In: International Symposium on Intelligent Robotic Systems (SIRS '95), Pisa, Italy

Acquisition of Elementary Robot Skills from Human Demonstration*

R. Dillmann, M. Kaiser, A. Ude

University of Karlsruhe, Institute for Real-Time Computer Systems & Robotics, 76128 Karlsruhe, Germany
Phone: +49 721 6084051 – Fax: +49 721 606740 – E-Mail: {dillmann|kaiser|ude}@ira.uka.de

Abstract. Similarly to human voluntary movements, a robot's elementary skills can coarsely be divided into two main categories: rapid transfer movements that are executed in an open loop with no opportunity for correction and slower controlled movements guided by sensorial feedback [6]. In this paper we consider two particular problems which arise when acquiring open-loop and closed-loop elementary skills: the segmentation of the trajectory into time intervals corresponding to different classes of the operator's motion and the preprocessing of the possibly bad examples that were recorded during a demonstration in order to make them suitable for skill learning. The proposed methods are based on the fuzzy set theory, statistics, and Shannon's information theory.

1 Introduction

The increasing availability of sophisticated sensor systems for both manipulation and mobile robots makes the development of smarter and more adaptive robots possible. Such high-performance systems are opening new markets to the robot technology, e. g. the recently emerging market for intelligent service systems. Since one of the major cost factors in robotic application is the cost of robot programming, the idea of robots becoming consumer products lets the interest in intuitive programming methods such as iconic programming and especially *Robot Programming by Human Demonstration (RPD)* [5, 10] grow rapidly.

Since humans can carry out motions with no apparent difficulty, one would expect the generation of elementary skills to be a relatively simple problem. However, it turns out that it is extremely difficult to duplicate this elementary operative intelligence, which is used by humans unconsciously, in a computer-controlled robot [11]. This fact gives rise to an idea to use human demonstrations to learn elementary skills, leading to a natural extension of the traditionally task-level oriented RPD.

2 Acquisition of robot trajectories

Let us first consider the specification of open-loop trajectories. To demonstrate a desired trajectory, the human operator simply moves the object to be manipulated or, alternatively, a specially designed teaching tool along it. We have developed a stereo vision system [20] to monitor the demonstrated motion (see Fig. 1). Motions that can be specified in this way are typically used for the navigation of a robot from one place to another. Recent research in this area comprises among others Delson and West [3], Tso and Liu [18] and Ogata and Takahashi [13].

Let $\boldsymbol{\vartheta}$ denote the demonstrated trajectory. Its functional form is of course not known in advance. The vision system returns a sequence of poses $\boldsymbol{p}_j$ measured at time instants t_j. The relation between the true trajectory and the measurements can be expressed in terms of the following regression model

$$\boldsymbol{p}_j = \boldsymbol{\vartheta}(t_j) + \varepsilon_j, j = 1, \ldots, n, \tag{1}$$

where ε_j are assumed to be zero mean, mutually independent random vectors with covariance matrices $\boldsymbol{\Gamma}_j$. The proper modelling of sensor noise is essential for the accurate trajectory reconstruction. A comprehensive discussion of this problem is given in [19].

The recorded motion usually exhibits different characteristics throughout the demonstration. For instance, its beginning and its end often consist of time intervals where no motion takes place. Such parts should be removed from the trajectory. The motion in between is normally not uniform, too; there are parts where the motion is relatively slow, such as for instance the approach and the depart phase of a pick and place

* This work is partially funded by the European Community under the project B-Learn II (Project No. 7274).

operation, and there are parts where the motion is faster. Transitions between different motion classes can have a bad effect on the reconstruction process because the type of noise caused by the human operator can significantly change when the motion type changes. To alleviate this problem, the recorded data should be segmented into components corresponding to the time intervals of relatively uniform motion.

