update) or Sanger [21] (hierarchical update). The resulting networks are simple and converge quickly, but each sensor sample requires updates at all network units, which is a major drawback under hard realtime constraints. Improving the convergence speed, lateral, inhibitory connections between the output units are introduced by Rubner and Schulten [20] for the asymmetrical and by Brause [2] for the symmetrical case. The adaptation speed was improved, but the amount of required weight updates was incremented as well (due to the new weights at the lateral connections). Nevertheless this direct generation of g_{PCA} can be considered here, for sensor modes where the dimensions n and n_p are both small and the main characteristics of the sensor data stream can be extracted continuously. The benefit would be a globally consistent PCA-function instead of a piecewise defined version given by the approximation method introduced (22)-(28).

3-3.2 Appraisal System

The appraisal system influences on the behaviour of the robot significantly. It is a pre-coded non-adaptive part of the system. In a sense it can be regarded as its ‘instincts’ or ‘genes’. For any desired behaviour a current state of evaluation and a gradient of this state are delivered, which can be utilized by reinforcement schemes in the optimization of the general as well as of the specific behaviours. Some of them (e.g. collision avoidance) can even be pre-trained based on the relevant appraisal module.

An evaluation Ξ_a given for a specific behaviour a is determined following the form:

$$\dot{\Xi} = g(\Xi, I, S) \tag{31}$$

where I is the current system input (the set of all sensor inputs after being preprocessed and compressed), and S is the current status (all inner states like the energy reservoir or the pain-value). In order to keep the appraisal results comparable, Ξ is limited to the range of $[0, 1]$, where ‘1’ means that the behaviour is completely satisfied. For each appraisal function correlating to a specific behaviour, only a small subspace of $I \times S$ needs to be considered.

3-3.3 General Behaviour Modules

The general behaviours are formed given an evaluation from the appraisal system. Each behaviour is modelled separately. Therefore representations are evolved for very individual instead of general purposes. Besides the appraisal signal, a behaviour module gets information about the current sensor situation, which can include a short sequence of sensor impressions if that is adequate. The output is a set of st probabilities reflecting the tendencies of different steering directions to increase the value from the appraisal system and a value representing the individual wish or urgency to control the robot in order to improve the local behaviour’s performance.

$$gbm: \Re^{n_p} \times [0, 1] \to [0, 1]^{st} \times [0, 1] \tag{32}$$

These modules will behave in a similar way in similar situations, despite the fact that similar situations can occur in different global contexts. Many different implementations of this association problem are possible. The main constraints that have to be fulfilled are quick adaptation, small computational complexity in the association (forward) path and a continuous learning ability.

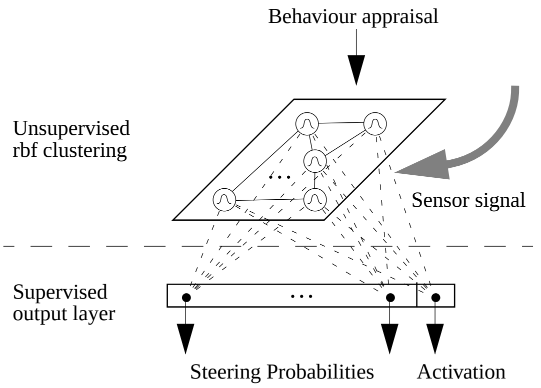

figure 2 : Supervised rbf clustering

The supervised radial basis function clustering network shown in figure 2 offers many more potential solutions than described here (see e.g. Moody and Darken [13] for a general discussion on rbf-networks and related concepts). The sensorial input $S_x^{n_p}$ from (28) is clustered together with the evaluation signal in an unsupervised preprocessing step. The ‘positions’ of the cell-clusters and ‘distances’ between them are employed as the position and variance parameters of Gaussian functions. A supervised output layer sums up the product of the activation values and the associated weights in the input space. By applying the structures described in section 3-3.1 for the clustering part, the number of Gaussian kernels as well as their relations can kept dynamical. Of course, the metric has to be different, in order to avoid to reproduce the clustering from the categorization, but to get a more general one. The similar concept of counterpropagation introduced from Hecht-Nielsen in [9] can be considered as an alternative. Standard competitive learning with a fixed amount of categories instead of dynamic clustering is applied there, but the overall structure is identical.

The unsupervised clustering in the first layer is driven completely by the input stream, i.e. it is especially not controlled by the intended outputs. Thus this first layer can be shared between the different general behaviour modules, reducing the computational complexity significantly.

