Acquisition Of Basic Mobility Skills From Human Demonstrations

Michael Kaiser, Michael Deck, Rudiger Dillmann
Institute for Real-time Computer Systems & Robotics
University of Karlsruhe
D-76128 Karlsruhe
GERMANY

Abstract: Reactive control of a mobile robot requires to map robot's perceptions into actions by means of a strategy that is goal-oriented. To explicitly encode this strategy is not an easy task. Specifically, it cannot be assumed that users of future service robots will be able to perform this kind of low-level robot programming. What is sought for is a method that allows to intuitively program the robot without requiring knowledge about robot's hardware, its sensor system, robot's perceptions and actions, and the actual relationship between them In this paper, the acquisition of basic mobility skills from human demonstrations is advocated to be such a method. Based on data obtained from manual operation of the robot only, a description of the demonstrated skill is generated, that is operational with respect to the robot. Since human demonstrations are neither optimal nor do they cover the real state space sufficiently, skill adaptation and extension are also considered.

Keywords: Machine Learning, Robotics, Man-Machine Systems, Programming Support

1. Introduction

Despite the diversity of mobile robot applications, and of the environment they are operating in, two characteristic modes of operation can always be distinguished (see also [34]):

> **Model-based operation**, including path planning and execution on the basis of an a priori and possibly continuously refined geometrical world model. This also includes mission planning, which might be based on a given model of the world's "semantics."
>
> **Reactive operation**, involving a direct coupling between robot's sensors and its actuators. Here, the next action of the robot is determined by the current sensor readings, possibly their history, and the current goal. These operations will from now on be referred to as the *basic* or *elementary skills* of the robot.

As modelling and planning makes only sense down to a certain level of abstraction, usually both modes of operation are combined. Then, elementary skills represent the interface between the planning and the control level in robot's architecture. They also determine the basic operators available for planning: Only if the robot is able to associate a symbolic operator with a sequence of actions that are possibly dependent on its perceptions, i.e. only if the robot can operationalize the operator by applying a particular skill, using this operator at the planning level makes sense [17]. This is obviously related to the problem of symbol grounding [11].

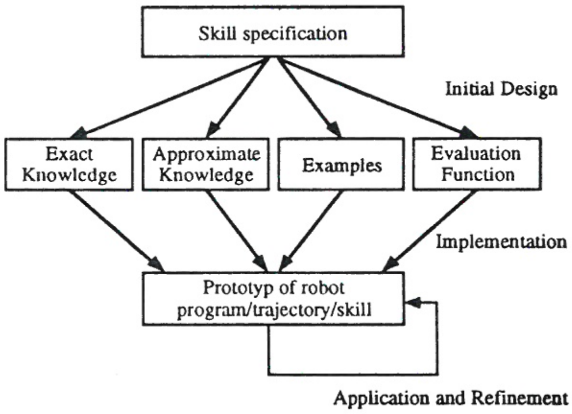

Figure 1. Approaches To the Design of Elementary Skills

To realize elementary skills requires to map robot's perceptions into actions by means of a strategy that is goal-oriented. Several possibilities to encode such a strategy exist (Figure 1). The "traditional" approach is the model-based one, which tries to a priori determine the skills application conditions and to systematically design a skill based on this identification. Since exact models are seldom feasible, usually qualitative models are employed, resulting, for example, in probabilistic approaches as in the case of occupancy grids [8,35], or in the use of fuzzy systems [28,30].

However, this explicit coding of the required strategy is not an easy task. Specifically, it cannot be assumed that users of future service robots will be able to perform this kind of low-level robot programming.

What can be expected from users of such robots is their ability to demonstrate strategies related to skills (e.g. by manually operating the robot), and to evaluate the strategy used by the robot, i.e. to evaluate robot's performance. Approaches to learning based on human-generated examples usually appear under the heading of "skill acquisition by human demonstration" and "behaviour cloning," respectively[1]. In robotics, these approaches are mostly related to manipulation skills [2,12,18]. The acquisition of basic mobility skills from human demonstrations has, for instance, been described by Pomerleau [27] and Reignier [29].

If only an evaluation of robot's performance is available, learning from examples is replaced by learning from reward and punishment (reinforcement learning, [4,31]). This kind of learning has already been successfully applied to the control of mobile robots [21,22,32]. However, a substantial amount of knowledge was always integrated into the actual reinforcement function or into the strategy used for exploration. Say, in successful applications, learning did not take place in a simple trial-and-error manner that could easily be supervised even by an unexperienced user. Instead, some kind of a hint was always required.

In this paper, a multistrategy approach to the realization of basic mobility skills is presented. The several steps involved in this approach are motivated by two observations:

- Human-generated examples alone are not enough to allow a robot to learn a skill sufficiently well [14,16].

- For skill refinement, only a scalar evaluation of robot's performance can be expected, such that skill refinement becomes a reinforcement learning task. Reinforcement learning, however, can only be successfully applied in the real world if the learning system is provided with initial hints at a suitable strategy. These hints can be obtained from user demonstrations [15].

Consequently, a human demonstration is used to initially build the skill and set up an on-line adaptation and extension mechanism. The adaptation itself is then performed using this initial information plus the feedback from the user.

Based on previous work [6,14], the following sections will focus mainly on the differences between the acquisition of mobility and the manipulation skills. Especially w.r.t. several learning techniques, the descriptions will be rather brief, however, references to the original publications will be given. The testbed for the work described here has been the mobile robot PRIAMOS and the corresponding simulation and control environment MARS [7].