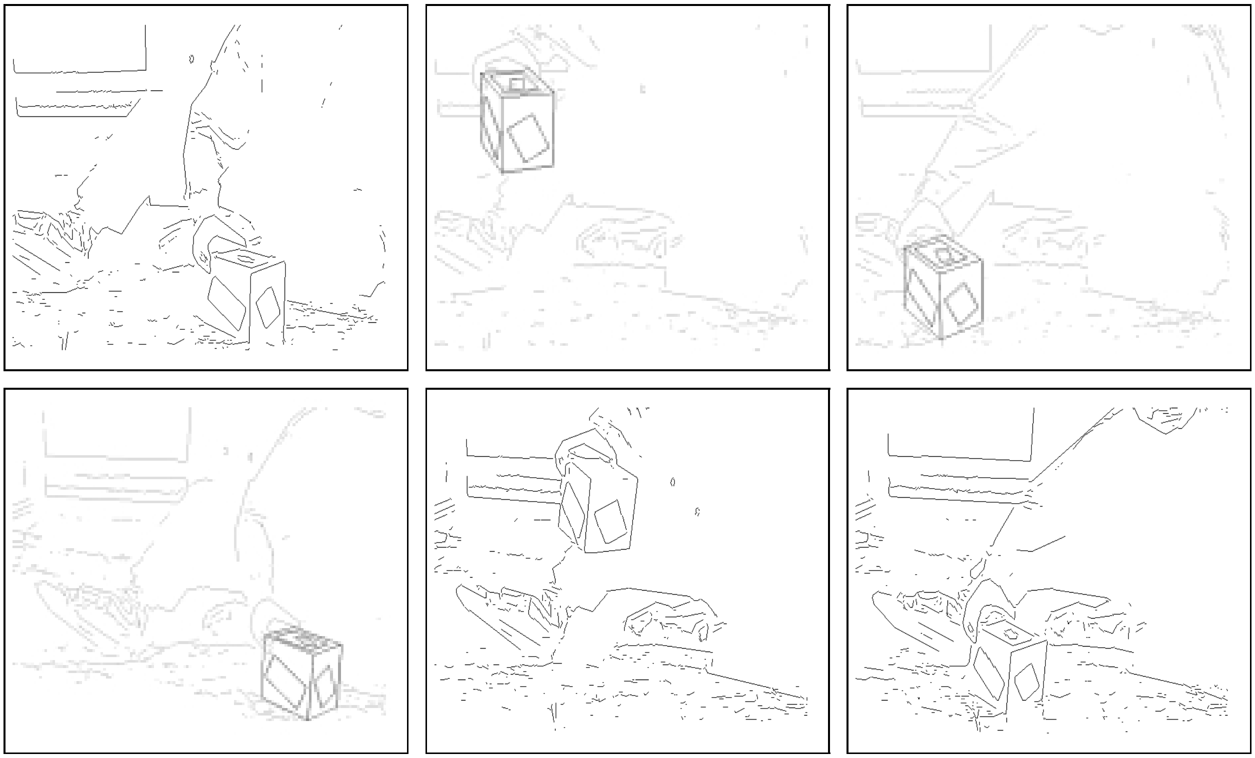

Fig. 1.: Demonstration of a trajectory: three stereo image pairs from a stereo image sequence

Let $v_j = \sqrt{\dot{x}(t_j)^2 + \dot{y}(t_j)^2 + \dot{z}(t_j)^2}$ be the linear velocity of the operator's hand at the time instant t_j. It can be calculated with the help of finite differences. The transition from quantitative attributes like v_j to qualitative abstractions can be achieved by use of the fuzzy set theory. Such an approach was employed in [9] to classify vehicle motions. One possible classification of the operator's hand motion is: zero, slow, normal, fast and very fast. We have defined trapezoidal fuzzy membership functions μ_Z, μ_S, μ_N, μ_F, μ_V with respect to the linear velocity (see Fig. 2) to characterize the above motion classes. Our definition of these functions assures that

$$\sum_I \mu_I(v_j) = 1, \;\; 1 \leq j \leq N. \tag{2}$$

Not more than two membership functions are different from zero at the given velocity. The initial choice of motion classes or, equivalently, fuzzy membership functions is very important for the success of the segmentation procedure and must be determined either experimentally or with the help of a data-driven technique.

The following procedure can be applied to segment the trajectory with respect to the predefined motion classes:

- Find the measurement in the calculated sequence of linear velocities which does not belong to any of the previously detected trajectory segments and which fulfills the following condition:

$$\mu_I(v_k) \geq \alpha_1, \;\; I \in \{Z, S, N, F, V\}. \tag{3}$$

- Detect the first l consecutive measurements to the left of the seed so that $\mu_I(v_j) < \alpha_2$, $\forall j \in \{i_1 - 1, \ldots, i_1 - l\}$, $i_1 < k$ and the first l consecutive measurements to the right of the seed so that $\mu_I(v_j) < \alpha_2$, $\forall j \in \{i_2 + 1, \ldots, \imath_2 + l\}$, $i_2 > k$. Normally: $\alpha_2 < 0.5 \leq \alpha_1$. The recorded poses between the time instants t_{i_1} and t_{i_2} generate a new trajectory segment. Its motion class is taken to be I.
- Repeat until there exist no seeds any more.