The cascade correlation architecture introduced by Fahlmann and Lebiere in [6] offers another potential effective implementation. Their structure starts with a simple perceptron, and adds hidden units (which are pre-trained ‘off-line’ i.e. without a connection to the output layer) sequentially whenever the required accuracy is not fulfilled. The resulting structure is compact and fast, but cannot shrink down again. The applied adaptation is out of the family of perceptron adaptation methods, i.e. the error is never propagated through multiple layers, leading to a significantly faster adaptation than standard multilayer backpropagation systems. Due to the fact that anything is controlled by the supervised output layer, none of the elements from the cascade correlation architecture can be shared with other general behaviour modules.

3-3.4 Specific Behaviour Module

The specific behaviours produce the same output structure as the general ones, but they are based on a different kind of information. The global context, which is the topological relation to all other recognized situations as well as the accumulated experience made in the current situation is considered.

The specific world model of the system is built up similarly to the dynamic clustering techniques presented in [24]. Extending the formally discussed method, local specific behaviours are attached to each situation in the topological space. The clustering in the topological space is triggered completely by the clustering systems from the different sensor modes, i.e. each time when a significantly different situation is found in any of the sensor modes, a new geometric cluster is assumed for the global model.

The technical details of this idea are discussed in the remaining part of this section. As a first step approaching this specific behaviour module, it can be regarded as a dynamic topologic clustering system for the position information. The adaptation steps (2) and (3) as well as the growing and shrinking methods are applied analogous to the clustering process in the categorization modules, but a new cell insertion strategy, the **external insertion**, is introduced. Without that additional growing method, the position clustering would result in a homogeneously filled up, two dimensional free space area, which is not suitable for self-localization or any other correlation process.

Assuming categorization networks N_1 to N_M and a topologic, position clustering network N_t, a reference function r is defined for each cell $c_t \in N_t$:

$$r: N_t \to N_1 \times \ldots \times N_M \,;\; r(c_t) = (c_{i_1}, \ldots, c_{i_M})$$

$$r_k: N_t \to N_k \,;\; r_k(c_t) = c_{i_k} \qquad (33)$$

with $c_{i_1} \in N_1, \ldots, c_{i_M} \in N_M$. r approximates the function r_{opt} defined by

$$r_{opt}(c_{opt_t}) = (c_{opt_1}, \ldots, c_{opt_M}) \qquad (34)$$

at any time, i.e. it is tried to keep a correlation between all sensor modes including the dead-reckoning information. $r(c_i)$ is initialized with $r_{opt}(c_i)$ in any spontaneous insertion or statistical insertion step.

Whenever in a sequence of l adaptation steps the intended value of r is mismatched and r_{opt} is stable:

$$\forall i, j = 1, \ldots, l: \qquad (35)$$

$$(r(c_{opt_i}) \neq r_{opt}(c_{opt_i})) \wedge (r_{opt}(c_{opt_i}) = r_{opt}(c_{opt_j}))$$

an *external insertion* is triggered and treated analogous to the spontaneous insertion steps (14), (16).

r is never changed for a specific cell. Whenever the environment changes at a certain spatial situation, new topologic cells are introduced and the old ones are 'washes out' after a certain amount of time.

Correlation (Self-Localization)

The basis for the position clustering network are the mobile robot's three degrees of freedom. Due to the influence of drift effects and other errors, the dead-reckoning information has to be closely correlated with the internal spatial model (including all the references to the categorization modules), in order to keep it locally consistent. This mutual stabilizing technique is very useful when applied to local world models. But, in fact, each representation is updated with the faults, errors and noise from the other. This principal problem prevents a globally consistent world model of arbitrary size including position when only local information is available.

One strategy used to stabilize the internal position is to delay the integration of the current sensor situation S_x by a number of adaptation steps (see section 3-3.1). In this way the internal position is correlated on the basis of the formerly integrated world-knowledge at this point, and not on the base of S_x, which would not make any sense.

The correlation of (x, y) and the orientation α with the accumulated world models is performed independently. First the correlation of the Cartesian (x, y) information is discussed.