2. Acquiring Elementary Skills from Human Demonstrations

"Skill" denotes the learned power of doing a thing competently. From a system's theoretic viewpoint, this means that for a given state $\mathbf{x}(t)$ the *skilled* system (the robot) should perform an action $\mathbf{u}(t)$ in order to achieve a goal that is associated with the particular skill. The action performed should be the result of a competent decision, i.e. it should be optimal with respect to an evaluation criterion (a reward) $r(\mathbf{x}(t))$ that is related to the goal to be achieved. Essentially, a skill **S** is therefore given through a control function

$$C_S : \mathbf{u}(t) = C_S(\mathbf{x}(t))$$

that implicitly encodes the goal associated with the skill and produces in each state $\mathbf{x}(t)$ a "competent" action $\mathbf{u}(t)$ and a function $r_S(\mathbf{x}(t))$ that evaluates the state $\mathbf{x}(t)$ w.r.t. the goal. To be able to apply a skill both safely and efficiently, a termination criterion $t_S(\mathbf{x}(t)) \rightarrow \{0,1\}$ and an error criterion $e_S(\mathbf{x}(t)) \rightarrow \{0,1\}$ are also required. The aim of skill acquisition from human demonstration is to approximate the functions C_S, r_S, t_S, and e_S from data obtained during human performance of the skill **S** in order to facilitate skill application as shown in Figure 2.

[1] Both expressions denote the same thing. They are, however, used separately by the Robotics and Machine Learning communities, and it has turned out that both communities do not know the efforts of one another.

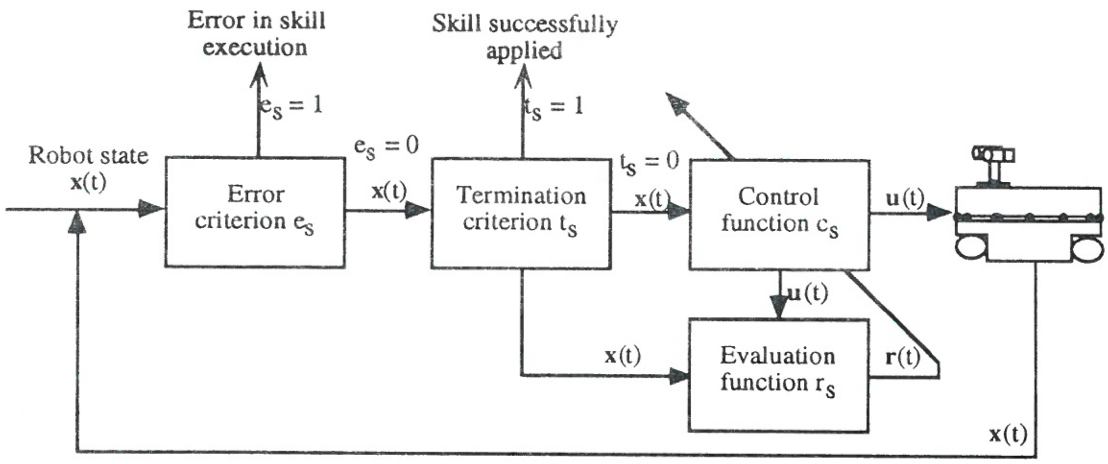

Figure 2. Skill Application

In the case of elementary or basic skills, the state **x** is given as a sequence of sensorial inputs **y**, i.e. $\mathbf{x}(t) = (\mathbf{y}(t-d), ..., \mathbf{y}(t-d-p))$, d, p >= 0, and the result of a human demonstration is a sequence $((\mathbf{y}(0), \mathbf{u}(0)), ..., (\mathbf{y}(T), \mathbf{u}(T)))$ of sensor measurements $\mathbf{y}(t)$ and actions $\mathbf{u}(t)$. For data originating from human demonstrations, $d = t_r$ is the reaction time of the operator during execution, and p is the minimum perceptual history that must be known in order to distinguish states which require different actions. While t_r can be estimated in advance, p is determined based on the demonstration data. If, for example, elementary mobility skills are being demonstrated, the $(\mathbf{y}(t), \mathbf{u}(t))$ pairs are given as distances measured by PRIAMOS' ultrasonic sensors and translational/rotational offsets, respectively. Due to the low sample frequency of 1 [Hz], $d = t_r = 1$ is chosen in this case.

3. What Is Special About Basic Mobility Skills?

During the demonstration of manipulation skills, usually a force/torque sensor provides robot's perceptions [14]. Such a sensor is able to deliver information at a rate of at least 30[Hz]; such that a demonstration of 20 seconds length results in more than 600 single training samples. Moreover, such a sensor does in general not suffer from drop outs, i.e. while the sensor readings may be noisy, individual perception components will not be missing completely. For preprocessing such examples, statistic means are therefore often valid [6,14].

In case of a mobile robot, the situation is different. First, the distance measuring sensors in use do in general not allow for scanning the environment at a high rate. Even with laser range finders, a complete 360^{o} scan of robot's surroundings usually requires more than 50[ms]: If ultrasound sensors are being used, the achievable scan frequency is seldom higher than 2[Hz]; resulting in about 50-100 samples per demonstration only. Beside the limitation of the maximum scan rate, distance measuring sensors have other characteristics that must be considered. They provide valid information only if objects reflecting the ultrasound waves or the laser beam are suitably oriented with respect to the robot. Therefore, even if an object is close to the sensor, it may happen that it cannot be detected. This results in steps in the sensor input. Also, the distance measured by the sensor does not match the real environmental situation at any instant.