This simple procedure results in fuzzy transitions between adjacent trajectory segments. However, it is quite sensitive to noise. This can result in rapid oscillations between different motion classes, which is a very

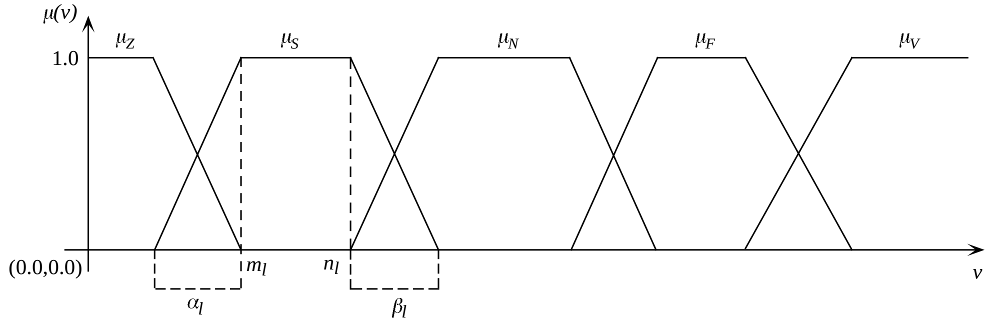

Fig. 2.: Fuzzy membership functions defining different motion classes

undesirable effect. It is not sensible to smooth trajectory segments that contain only a few measurements because every smoothing procedure needs a certain amount of measurements in order to recognize the trend in the data. Hence it is necessary to require that each detected motion class contains at least some minimal number of measurements. We enforce this by adding segments which are too short to one of the neighbouring segments until the required minimal length has been achieved. The motion class of the longer of both segments is assigned to the newly generated combined segment. The grade of membership to each trajectory segment can be calculated as follows:

- 1 if the data point belongs to only one segment.
- If the data point belongs to the transition between the two neighbouring trajectory segments of motion class I respectively J, then the grade of membership is equal to:
  - $(\mu_I(v_j),\ \mu_J(v_j))$ if $\mu_I(v_j)\mu_J(v_j) \neq 0$ (the usual case),
  - $(1,\ 0)$ if exactly one of both membership functions is equal to 0 at v_j,
  - $(0.5,\ 0.5)$ if both membership function are equal to zero at v_j.

Let m be the number of detected trajectory segments and let $\boldsymbol{U} = (\mu_{ij})$ denote the $m \times n$ matrix which coefficients are equal to the grade of membership of the j-th data point to the i-th trajectory segment. It is easy to see that $\mu_{ij} \in [0,1]$ and that $\sum_{i=1}^{m} \mu_{ij} = 1,\ 1 \leq j \leq n$. Hence $\boldsymbol{U}$ can be viewed as a fuzzy partition matrix. Let $\boldsymbol{x}_j = [t_j, v_j]^T, 1 \leq j \leq n$. $\boldsymbol{U}$ and $\boldsymbol{x}_j$ can be used as an input to one of the general fuzzy clustering algorithms. They can be employed for the refinement of the initial segmentation of the trajectory. We have obtained good results with the help of the so called fuzzy-c-means-algorithm (see Zimmermann [22]). This algorithm is based on the minimization of the criterion

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mu_{ij}^{w} \|\boldsymbol{x}_j - \boldsymbol{u}_i\|^2,\ w > 1, \tag{4}$$

over all partition matrices $\boldsymbol{U}$. The center $\boldsymbol{u}_i$ of each fuzzy cluster is defined as

$$\boldsymbol{u}_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{w} \boldsymbol{x}_j}{\sum_{j=1}^{n} \mu_{ij}^{w}} \tag{5}$$

We omit the details of the actual minimization procedure.

The above method segments the trajectory into time intervals $\{[t_k, s_k]\}$ with $t_k < t_{k+1} < s_k < s_{k+1}$.[2] We have developed a non-parametric regression technique based on smoothing vector splines to reconstruct each trajectory segment $\boldsymbol{h}_k$. It is based on finding the balance between the two conflicting goals: goodness-of-fit and smoothness. They are combined in the following criterion

$$\mathrm{F}(\boldsymbol{h}_k) = \mathrm{E}(\boldsymbol{h}_k) + \sum_{i=1}^{D} \lambda_{ki} \mathrm{J}_i(\boldsymbol{h}_k),\ \lambda_{k,i} > 0,, \tag{6}$$

where

$$\mathrm{E}(\boldsymbol{h}_k) = \frac{1}{n_k} \sum_{j=1}^{n_k} (\boldsymbol{p}_{k_j} - \boldsymbol{h}_k(t_{k_j}))^T \boldsymbol{\Gamma}_{k_j}^{-1} (\boldsymbol{p}_{k_j} - \boldsymbol{h}_k(t_{k_j})),\ \mathrm{J}_i(\boldsymbol{h}_k) = \int_{t_k}^{s_k} (h_{k,i}^{(m)}(t))^2\, dt,$$

Normally, $m = 2$ and the approximating trajectory is a twice continuously differentiable cubic vector spline. There are many ways to determine the optimal smoothing parameter $\boldsymbol{\lambda}_k$ and the optimal trajectory $\boldsymbol{h}_k$ in

[2] Actually, some more simple technicalities are needed to enforce this.

the case of scalar measurements. The most notable among them are the cross-validation and the generalized cross-validation [21]. However, it is well known that the error in the measurements provided by a stereo vision system can be adequately modelled only by use of full multidimensional probability distributions. Therefore we have extended the approximating techniques for scalar measurements to the case of vector measurements with multidimensional covariance matrices (see [19]).