A position estimation p is calculated as a mean value of all position estimations p_k relating to the sensor modes k:

$$p = \sum_{k=1}^{M} \frac{pos(p_k((c_{opt_t}), \{\ \}))}{|set(p_k((c_{opt_t}), \{\ \}))|} \tag{36}$$

where p_k is defined as:

$$p_k(c, B) = \begin{cases} (0, 0, B) \\ \quad ; \|\|S(r_k(c)) - S(r_k(c_{opt}))\|\| > d_k \\ (S(c) + np_k(c, B), B \cup ns_k(c, B)) \\ \quad ; \|\|S(r_k(c)) - S(r_k(c_{opt}))\|\| \le d_k \end{cases} \tag{37}$$

$$np_k(c, B) = \sum_{c_j | ([c, c_j] > 0) \wedge (c_j \notin B)} pos(p_k(c_j, B \cup \{c\})) \tag{38}$$

$$ns_k(c, B) = \bigcup_{c_j | ([c, c_j] > 0) \wedge (c_j \notin B)} set(p_k(c_j, B \cup \{c\})) \tag{39}$$

with:

$$p_k : N_t \times \{N_t\} \to \Re \times \Re \times \{N_t\} \tag{40}$$

$$pos(x \in \Re, y \in \Re, B \in \{N_t\}) = (x, y) \tag{41}$$

$$set(x \in \Re, y \in \Re, B \in \{N_t\}) = B \tag{42}$$

p_k accumulates all cells and their positions of similar sensor situations in the (sensor space k) surrounding c_{opt} in a radius of d_k. The current internal position (x, y) is then adapted towards p by:

$$(x, y)^{new} = (x, y) - \varepsilon_p \cdot ((x, y) - p) \tag{43}$$

where ε_p reflects the reliability of the estimation p in relation to the dead-reckoning information.

Assuming a stable (x, y) information, the correlation of the robot's orientation is done on the basis of the cell c_{opt} only. Otherwise different perspectives of the same place would be considered in order to re-calibrate the orientation at a certain place, which would not make sense. The measurement of the actual orientation estimation has to be defined for each sensor mode individually. For rotation-symmetrical sensor systems a cross correlation or a method suggested in [24] could be adequate. Finally α is corrected analogously to the Cartesian information in (43).

Specific Behaviours

The function sb_l represents specific experiences regarding behaviour $l \in 1 \ldots N$ for any unit in N_t, with:

$$sb_l : N_t \to [0, 1]^{st} \times [0, 1] \tag{44}$$

Identical to the general behaviour modules, a set of st steering direction probabilities and a value representing the individual wish to control the robot for this specific behaviour is delivered. Thus the results can be compared, coordinated, and integrated immediately together with the results from each general behaviour module in the action selection phase described next.

3-3.5 Action Selection

The action selection problem has to be solved in two stages. First and for each behaviour the outputs of the general and the specific world model have to be processed. Due to the fact that the output formats of the general and the specific models are identical both could be superimposed. An obvious criterion for the degree of influence from each module is the individual amount of experience their suggestions are based on (which is encoded in the activation-[wish]-output attached to every suggested set of steering directions).

Second the appropriate behaviour for the current situation has to be chosen. The selection is done by a winner-take all system on the activation values with some integrating and thresholding components added. Once a behaviour is selected, its chance to be selected in the next step again is enhanced thus generating some continuity in the overall behaviour of the system.

4. Adaptation

The adaptation method is a central aspect in any mobile robot scenario, which is characterized by a high degree of dynamics and flexibility.

As a result of the adaptation, the robot should behave according to his appraisal system, the actual environment *and the robot itself*. This includes the robot's static and dynamic characteristics, but also his behaviour during the learning phase. Assuming that the adaptation is not a straightforward gradient decent but a process overcoming local minima, there is a (usually strong) stochastical component included. One should be aware that most models are constructed on the basis of this stochastic decisions, which are taken during the adaptation. Thus the result can be optimal in sense, but is usually not unique. This phenomenon has some parallels to the way how human beings explain concepts using the metaphors and ideas made before, where a lot of these terms are build up occasionally and with strongly individual backgrounds. Furthermore neurobiological observations suggest that the same behaviour is represented by different activation patterns in the same species. For instance, Freeman [7] reports that each time a change is made in the olfactory memory store by adding a new stimulus or changing the reinforcement contingency of a stimulus, the central patterns for other stimuli also change. Finally, to say it with a philosophical citation: 'All that we are is the result of what we have thought: it is founded on our thoughts, it is made up of our thoughts.' – Dhammapada I.I.