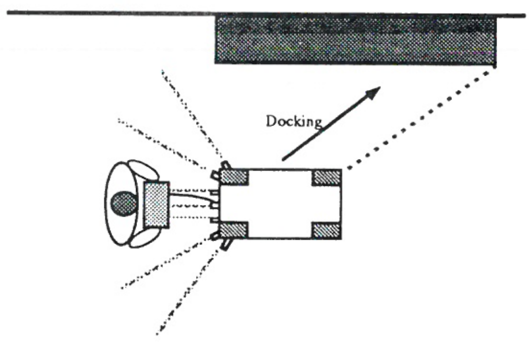

Figure 3. The Influence of the Operator's Presence on the Recorded Data

Finally, in case of mobile robots often the presence of the operator influences the sampled data (Figure 2). Since the sensors being used scan the area around the robot, they will usually also perceive the operator giving the demonstration. If this is the case, the impaired sensors' measurements must be removed in advance to any kind of data analysis and learning.

4. Generating Training Data

The data obtained from a human demonstration of a skill to be performed by PRIAMOS consist of 24 ultrasonic distance measurements, two translational offsets **Dx** and **Dy**, and a rotational offset **D**α sampled at any instant. In the ideal case, these data represent the skill to be acquired perfectly. In reality, however, this will seldom be the case. Several sources of suboptimality exist that result in disturbances, the most prominent being [16]:

1. the existence of incorrect actions that must be corrected at a later instance and
2. the human tendency to perform "bang-bang" instead of smooth control.

The effect of these suboptimalities cannot be neglected. Consequently, the robot should not simply copy the human when performing the demonstrated skill. Instead, it should avoid those mistakes that are obvious from the very beginning, and overcome other deficiencies later, i.e. through adaptation of the acquired skill. Additionally, only the information that is really relevant to realizing the skill should be employed. In particular, it is therefore necessary

1. to identify relevant action components (i.e. relevant degrees of freedom),
2. to identify relevant perception components (i.e. relevant sensors), and
3. to identify and remove actions that are obviously not optimal.

4.1 Identification of Relevant Actions

To rank the importance of a particular action component for the given skill, the contribution of this component to executing the skill is determined. If $\|\mathbf{u_i}(t)\|$ is the normed[2] contribution (e.g. the change in position w.r.t. a particular degree of freedom) of action component **i** at time t, the individual contribution of this component is

$$K_i = \frac{\sum_{t=0}^{T} \|u_i(t)\|}{\sum_{j=1}^{\dim(u)} \sum_{t=0}^{T} \|u_j(t)\|}$$

where T denotes the amount of samples taken during the demonstration, i.e. the length of the demonstration. The set **U** of relevant action components can be determined as the minimum subset of components of **u** whose combined contribution is above a certain threshold $0 < \delta <= 1$. Usually, δ_R is chosen to be $0.9 <= \delta_R <= 1$, depending on the confidence in the efficiency of the human demonstration. For all experiments described in Section 7 (see results in Section 7).

4.2 Identification of Relevant Perceptions

In case of manipulation skills that require compliant motion, often a direct and constant dependency between the change in perceptions (e.g. forces and torques) and the commanded actions exists. Based on knowledge of relevant actions, such a dependency can be used to identify relevant perceptions [6], if sufficiently many samples have been taken during the demonstration.

This is however not the case if mobility skills are being demonstrated that use PRIAMOS' ultrasonic sensors (see Section 3). Due to the limited amount of samples, it is also not possible to statistically determine the reaction time of the operator, the dead-time of the system, or the perceptual history required to generate a "competent" action. While the system dead-time can be found by other means, e.g. by an analysis of robot's control system, the remaining identification tasks have to be solved by means of heuristics.

[2] The norm || || must take the different nature of the individual degrees of freedom into account. For PRIAMOS, it is defined as $\|x\| = |x|$ for the translational degrees of freedom, and $\|x\| = |x|/2$ [deg] for the rotational ones.
aa

1. Let ((**y**(0); **u**(0)),...,(**y**(T), **u**(T))) be a sequence of sensor measurements **y**(t) and actions **u**(t) resulting from a demonstration.
2. Define valid(**y**(t)) := 0:15m y(t) < 5m; i.e. a sensor i is valid at time t iff the measured distance **y**(t) is within the sensor-specific validity bounds. The given limits are for PRIAMOS' ultrasonic sensors.
3. Let MIN_S be the minimum number of sensors that should be valid at any instant t.
4. Let **A** be the set of **allowed** sensors, i.e. those sensors that have not been impaired by the operator's presence.
5. Let **R** be the set of sensors that are valid and required to identify the goal situation, i.e. **R** = {i: 1<= i <= 24 and valid(y_i (T))}
6. Let **R** := **R** ∩ **A**
7. Let **Y** := ∅ be the set of unresolved samples.
8. For any instant t
   (a) Let **V**(t) be the set of valid sensors at time t.
   (b) Let **V**(t) := **V**(t) ∩ **R**
   (c) If |**V**(t)| < MIN_S
      let **Y** := **Y** ∪ {(**y**(t), **u**(t))}
9. If **Y** = ∅ terminate.
10. Let j ε **R** ∩**A** be the sensor that is valid for most samples in **Y**. j must also be valid for at least one sample.
11. If no such sensor exists, terminate.
12. Let **R** := **R** ∪ {j}, let **A** := **A** - {j}.
13. Proceed with 7.

Figure 4. Selection of Relevant Sensors

The procedure shown in Figure 4 selects the necessary sensors by starting from those sensors that are required to define the termination criterion t_S (Figure 4, Step 5). These sensors may be selected by the operator, or they are chosen to maximize the distance between the terminal situation and other situations that occur in the course of the demonstration.