In general, separately reconstructed neighbouring trajectories do not end in a common point. To assure the smooth transition between them, the neighbouring trajectories must be glued together. Let us define the following function

$$f(t) = (2t+1)(1-t)^2. \tag{7}$$

This function has some convenient properties

$$\begin{aligned} &0 \le f(t) \le 1,\ 0 \le t \le 1, \\ &f(0) = 1,\ f(1) = 0,\ f'(0) = f'(1) = 0,\ f''\left(\frac{1}{2}\right) = 0. \end{aligned} \tag{8}$$

Let $[t_k, s_k]$ and $[t_{k+1}, s_{k+1}]$ be two neighbouring time intervals with corresponding trajectories $\boldsymbol{h}_k$ and $\boldsymbol{h}_{k+1}$, respectively. The joint approximating trajectory $\boldsymbol{h}_{k,k+1}$ on the overlapping time interval $[t_{k+1}, s_k]$ can be specified as follows

$$\boldsymbol{h}_{k,k+1}(t) = \left(1 - f\left(\frac{s_k - t}{s_k - t_{k+1}}\right)\right)\boldsymbol{h}_k(t) + f\left(\frac{s_k - t}{s_k - t_{k+1}}\right)\boldsymbol{h}_{k+1}(t),\ t_{k+1} \le t \le s_k. \tag{9}$$

One can easily show that $\boldsymbol{h}_{k,k+1}(t_{k+1}) = \boldsymbol{h}_k(t_{k+1})$, $\boldsymbol{h}'_{k,k+1}(t_{k+1}) = \boldsymbol{h}'_k(t_{k+1})$ and $\boldsymbol{h}_{k,k+1}(s_k) = \boldsymbol{h}_{k+1}(s_k)$, $\boldsymbol{h}'_{k,k+1}(s_k) = \boldsymbol{h}'_{k+1}(s_k)$. Thus the combined trajectory is continuously differentiable.

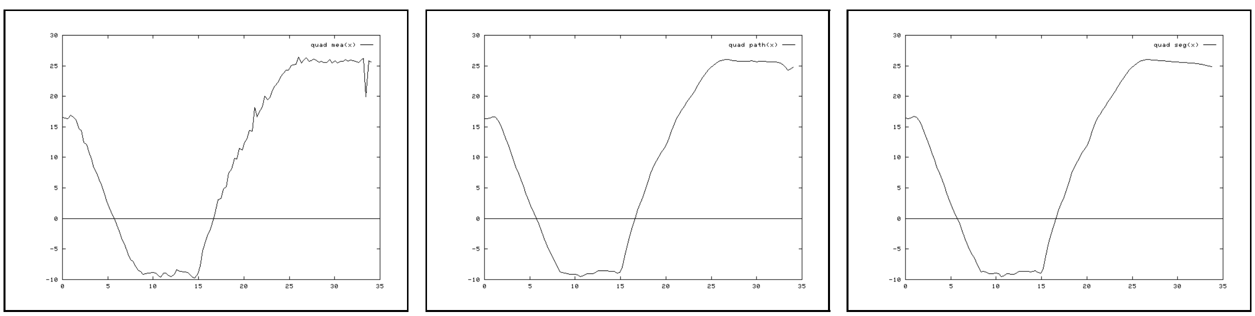

Fig. 3.: The interpolated measurements, the trajectory calculated without the previous segmentation and the trajectory calculated using the results of the segmentation procedure. The improvement in the accuracy of the reconstructed trajectory at the right boundary can be noticed. Only one component of the reconstructed 6-D trajectory is depicted in this figure.

The result of the proposed reconstruction procedure is shown in Fig 3. It significantly reduces the noise caused by both the sensor system and the human operator and is fairly resistant to outliers. If more than one demonstration is available, the presented method can be used for improving trajectories obtained from each single demonstration. The improved trajectories and the method of Delson and West [3] can be employed afterwards to combine multiple demonstrations.

3 Acquisition of elementary sensor-based skills

Elementary sensor-based skills can be defined as *operations that are realized through a direct coupling between the robot's sensors and its actuators and require a constant focus of attention during execution.* This focus of attention will in most cases be an object (such as a mating part in case of an assembly task), but it might as well be another robot (if the task is, for instance, visual tracking).
A typical example of an elementary skill is the *insertion* skill. The peg-into-hole task (Fig. 4) for which it is required is a typical task for an assembly robot. This task has also been widely used as a benchmark.
Essentially, such a skill s is given through a **control function**

$$C : \boldsymbol{u}(t) = C(\boldsymbol{y}(t-d), \ldots, \boldsymbol{y}(t-d-p)). \tag{10}$$

with $\boldsymbol{u}$ being the control output and $\boldsymbol{y}$ being the sensory input. Given a sufficient amount of examples $((\boldsymbol{y}(t-d), \ldots, \boldsymbol{y}(t-d-p)), \boldsymbol{u}(t))$, C can be approximated through, e.g., a neural network, a set of fuzzy