Adaptation is accomplished with respect to both the individual behaviours, and their coordination. Whereas the behaviours themselves can be adapted applying reinforcement learning methods, coordination produces a further complex level of dynamics. Assuming a distributed architecture, the coordination is performed in terms of cooperative decision making, i.e. every module tries to estimate its own current significance with respect to the whole system (expressed in the activation-value delivered by each behaviour-module). A kind of global significance is delivered by the appraisal system, where some general relations are introduced by applying the same metric for all behaviour evaluation functions. The actual significance of an individual behaviour depends on both these global relations *and* the current situation (or the history leading to that situation) as well. The influence of this second part is adapted in every behaviour module individually.

Adaptation to the actual environment is implemented in similar ways regarding pre-training and on-line learning in the general models. The specific models are not pre-trained, because they need to know the real situations before they can adapt to them. Nevertheless the adaptation criteria are the same for the specific and the general models, namely the individual appraisal function for each intended behaviour.

Pre-training

Applying dynamic clustering methods with supervised radial basis functions or similar systems in the output layer (see section 3-3.3), these modules can be pre-tuned together with the categorization networks employing noisy, arbitrary input data and the preset appraisal function for each behaviour. Pre-training does not make sense for every kind of behaviour. Thus some behaviours have to be left open for arbitrary exploration during the on-line phase.

On-line-learning

During the real-world tests, the general as well as the specific behaviours are continuously adapted according to the appraisal functions and the impressions and interactions with the environment. The evaluations are the same for specific and general modules, because they model the same behaviours. With general models, the adaptation is identical to the pre-training phase, except that the data comes from the actual environment now. In the specific model, the adaptation depends on the general behaviours, the neighbourhood values in the topological layer, the history at the specific place in space, and the appraisal functions.

5. What could be shown?

One of the main purposes of the proposed model is to get a closer insight into the meaning of general and specific models in the context of several intended behaviours. The behaviours based on local or global world models can be faded or switched. Therefore, significances in different behaviour modes could lead to a better understanding of how specific and general knowledge about the world is organized.

The second aspect of the model is the superposition of behaviours, the combination or selection process itself. The bottleneck of the need to execute parallel actor sequences in a strictly sequential manner is an open question. This model could serve as a basis for structured experiments regarding this problem.

6. Intended Experiments

The main purpose of the suggested model is to investigate how specific and general world models can be combined in different tasks. Thus some well defined test scenes with a limited but relevant number of features is to be constructed. The basic elements in these test scenes are:

- *The nest:* A static place in the environment where the robot should move to in a regular manner. It is regarded to be a safe place, i.e. if a certain amount of pain is being accumulated at the robot during it´s travels, it is recommended to find the nest and stay there for a while.

- *Obstacles:* Passive objects, which cannot be eliminated, but some of them can be moved. Thus the robot can actively change its geometric environment.
- *Energy sources:* In order to survive, the robot needs energy in a regular manner. The robot can accumulate energy up to a certain limit, whereas any action (including calculations) consumes energy.
- *Pain sources:* They can reduce the on-board energy or lead to other long term effects. Moreover they can move and even follow the robot.
- *Fun sources:* Some places should be preferred for some reasons (given by the feel-good value function). This idea is introduced in order to have an instrument for further structuring the test environments.
- *Partners:* This kind of objects is introduced as an externally given navigation goal. The partner can be other robots or is just placed by the experimenter as an unforeseen behavioural goal. Moreover, these objects are a gateway to practical applications in the mobile robot field.

All experimental environments are a configuration of these elements. Features not detectable locally, but essential for some behaviours are appreciated. Therefore the need for a context-aware, global, specific behaviour can be forced. Moreover some situations and the appropriate behaviours should differ significantly from the typical case for the same reason.

The set of appraisal functions will not change over different environments. They are regarded as species-typical pre-programming.

7. Conclusions

It has been argued in this article that neither a purely local, reflexive, and general nor a global, context-aware, and specific model of a complex system consisting of a mobile robot, its environment, and an a-priori given appraisal system is promising to face problems in open and dynamic scenarios. Overcoming the drawbacks of both approaches, a system which is explicitly built on the combination or coexistence of both methods is suggested.

As a central point, this article likes to promote the discussion about principles and categories on world models in the context of global and specific behaviours. A physical robotics environment supporting (or rejecting) the proposed ideas will be set up.

The author likes to thank Bärbel Herrnberger, Ralf Möller, Rolf Pfeifer, and Herbert Jaeger for fruitful discussions supporting this work.