Then, for each situation, it is checked if sufficiently many sensors (i.e. more than or equal to MIN_S sensors) are measuring valid values (Step 8). If necessary, new sensors are added (Steps 10 and 12). If the sensor selection procedure does not result in a sufficiently large set of sensors, it is necessary to reduce the number of required sensors. If not possible, a new demonstration must be made.

Following the sensor selection procedure, the perception vector **y** is reduced to the necessary sensors, such that the following training data for the functions C_S, t_S, and e_S are finally available:

1. a sequence ((**y**(0), **u**(0)), ..., (**y**(T), **u**(T))) of (possibly reduced) perception vectors **y**(t) and action vectors **u**(t) that approximately represent the function C_S;

2. a hyperinterval $[[y_{m1}, y_{M1}], \ldots, [y_{mn}, y_{Mn}]]$ with $y_{mi} < y_{Mi}$ for 1 <= i <= n, n = dim(y) describing the allowed range of perceptions (i.e. the function e_S), and

3. a sequence of perceptions representing non-terminal states (**y**(0), ..., **y**(T-1)) and the goal state **y**(T) that serve for building the function ts.

4.3 Removing Incorrect Actions

Since we cannot assume to have any knowledge about the skill to be learned apart from the given examples, altering the sampled action vectors in order to possibly achieve better performance does not involve any global optimization technique. We can only assume that the amount ||**u**|| of an action is proportional to its effect and that the subsequent application of two actions $\mathbf{u}_1$ and $\mathbf{u}_2$ with $\mathbf{u}_1 = \alpha \, ||\mathbf{u}_2||$, is equivalent to applying $(1 + \alpha)\mathbf{u}_2$. Then, the following preprocessing steps can safely be executed:

- Removal of all actions that do not contribute at all to solving the task, i.e. removal of all samples (**y**, **u**) with $||\mathbf{u}|| <= \delta_R$, $\delta_R > 0$.

- Smoothing of all corrective motions, i.e. if $\mathbf{u}(t) \approx \alpha\mathbf{u}(t+1)$[3] and $\alpha < 0$,

  set $\mathbf{u}(t) = \mathbf{u}(t+1) = 1/2(\mathbf{u}(t) + \mathbf{u}(t+1))$.

4.4 Initializing the Evaluation Criterion

For building the function r_S, a sequence (((**y**(0), **u**(0)), r(0)), ..., ((**y**(T), **u**(T)), r(T))) of ((perception, action), reward) triples is required. Since the optimal skill is not known in advance,

[3]In general, no strict equality u(t) = αu (t+1) will be achieved. Therefore, we consider u(t) = αu(t+1) iff 1 <= ι <= dim(u): $u_i(t) = \alpha_i u_i(t+1)$, $\alpha-\varepsilon <= \alpha_i <= \alpha+\varepsilon$.

determining r(t) can only be based on heuristics. If it is assumed that demonstration data ((**y**(0), **u**(0)), ...,(**y**(T), **u**(T))) are given and the goal state is represented by y(T), $r = r(t) = r_S(y(t))$ can be initialized as

$$r = \begin{cases} r + if \|y(T) - y(t)\| < \|y(T) - y(t-1)\| \\ r - else \end{cases}$$

Thus, any state that is closer to the goal state than its predecessor is given a positive reward r_+, while any state that is further way is given a negative reward r-. Similar to the initial control function C_S, the initially learned evaluation criterion r_S will only be an indication of what the optimal evaluation criterion looks like. Therefore, r_+ and r- are not set to the extrema of the range of possible evaluations. E.g. if the reward obtained after skill execution will be in [-1, 1], $r_+ = 0.7$ and $r_- = 0.2$ are reasonable choices.

5. Off-line Skill Learning

During off-line learning, the functions C_S, r_S, t_S and e_S are built from the generated training data. To this aim, the formalism to be used for representing these functions has to be selected first. This is in general done ad hoc, i.e. the representation is chosen to be suitable for learning a specific skill under specific conditions [1,12,27]. However, an untrained user cannot be expected to perform this selection for every skill he/she wants the robot to acquire. Therefore, it is necessary that the representation is constructed from the training data. Secondly, the chosen representation must allow for incremental learning, since on-line adaptation of the skill is mandatory.

Both requirements exclude multilayer perceptrons [20,27] or non-incremental decision-tree techniques [33] from being used in a general approach. Also, the specification of the functional form of the skill by the user, which reduces skill learning to the identification of numerical parameters [1], is not an appropriate solution. In B-Learn II, the investigation of several function approximation techniques [3,13,21,25] leads to the selection of neural networks based on local receptive fields [23], such as Radial-Basis Function Networks (RBFs) [26].

Such networks can be built from training data [3,19,23,24], which is extremely important in a setting that asks for automation of the learning phase. Additionally, these networks allow for directly assessing the knowledge that is available with respect to a particular situation. Say, by checking the activation of the individual clusters available in such a network, it is possible to easily distinguish if the network is recalling an example it has seen before, and to which extent it is generalizing.

For C_S and r_S, the clustering algorithm described in [3] is applied to generate the initial network. Afterwards, the resulting networks are trained via gradient descent.