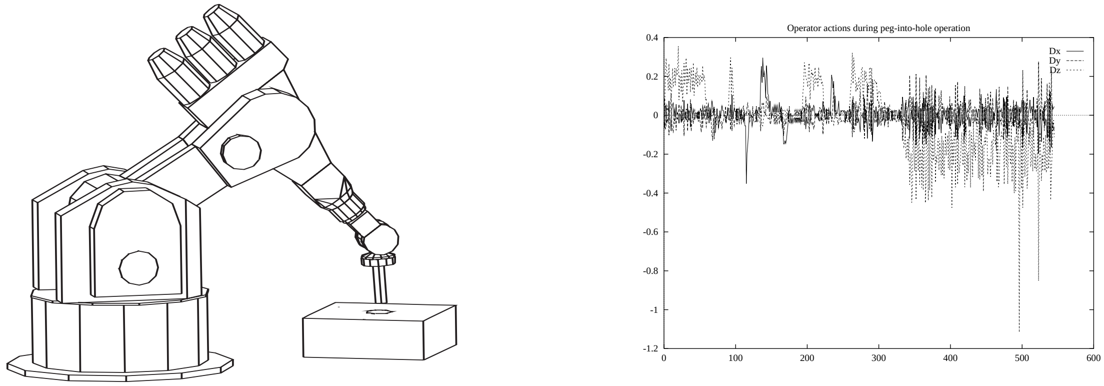

Fig. 4.: Left: The peg-into-hole task is a typical example of an operation requiring continuous control. Right: Actions (D_x, D_y, D_z) (commanded translation offsets) recorded from a human demonstration of the peg-into-hole task, including the extraction of the peg from the hole.

rules, or a regression tree [2]. These techniques work well if the presented examples are "good," i.e., not contradictory, goal-oriented and sufficiently distributed over the input space.
However, this is in general not the case, if the samples $(\boldsymbol{F}(t), \boldsymbol{M}(t), \boldsymbol{D}(t), \boldsymbol{R}(t)), t \in \{0, \ldots, T\}$, i.e., the measured forces and torques and the commanded translational and rotational offsets are recorded from a human demonstration. The noisy translation offsets sampled from a demonstration and shown in Fig. 4 can be considered the rule rather than the exception. This suboptimality of sampled data with respect to the task to approximate the function C originates from different sources [7], the most prominent being

1. *unnecessary actions* that do not contribute to achieving the final goal,
2. *incorrect actions* that require corrective actions at a later point in time, and
3. *unmotivated actions* that are in no detectable way linked to the sensorial input.

In this scenario, the usual model of the skill acquisition process comprising the phases of example generation, "strategy extraction" (learning), and skill application [1, 15, 17, 16] must be extended. In particular, the steps of **example preprocessing** aiming at **training data generation** as well as the need for **on-line skill enhancement** must be explicitely considered (Fig. 5).

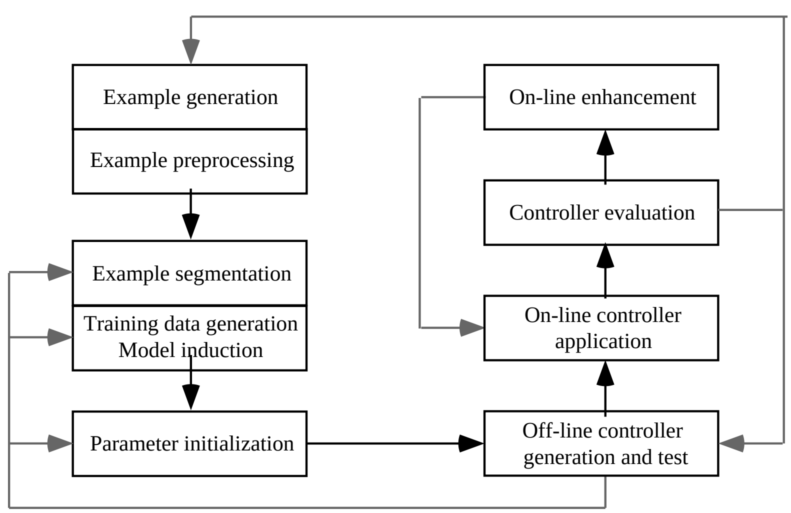

Fig. 5.: Different phases of the skill acquisition process. Gray arrows mark feedback loops.