References

[1] S. Amari
Dynamics of pattern formation in laternal-inhibition type neural fields
Biological Cybernetics, 27, pp. 77-87, 1977

[2] Rüdiger W. Brause
A symmetrical lateral inhibition network for PCA and feature decorrelation
Proc. of the International Conference on Artificial Neural Networks (ICANN) ‘93, pp. 486-489, Springer, Amsterdam, The Netherlands

[3] Thomas Edlinger, Ewald von Puttkamer
Exploration of an indoor environment by an autonomous mobile robot
Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS ‘94), Munich, Germany, Sept. 12-16, pp. 1278-1284

[4] A. Elfes
Sonar-based real-world mapping and navigation
IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3, pp. 249-265, 1987

[5] Christoph Engels, Gregor Schöner
Dynamic fields endow behaviour-based robots with representations
Robotics and Autonomous Systems 14 (1995), pp. 55-77, Elsevier, Amsterdam

[6] Scott E. Fahlmann, Christian Lebiere
The cascade-correlation learning architecture
Technical report CMU-CS-90-100, August 29, 1991, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

[7] Walter J. Freeman
Qualitative overview of population neurodynamics
Chapter in: Ventriglia F (ed.) Neural Modelling and Neural Networks. New York, Pergamon Press. pp 185-216, 1994

[8] Bernd Fritzke
A growing neural gas network learns topologies
G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), Advances in Neural Information Processing Systems 7, MIT Press, Cambridge MA, 1995, p. 625-632

[9] Robert Hecht-Nielsen
Counterpropagation networks
Applied Optics, Vol. 26, No. 23, Dez. 1987

[10] Herbert Jaeger, Thomas Christaller
Dual dynamics: Designing behaviour systems for autonomous robots
to appear in the proc. of Artificial Life & Robotics (AROB) ‘97, Beppu, Kyuchu, Japan

[11] Benjamin J. Kuipers, Yung-Tai Byun
A robust, qualitative method for robot spatial learning
Proceedings of the AAAI 1988

[12] Pattie Maes
Behaviour-based artificial intelligence
Proceedings of the second international conf. on the simulation of adaptive behaviour “From animals to animats”, Honolulu, Cambridge MA, MIT Press, 1992

[13] J. Moody, C. Darken
Fast learning in networks of locally-tuned processing units
Neural Computation 1, pp. 281-294

[14] Takayuki Nakamura, Seiichi Takamura, Minoru Asada
Behaviour-based map representation for a sonar-based mobile robot by statistical methods
Proc. of RSJ/IEEE International Conference on Intelligent Robots and Systems, 1996

[15] Ulrich Nehmzow
Experiments in competence acquisition for autonomous mobile robots
Ph.D. Thesis - University of Edinburgh 1992

[16] Stefano Nolfi
Using emergent modularity to develop control systems for mobile robots
Technical report 96-04, Department of Neural Systems and Artificial Life, Institute of Psychology, C.N.R., Rome, 1996

[17] E. Oja
Neural networks, principal components, and subspaces
Intl. Journal of Neural Systems 1, pp. 61-68, 1989

[18] Rolf Pfeifer
Symbols, patterns, and behaviour: Towards a new understanding of intelligence
Proc. of the Japanese Conference on Artificial Intelligence, 1996

[19] Tony J. Prescott, John E. W. Mayhew
Adaptive local navigation
Technical report, Artificial Intelligence and Vision Research Unit, Sheffield University, Sheffield S10 2TN, United Kingdom, 1994

[20] J. Rubner, K. Schulten
Development of feature detectors by self-organization
Biological Cybernetics 62, pp. 193-199, 1990

[21] T.D. Sanger
Optimal unsupervised learning in a single layer linear feedforward neural network
Neural Networks 2, pp. 459-473, 1989

[22] Luc Steels
Intelligence – dynamics and representation
Luc Steels (ed.) The biology and technology of intelligent autonomous agents, NATO ASI Series F: Computer and Systems Sciences, Vol. 144, pp.72-89, Springer, 1995

[23] Bill Triggs
Model-based sonar localisation for mobile robots
Robotics and Autonomous System 12 (1994), pp. 173-186, Elsevier, Amsterdam

[24] Uwe R. Zimmer
Robust world-modelling and navigation in a real world
Neurocomputing Vol. 13, I. 2-4, pp. 241-258, Elsevier Science, Amsterdam, 1996

[25] Uwe R. Zimmer
Minimal qualitative topologic world models for mobile robots
Proc. of Artificial Neural Networks and Expert Systems '95 (ANNES ‘95), November 20-23, 1995, Dunedin, New Zealand