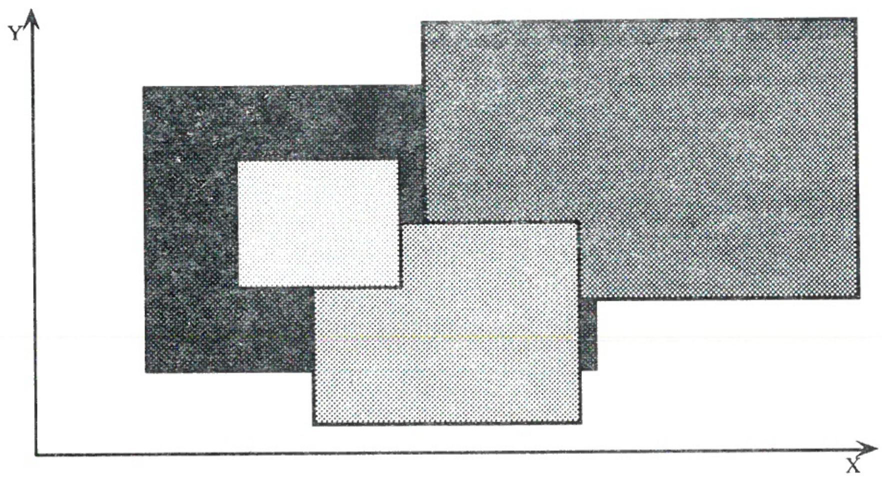

Figure 5. Two-dimensional Region List Drawn "in Reverse Order." During Evaluation, the Brighter Regions Are Considered First

The criterion functions t_S and e_S are represented as a region list, i.e., an ordered list of labelled hyperintervals ((Figure 5). Two lists originate from the demonstration data: the first initially consisting only of a hyperinterval representing e_S

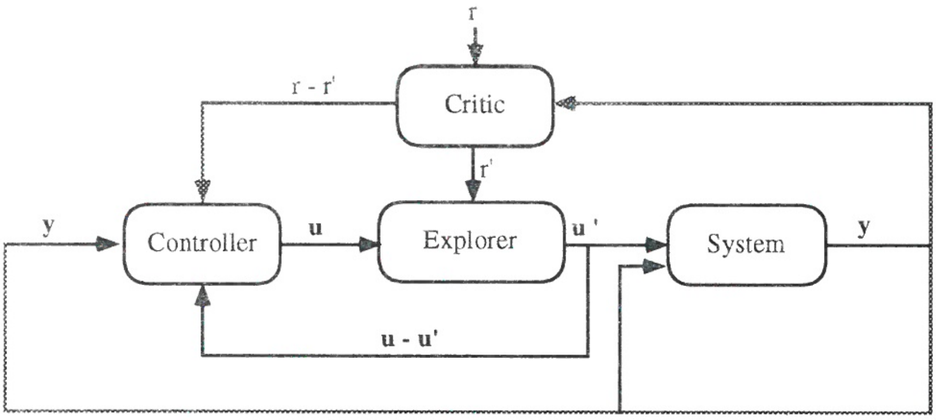

Figure 6. Direct Adaptation Scheme with Critic (Evaluation Function) and Explorer (Exploratory Elements)

that describes the encountered state space, and a second one that represents the goal state (i.e. the termination criterion t_S) by a hyperinterval, too. For building these hyperintervals, in case of the error criterion all relevant sensors are being used. Initially, all states are considered to be valid that are in the perceptual boundaries for the given skill. For building the termination criterion, we start from the final state and extend the state description in a greedy way until a sufficient distance of the final state to all other states occurring during the demonstration exists. Then, we define the termination criterion as a hypercube with the goal state as its center and its width depending on the resolution of the sensors and the desired positioning accuracy.

6. Skill Refinement

Refining a basic skill on-line means to alter the functions Cs; ts; and es with respect to some external feedback. The actual mechanism used for this adaptation depends on the representation of the functions as well as on the information contained in the feedback [13].

The minimum feedback that is assumed to be available in the context of learning from human demonstrations is an evaluation of the performance of the robot after the application of an individual skill. The task is therefore to refine a continuous real-valued function on the basis of a delayed reinforcement signal. Gullapalli's approach based on stochastic real-valued units (SRV units, [9,10]) provides a suitable starting point for solving this direct adaptation task (Figure 6), which can formally be described as follows:

Given: An initial skill represented by functions C_S, t_S, and e_S, a model r_S that acts as a critic, and an external feedback source providing a scalar evaluation r of the effect of the skill application.

Determine: New functions C^n_S, t^n_S, and e^n_S whose application results in a higher, if possible optimal evaluation, and a new model r^n_S that takes the changes introduced by the adaptation into account.

Since C_S and r_S are represented as RBF networks, the refinement of both functions comprises the following steps[4]:

Extension of C_S towards new situations: The local representation employed in RBF networks allows for detecting situations that have not been encountered so far. If the activation r_i of all neurons i representing a cluster is below a threshold r_{min}, i.e. if

$$\forall i \in \{1,\ldots,n\}: r_i(x) < r_{\min}$$

it can be concluded that **x** is a new situation. If it is desired to extend the network to cover this situation, a new cluster n+1 is generated. The action **u** (i.e. the weights $w_{n+1,i}$) to be associated with this cluster can be requested from the user or cloned from the action associated with

[4]Explanations are given for C_S. For r_S, actions u should be replaced by rewards r.

the cluster that is nearest to the new one. In any case, the width σ of the new cluster is to be initialized such that it does not affect the already existing clusters. If $1 <= k <= n$ is the index of the nearest existing cluster, and δ is a threshold that controls the maximum overlap between clusters, σ is initialized as $\sigma = r_k^{-1}(x,\delta)$. In case of Gaussian transfer functions r_i, this means that

$$\sigma = \frac{\|x - x_k\|}{\sqrt{-\ln \delta}}$$

Adaptation of a known action u: The typical action to be untertaken in skill refinement is the adaptation of the action **u** calculated by evaluating the network representing C_S in the given situation **x**. Assuming that the new action $\mathbf{u}_n$ obtained a feedback signal r_n, whereas the original action **u** resulted in an evaluation r with $r < r_n$, the network representing C_S is adapted on-line by minimizing the error $(u_j - u_{nj})$ and using $\eta = \eta_0 \operatorname{sgn}(r_n - r)$ as learning rate.