3.1 Preprocessing example data

The main task during the **preprocessing** phase is to enhance the quality of the recorded demonstration in order to make the recorded raw samples suitable for a more thorough analysis needed for generating the training data afterwards. At this step, no specific knowledge about the task or the robot is involved. We only assume that the amount $||\boldsymbol{u}||$ of an action is proportional to its effect and that the subsequent application of two actions $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ with $\boldsymbol{u_1} = \alpha \boldsymbol{u_2}, \alpha \in \Re$ is equivalent to applying $(1+\alpha)\boldsymbol{u_2}$. This assumption is reasonable in the context of both robot manipulators and mobile robots, such that the individual preprocessing steps that can be performed are the following:

- Removal of all actions that do not contribute anything to solving the task, i.e., removal of all samples $(\boldsymbol{y}, \boldsymbol{u})$ with $||\boldsymbol{u}|| \leq \delta, \delta \geq 0$.

- Smoothing of all corrective motions, i.e., if $\boldsymbol{u}(t) \approx \alpha \boldsymbol{u}(t+1)^3$ and $\alpha < 0$, set $\boldsymbol{u}(t) = \boldsymbol{u}(t+1) = \frac{1}{2}(\boldsymbol{u}(t) + \boldsymbol{u}(t+1))$.

Fig. 6 shows the offsets (D_x, D_y, D_z) recorded from a demonstration of the peg-into-hole task after the preprocessing step. The relevant parameters were set to $\delta = 0.05[mm] = \frac{1}{2}\times$ (position accuracy of the Puma 260 in use), and $\epsilon = 0.1\bar{\alpha}, \bar{\alpha} = \frac{1}{dim(\boldsymbol{u})} \sum_{i=1}^{dim(\boldsymbol{u})} \alpha_i$.

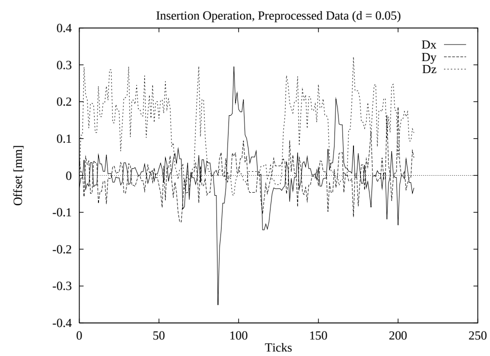

Fig. 6.: Actions (D_x, D_y, D_z) after preprocessing. Note that the peg extraction phase has been removed.

3.2 Generating training data

The step of training data generation, following the preprocessing of the recorded example, aims at reducing the dimension of both the perception and the action space. This reduction to the **necessary information**, i.e., to those components of the actionvector $\boldsymbol{u}$ that were relevant during the demonstration, as well as to those inputs $\boldsymbol{y_i}$ that were actually used by the operator, significantly eases the task of learning the skill [4]. We have already shown [8, 7] that an information gain measure is a suitable mean for performing this identification task. However, this approach requires both a sufficient amount of samples taken during the demonstration, and a minimum quality of the demonstration in terms of consistency of the commanded actions. If tasks involve multidimensional action vectors, the latter requirement is often not fulfilled.
A second possibility to rank the importance of a particular component of the action vector for the general skill is to measure the individual contribution of this component to performing the skill. If $||\boldsymbol{u_i}(t)||$ is the normed[4] action component i at time t, the individual contribution of this component is

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

Afterwards, the set U of necessary action components can be determined as

$$U : U \in 2^{\{1,\ldots,dim(\boldsymbol{u})\}} : \sum_{i \in U} K_i \geq \rho_{min} \text{ and } |U| = min_{\{I \subseteq 2^{\{1,\ldots,dim(\boldsymbol{u})\}} : \sum_{i\in I} K_i \geq \rho_{min}\}} \{|I|\},$$

i.e., as the minimum subset of components of $\boldsymbol{u}$ whose combined contribution is above a certain threshold ρ_{min} with $0 < \rho_{min} \leq 1$. Usually, ρ_{min} is chosen to be $\rho_{min} \in [0.9, 1.0]$, depending on the confidence in

[3] In general, no strict equality $\boldsymbol{u}(t) \approx \alpha \boldsymbol{u}(t+1)$ will be achieved. Therefore, we consider $\boldsymbol{u}(t) = \alpha \boldsymbol{u}(t+1)$ iff $\forall i \in \{1, \ldots, dim(\boldsymbol{u})\} : \boldsymbol{u_i}(t) = \alpha_i \boldsymbol{u_i}(t+1), \alpha_i \in [\alpha - \epsilon, \alpha + \epsilon]$.