The only kind of information that can be expected from the user during the skill application and refinement process is an evaluation after the termination of the skill execution[5]. To perform the actual adaptation, we use the following rules:

1. If the user aborted the operation, he/she is asked if the current state is a goal state or an error state. The corresponding function (i.e. e_S resp. t_S) is updated.

2. Otherwise, the user is asked if the criterion that fired did so correctly. If this is the case, the control function C_S is updated, otherwise, the criterion is updated. In the latter case, an additional feedback can be given by the user that can be used for updating the control function C_S.

3. If the control function is updated, the critic r_S is updated, too.

[5]Termination means that either a robot-specific error occurs, the goal state known from the demonstration is reached, or the user stops the robot.

If the control function must be updated, the contribution of any particular action **u**(t) to the obtained evaluation must be determined. To solve this temporal credit assignment problem, often exponentially discounted rewards are used. However, our approach has been to use an exponentially discounted adaptation rate.

For updating the region list, we simply include or exclude the final state **x**: If a new goal or error state must be learned, a corresponding region is inserted. Otherwise, a new unlabelled region covering the current state is inserted, thereby excluding a part of the old region from the error or the termination criterion.

7. Experiments

For the evaluation of the described method, several skills were taught to PRIAMOS (for a thorougher evaluation, see [5]). The skills we use for demonstration purposes are the sensor-based motion around a corner (Figure 7) and docking (Figure 9).

In the first experiment, the skill to be learned was the reactive motion around a corner (Figure 7).

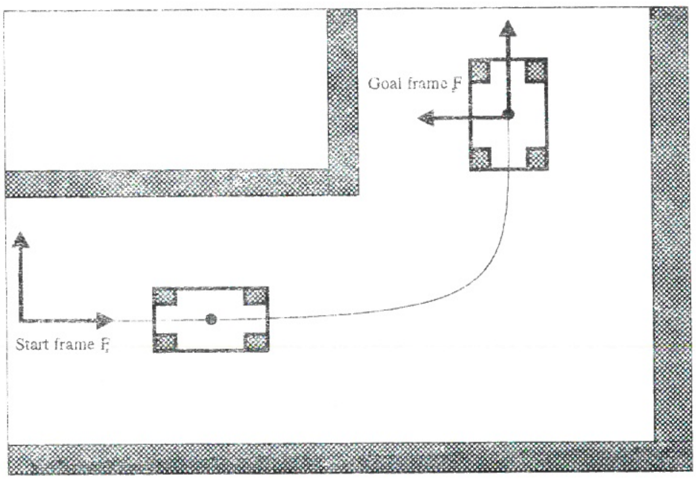

Figure 7. Motion Around A Corner

This skill was demonstrated ten times, resulting in example files containing between 40 and 52 samples each. The sampled data were analyzed and preprocessed, using $MIN_S = 4$ and $\delta_R = 0.05$.

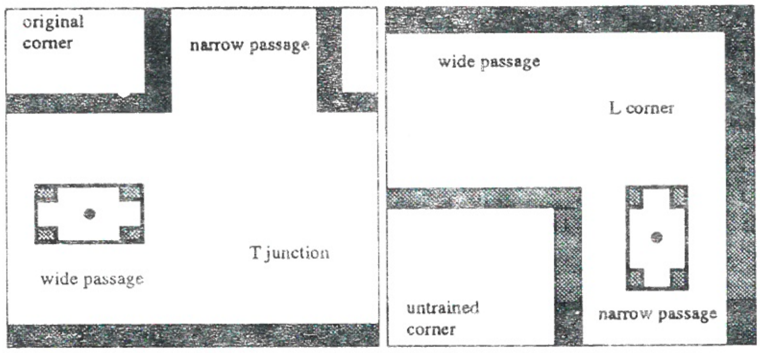

Figure 8. T-shaped Corner Used for Teaching (left); L-shaped Corner Used for Tests (right)

Using this parameterization, 39 to 49 samples remained in each example file (resulting in a total of 449 samples) Following that, RBF networks were constructed to represent the functions C_S and r_S, resulting in a network of 224 clusters for C_S.

Table 1. Skill application and refinement (test 1): Neurons inserted in the networks representing C_S and r_S and new termination condition t_S generated during 4 subsequent skill applications in known environment, but starting from different locations. The goal was reached in any of the trials.

Function	Trial 1	Trial 2	Trial 3	Trial 4
C_S	9	4	1	0
r_S	9	4	1	0
t_S	1	0	0	0
Success	0	1	1	1

Table 2. Skill refinement (test 2): Neurons inserted in the networks representing Cs and r_S and new termination condition t_S generated during 4 subsequent skill applications in an unknown environment. The goal was reached in any of the trials.

Function	Trial 1	Trial 2	Trial 3	Trial 4
C_S	12	4	0	0
r_S	12	4	0	0
t_S	1	1	0	0
Success	0	1	1	1

To evaluate the learned skill, it has been applied in the refinement of two different environments. Table 1 shows the results for the training environment (a T-shaped corner), Table 2 gives the results for the L-shaped test environment. Obviously, a certain area of the input space was not covered by the initial demonstrations, such that new neurons had to be inserted during the first trials. After the third trial, however, the input space seemed to be covered sufficiently.

A second skill to be demonstrated, learned, and refined was the docking skill (Figure 9). Here, two demonstrations using the real robot resulted in 75 and 84 samples, respectively. By eliminating irrelevant actions with $\delta_R = 0.05$, and combining the two demonstration files, training files containing 122 samples for both C_S and r_S were obtained. With $MIN_S = 4$, 14 sensors (out of 24) were chosen to be potentially relevant. From these data, the clustering algorithm generated networks containing 89 (for C_S) and 35 (for r_S) regions.

Table 3. Skill application and refinement (test 3): Neurons inserted in the networks representing Cs and rs and new termination condition ts generated during 4 subsequent skill applications in known environment, but starting from different locations. The goal was reached in any of the trials.

Function	Trial 1	Trial 2	Trial 3	Trial 4
C_S	11	3	0	0
r_S	13	4	0	0
t_S	1	0	0	0
Success	0	1	1	1

Table 3 shows the results obtained from applying the initially learned skill. Most notable is the relatively large number of neurons inserted during the first trial. At the end of this trial, the termination condition was not correctly identified, such that the termination criterion had to be extended as well. This extension, however, proved to be sufficient for all subsequent trials.

With respect to the applicability of the presented programming technique, we believe that the assumptions we have made, i.e. to rely on the ability of a human teacher to demonstrate a solution to a given task, and to provide at least a qualitatively correct evaluation of robot's performance, are realistic. Obviously, we cannot expect that the action and perception skills acquired via an interactive learning approach are comparable to those originating from an in-depth task analysis and explicit robot programming. However, especially if robots are to become consumer goods, they will be exposed to users who are not at all familiar with computers or robots. For such users, explicitly programming their robot according to their personal requirements is not an option, whereas teaching by showing, i.e. Robot Programming by Demonstration, definitely is.

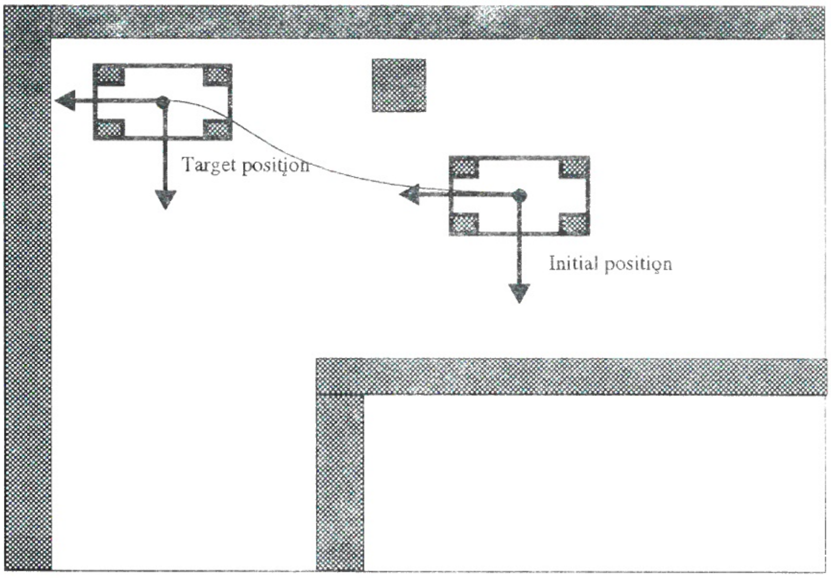

Figure 9. Docking

8. Conclusions

Throughout this paper, an approach to acquiring basic mobility skills from human demonstrations has been presented. This approach takes the explicit constraints w.r.t. real mobile robots and their sensor systems as well as the need for example preprocessing and skill refinement into account. The skill refinement procedure that was presented worked well in the presented examples. This effect is certainly due to the fact that the initial examples provided by the user were already good evidence of the strategy to be learned. In case of really bad teacher performance, a refinement technique relying only on a scalar feedback to be provided at termination time will not be sufficient.

If, however, good teacher performance can be expected, we consider the main limitations of the technique to derive from the ultrasonic sensors, whereas the tools and methods in current use seem to be quite stable. Consequently, our future work will concentrate on the use and integration of additional sensor systems, especially a laser scanner.

Acknowledgment

This work has been funded by the ESPRIT Project 7274 "B-Learn II". It has been carried out at the Institute for Real-Time Computer Systems & Robotics, Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, University of Karlsruhe, Germany.

REFERENCES

1. ASADA, H. and YANG, B.-H., . **Skill Acquisition from Human Experts Through Pattern Processing of Teaching Data**, Proceedings of the IEEE International Conference on Robotics and Automation, 1989.

2. ASADA, H. and LIU, S., **Transfer of Human Skills To Neural Net Robot Controllers**, Proceedings of the IEEE International Conference on Robotics and Automation, 1991.

3. BAROGLIO, C., GIORDANA, A., KAISER, M. NUTTIN, M. AND PIOLA,. R., **Learning Controllers for Industrial Robots,** MACHINE LEARNING, 1996.

4. BARTO, A. G., SUTTON, R.S. AND ANDERSON. C.W., **Neuron-like Elements That Can Solve Difficult Learning Control Problems,** IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, 1983.

5. DECK, M., **Interaktive Akquisition von Elementarfähigkeiten für mobile Roboter**. Master's Thesis. Universität Karlsruhe, Fakultät für Informatik, Institut für Prozeßrechentechnik und Robotik, 1995.

6. DILLMANN, R., KAISER, M. and UDE, A.,. **Acquisition of Elementary Robot Skills from Human Demonstration**, International Symposium on Intelligent Robotics Systems, Pisa, Italy, 1995.