[4] The norm $||||$ must take the different nature of the individual degrees of freedom into account. For the Puma 260 manipulator, the norm is defined as $||x|| = \frac{|x|}{1[mm]}$ for the translational degrees of freedom, and $||\alpha|| = \frac{|\alpha|}{2[deg]}$ for the rotational ones.

the efficiency of the human demonstration. Setting ρ_{min} to the "default" value of 0.95 resulted in selecting D_x, D_y, and D_z as the necessary action components.
After the identification of the necessary action components, the analysis of the perception components takes place. What is used for determining the relevant perception components is the set of partial derivatives $\frac{\partial \boldsymbol{y_j}(t)}{\partial \boldsymbol{u_i}(t-d)}$, i.e., the change of a particular sensorial input with respect to the change in the commanded input[5]. These derivatives are specific to the system and do not depend on the quality of the demonstration. Therefore, if it is possible to determine those perception components $\boldsymbol{y_j}$ whose change depends on any of the action components $\boldsymbol{u_i} \in U$, these $\boldsymbol{y_j}$ can be assumed to be the relevant perceptions. To determine those components, an information gain measure can be employed. For each pair $(\boldsymbol{y_j}, \boldsymbol{u_i}), j \in \{1, \ldots, dim(\boldsymbol{y})\}, \boldsymbol{u_i} \in U$, $I(\boldsymbol{y_j}, \boldsymbol{u_i})$ is defined as

$$I(\boldsymbol{y_j}, \boldsymbol{u_i}) := E(\Delta \boldsymbol{y_j}) - E^c(\Delta \boldsymbol{y_j} | \boldsymbol{u_i})$$

where E denotes the estimated entropy of $\Delta \boldsymbol{y_j}$ and $E^c(\Delta \boldsymbol{y_j}|\boldsymbol{u_i})$ is the conditional entropy of $\Delta \boldsymbol{y_j}$ given $\boldsymbol{u_i}$. The set Y of interesting perception components $\boldsymbol{y_k}$ can then be determined as

$$Y = \bigcup_{j \in \{1, \ldots, dim(\boldsymbol{y})\}} \boldsymbol{y_j} : \kappa E(\Delta \boldsymbol{y_j}) \leq \sum_{\boldsymbol{u_i} \in U} I(\boldsymbol{y_j}, \boldsymbol{u_i})$$

with $0 < \kappa \leq 1$. Since coupled dependencies are likely to exist, usually κ is chosen to be in $[0.3, 0.5]$. For $\kappa = 0.4$, the method calculated $Y = \{F_z, M_x, M_y\}$.

3.3 Offline skill learning

After the sets U and Y have been identified, the skill represented by the sampled data is to be learnt off-line. Because of their support of incremental learning, the possibility to represent alternative actions, the existence of algorithms that allows the network construction given a set of samples, and the straightforward symbolic interpretation, radial-basis function networks [12, 14] were chosen for this task.
Based on the preprocessed data, the network representing the insertion skill was constructed using a clustering algorithm [2], which generated a network featuring 50 clusters. Afterwards, the network was trained using conventional error backpropagation until convergence was achieved. Since the given examples could not assumed to be optimal, the convergence criterion for this kind of learning task was defined as

$$\forall t \in \{0, \ldots, T\} : \boldsymbol{u}(t) = \alpha \boldsymbol{u}^*(t) \text{ with } 0.5 < \alpha \leq 1.0,$$

where $\boldsymbol{u}^*$ is the reduced action (with only components $\in U$) performed by the human operator.

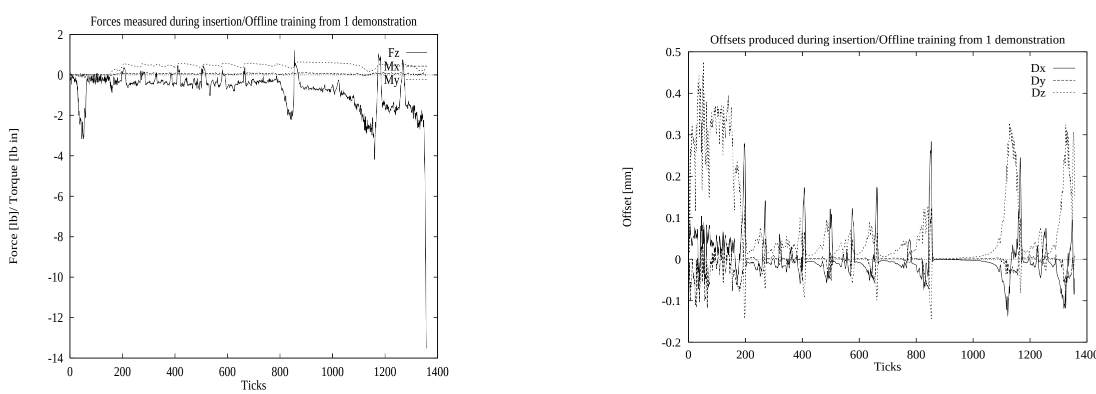

Fig. 7.: Left: Forces measured during the peg-into-hole insertion under neural network control. Right: Offsets produced by the network.

This network was able to control a peg-into-hole insertion on the Puma 260. However, as Fig. 7 shows, the performance obtained from offline-learning based on a single demonstration is not satisfactory. Therefore, online adaptation is a definite requirement and a topic of current work.