7. DILLMANN, R., KAISER, M., WALLNER, F. and WECKESSER, P., **PRIAMOS: An Advanced Mobile System for Service**, Inspection, and Surveillance Tasks, in T.Kanade, H. Bunke and H. Noltemeier (Eds.) Modelling and Planning for Sensor Based Intelligent Robot Systems , WORLD SCIENTIFIC, 1995.

8. ELFES, A., **Using Occupancy Grids for Mobile Robot Perception and Navigation**. IEEE COMPUTER, 1989.

9. GULLAPALLI, V., **A Stochastic Reinforcement Learning Algorithm for Learning Real- Valued Functions,** NEURAL NETWORKS 3, 1990.

10. GULLAPALLI, V., FRANKLIN, J.A. and BENBRAHIM, H., **Acquiring Robot Skills via Reinforcement Learning,** IEEE CONTROL SYSTEMS MAGAZINE 14(1), 1994.

11. HARNAD, S., **The Symbol Grounding Problem**, PHYSICA D 42, 1990.

12. HIRAI, S., NOGUCHI, H. and IWATA, K., **Transplantation of Human Skillful Motion to Manipulators in Insertion of Deformable Tubes**, IEEE International Conference on Robotics and Automation, 1995.

13. KAISER, M., RETEY, A. and DILLMANN, R., **Designing Neural Networks for Adaptive Control**, IEEE International Conference on Decision and Control (34th CDC), 1995.

14. KAISER, M., RETEY, A. and DILLMANN,R., **Robot Skill Acquisition via Human Demonstration**, Proceedings of the International Conference on Advanced Robotics, 1995.

15. KAISER, M. and DILLMANN, R., **Hierarchical Learning of Efficient Skill Application for Autonomous Robots**, International Symposium on Intelligent Robotics Systems, Pisa, Italy, 1995.

16. KAISER, M., FRIEDRICH, H. and DILLMANN, R., **Obtaining Good Performance from A Bad Teacher**, International Conference on Machine Learning, Workshop on Programming by Demonstration. Tahoe City, CA, USA, 1995.

17. KAISER, M., KLINGSPOR, V., MILLAN, J. del R., ACCAME, M., WALLNER, F. and DILLMANN, R., **Using Machine Learning Techniques in Real-world Mobile Robots**, IEEE EXPERT, April 1995.

18. KOEPPE, R. and HIRZINGER, G., **Learning Compliant Motions by Task-demonstration in Virtual Environments**, 4th Int. Symp. on Experimental Robotics, 1995.

19. KWOK, T.-Y. and Yeung, D.-Y., **Constructive Feedforward Neural Networks for Regression Problems: A Survey**. Technical Report HKUST-CS95-43. Hong Kong University of Science and Technology, Department of Computer Science, 1995.

20. LIU, S. and ASADA, H. , **Teaching and Learning of Deburring Robots Using Neural Networks,** Proceedings of the IEEE International Conference on Robotics and Automation, 1993.

21. MILLAN, J. del R., **Rapid, Safe, and Incremental Learning of Navigation Strategies,** IEEE TRANSACTIONS ON

SYSTEMS, MAN, AND CYBERNETICS, 1996.

22. MILLAN, J. del R. and TORRAS, C., A **Reinforcement Connectionist Approach to Robot Path Finding in Non-maze-like Environments**, MACHINE LEARNING, 1992.

23. MOODY, J. and DARKEN, C. , **Learning with Localized Receptive Fields**, in, T. Sejnowski, D. Touretzky and G. Hinton (Eds.) Proceedings of the Connectionist Models Summer School , Carnegie Mellon University, 1988.

24. MUSAVI, M.T., AHMED, W. , CHAN, K.H. , FARIS, K.B. and HUMMELS, D.M., **On the Training of Radial Basis Function Classifiers**, NEURAL NETWORKS 5, 1992.

25. NUTTIN, M., GIORDANA, A., KAISER, M. and SUAREZ, R. ,. **B-Learn II - D 203. B-Learn II - ESPRIT BRA** Project No. 7274, 1994.

26. POGGIO, T. and GIROSI, F. , **Networks for Approximation and Learning,** Proceedings of the IEEE 78(9), 1990.

27. POMERLEAU, D. A., **Efficient Training of Artificial Neural Networks for Autonomous Navigation**, NEURAL COMPUTATION 3, 1991.

28. REIGNIER, P., **Fuzzy Logic Techniques for Mobile Robot Obstacle Avoidance**, International Symposium on Intelligent Robotic Systems, Zakopane, Poland, 1993.

29. REIGNIER, P., HANSEN, V. and CROWLEY, J.L., **Incremental Supervised Learning for Mobile Robot Reactive Control**, INTELLIGENT AUTONOMOUS SYSTEMS 4, 1995.

30. SONG, K.-T. and TAI, J.-C., **Fuzzy Navigation of A Mobile Robot**, Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems, 1992.

31. THORNDIKE, E. L., **Animal Intelligence**, MACMILLAN, New York, 1911.

32. THRUN, S. B., COLUMBUS: **An Autonomously Exploring Mobile Robot**, ADVANCES IN NEURAL INFORMATIONPROCESSING SYSTEMS 5, 1992.

33. URBANCIC, T. and BRATKO, I. , **Reconstructing Human Skill with Machine Learning**, Proceedings of the European Conference on Artificial Intelligence, 1994.

34. WALLNER, F. and DILLMANN, R., **Efficient Mapping of Dynamic Environment by Use of Sonar and Active Stereo-vision**, International Symposium on Intelligent Robotic Systems `94, 1994.

35. WALLNER, F., LÜTH. C. and LANGINIEUX, F., **Fast Local Path Planning for A Local Robot.**, Proc. of the Second International Conference on Automation, Robotics, and Computer Vision, 1992.