[5] Since the recorded $\boldsymbol{u_i}$ are translational or rotational offsets and not positions, in the specific case considered here we are interested in $\frac{\partial \boldsymbol{y_j}(t)}{\boldsymbol{u_i}(t-d)}$. d is specific to the system and has been identified before. For the Puma 260 and the controller in use, $d = 5$.

4 Summary

Throughout this paper, the acquisition of both open-loop and closed-loop skills from human demonstrations has been considered. It has been shown that, given an appropriate analysis of the examples obtained from the demonstrations, skills can be acquired that are executable on the real robot. However, especially in the case of closed-loop skills, user demonstrations are seldom optimal. Both preprocessing and learning procedures must take this fact into account. Also, skill refinement becomes a must in this context.
Our future work will therefore focus on two aspects. On the one hand, the preprocessing techniques will be developed further, e.g., by utilizing digital filters to enhance the robustness of the trajectory segmentation. On the other hand, we will investigate to which extent models and feedback signals required for on-line adaptation can be **automatically** generated from the demonstration and the intention of the user.

Acknowledgement: This work has been partially funded by the ESPRIT Project 7274 "B-Learn II". It has been performed at the Institute for Real-Time Computer Systems & Robotics, Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, Department of Computer Science, University of Karlsruhe, Germany. A. Ude is on leave from the Jožef Stefan Institute, Ljubljana, Slovenia.

References

1. H. Asada and S. Liu. Transfer of human skills to neural net robot controllers. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991.
2. C. Baroglio, A. Giordana, M. Kaiser, M. Nuttin, and R. Piola. Learning controllers for industrial robots. *Machine Learning*, 1995.
3. N. Delson and H. West. Robot programming by human demonstration: The use of human inconsistency in improving 3D robot trajectories. In *Proc. IEEE/RSJ/GI Int. Conf. Intelligent Robots and Systems*, pages 1248–1255, Munich, Germany, September 1994.
4. A. Giordana, M. Kaiser, and M. Nuttin. On the reduction of costs for robot controller synthesis. In *International Symposium on Intelligent Robotic Systems (IRS '94)*, pages 187 – 197, Grenoble, France, 1994.
5. R. Heise. *Demonstration instead of programming: Focussing attention in robot task acquisition.* Research report no. 89/360/22, Department of Computer Science, University of Calgary, 1989.
6. P. N. Johnson-Laird. *The Computer and the Mind: An Introduction to Cognitive Science.* Harvard University Press, Cambridge, 1988.
7. M. Kaiser, H. Friedrich, and R. Dillmann. Obtaining good performance from a bad teacher. In *International Conference on Machine Learning, Workshop on Programming by Demonstration*, Tahoe City, California, 1995.
8. M. Kaiser, A. Retey, and R. Dillmann. Robot skill acquisition via human demonstration. In *Proceedings of the International Conference on Advanced Robotics (ICAR '95)*, 1995.
9. H. Kollnig and H.-H. Nagel. Ermittlung von begrifflichen Beschreibungen von Geschehen in Straßenverkehrsszenen mit Hilfe unscharfer Mengen. *Informatik Forsch. Entw.*, 8:186–196, 1993.
10. Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1995.
11. J.-C. Latombe. *Robot Motion Planning.* Kluwer, Boston, 1991.
12. J. Moody and C. Darken. Learning with localized receptive fields. In T. Sejnowski D. Touretzky, G. Hinton, editor, *Proceedings of the Connectionist Models Summer School.* Carnegie Mellon University, 1988.
13. H. Ogata and T. Takahashi. A geometric approach to task understanding for robot assembly operations. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 58 – 64, 1993.
14. T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, September 1990.
15. D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3:88 – 97, 1991.
16. P. Reignier, V. Hansen, and J.L. Crowley. Incremental supervised learning for mobile robot reactive control. In *Intelligent Autonomous Systems 4 (IAS-4)*, pages 287 – 294. IOS Press, 1995.
17. J. G. Schneider and C. M. Brown. Robot skill learning, basis functions, and control regimes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, 1993.
18. S. K. Tso and K. P. Liu. Visual programming for capturing of human manipulation skill. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 42–48, Yokohama, Japan, July 1993.
19. A. Ude. Trajectory generation from noisy positions of object features for teaching robot paths. *Robotics and Autonomous Systems*, 11(2):113–127, September 1993.
20. A. Ude, H. Bröde, and R. Dillmann. Object localization using perceptual organization and structural stereopsis. In *Proc. Third Int. Conf. Automation, Robotics and Computer Vision*, pages 197–201, Singapore, November 1994.
21. G. Wahba. *Spline Models for Observational Data.* SIAM, Philadelphia, 1990.
22. H.-J. Zimmermann. *Fuzzy Set Theory — and its Applications.* Kluwer, Boston, 1991.

This article was processed using the TEX macro package with SIRS95 style