Recognition and Teaching of Robot Skills by Fuzzy Time-Modeling

Rainer Palm, Bourhane Kadmiry, Boyko Iliev, and Dimiter Driankov
AASS, Dept. of Technology Orebro University SE-70182 Orebro, Sweden
Emails: rub.palm@t-online.de, {bourhane.kadmiry, boyko.iliev, dimiter.driankov}@aass.oru.se

Abstract – Robot skills are low-level motion and/or grasping capabilities that constitute the basic building blocks from which tasks are built. Teaching and recognition of such skills can be done by Programming-by-Demonstration approach. A human operator demonstrates certain skills while his motions are recorded by a data-capturing device and modeled in our case via fuzzy clustering and Takagi-Sugeno modeling technique. The resulting skill models use the time as input and the operator's actions and reactions as outputs. Given a test skill by the human operator the robot control system recognizes the individual phases of skills and generates the type of skill shown by the operator.

Keywords – Fuzzy modeling, time clustering, robot skills, Programming-by-Demonstration

1 Introduction

Robot skills are low-level motion and/or grasping capabilities that constitute the basic building blocks from which tasks are built. One major challenge in robot programming is to be able to program skills in an easy and fast manner with high accuracy. *Programming by Demonstration (PbD)* (PbD) is one such approach that has the afore mentioned properties. In PbD a human who demonstrates the skills is equipped with data-capturing devices (e.g., data glove, cameras, haptic devices etc.). The demonstrator performs a skill while the robot captures the associated motion data, analyzes it and generates a robot-centered model of the demonstrated skill, that is a corresponding robot skill. Once the robot has acquired a number of robot skills from demonstrations it is able to recognize a demonstrated human skill as one of its already available robot skills. In a final step, when a task is demonstrated then the robot recognizes the robot skills that constitute it and thus creates a program consisting of these skills. This approach can be used not only for industrial robots but also in the fields of prosthetics, humanoid service robots, remote control and teleoperation in hasardous and dangerous environments, and last but not least in the entertainment industry. However, such applications are relatively few so far due to the lack of appropriate sensor systems and some unsolved problems with the man-robot interaction. Selected skills are

- contour following
- assembly (peg-in-hole insertion)
- handling of objects
- grasping of objects.

Different techniques for recognition of skills have been applied for PbD. For the manipulation domain Morrow and Khosla describe the construction of a library of robot capabilities by analysis and identification of tasks using a camera and a force-torque sensor [1]. Their approach is to develop a sensorimotor layer which integrates sensing into the robot programming primitives. In [2] Kaiser and Dillmann describe a neural net approach for the initial skill learning and reinforcement learning, skill refinement and adaptation. Experimental results were shown by an insertion example and a door-opening experiment. In the context of task learning Geib et. al. proposed an approach to integrating high-level artificial intelligence planning technology with low-level robotic control [3]. Kwun Han and M. Veloso describe an automated recognition of the behavior of robots using HMMs to represent and recognize strategic behaviors of robotic agents [4]. In the field of recognition of robot behaviors the following publications are important: Zoellner et. al. [5] use a data glove with integrated tactile sensors for behavior recognition which is based on support vector machines (SVM). Ekvall and Kragic [6] apply Hidden Markov Models (HMM) and address the PbD-problem using the arm trajectory as an additional feature for grasp classification. Li et. al. [7] use the singular value decomposition (SVD) for the generation of feature vectors of human grasps and support vector machines (SVM) which are applied to the classification problem. A fuzzy logic approach for gesture recognition was published by Bimber [8]. The method is applied to 6 d.o.f. (degrees of freedom) trajectories of a human arm but cannot cope with more than 6 dimensions. Despite the advanced state of the art, the cited methods on dynamic classification do not consider the evolution of a robot behavior in time and space. This appears to be a disadvantage because neither the estimation of the occurence of a specific skill nor the problem of segmentation can be solved in a general way. The method presented in this paper tries to overcome some of these drawbacks by modeling trajectories in time and space. Its advantage over other methods like HMM or Gaussian Mixture Models is discussed in [9] and [10]. To solve a 'fuzzy' task like skill recognition it turns out that fuzzy modeling is a suitable approach to deal with. The main focus of this method is *programming by demonstration* and *recognition of human skills* using fuzzy clustering and Takagi-Sugeno (TS) fuzzy modeling (see also [11] and [12]). The paper is organized as follows: In Section 2 a general approach to learning skills by partitioning them into phases is discussed. Section 3 deals with fuzzy time-modeling and the segmentation principle. Section 4 describes the recognition of phases and the decision process for the classification of skills. Section 5 presents simulations and experimental results. The final Section 6 draws some conclusions and directions for future work.

ISBN: 978-989-95079-6-8

2 Programming of robot skills by human demonstration

Programming of robot skills requires the building of a library of models of skills being taught or trained by a human demonstrator. In a next step newly demonstrated skills lead to test models which are then compared with the training models (skills) of the library. By such a comparison the robot is able to recognize these newly demonstrated skills. Finally a robot task including the recognized skills can automatically be generated (see Fig. 1). For the training phase two main tasks are needed to perform: *segmentation* of human demonstrations into skill phases and *phase modeling* of the skill phases. *Segmentation* means a partition of the data record into a sequence of episodes, where each one contains a single skill phase. For the test phase three main tasks are needed to perform: *segmentation* of the human test demonstrations, *phase recognition*, and *skill classification*. *Phase recognition* means to recognize the phases performed in each episode. The third task is to connect the recognized skill phases in such a way that a full skill can be identified.

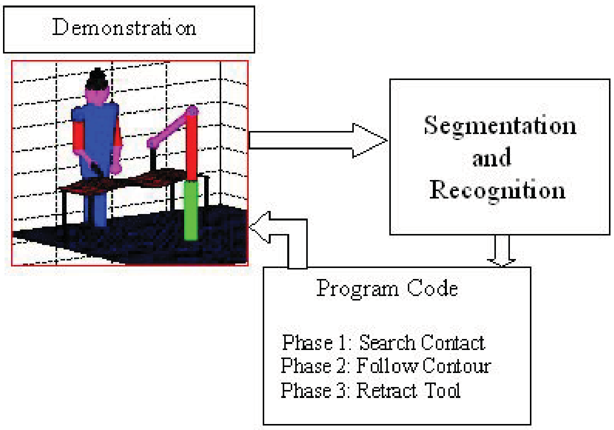

Figure 1: Learning skills from human demonstrations

3 Fuzzy time-modeling and the segmentation principle

Let a skill be partitioned into a sequence of phases as described above. Each phase starts and ends with a discrete event coming either from a discrete sensor or from some appropriate preprocessing of continuous sensor signals. The structure of a skill can be described most appropriately by a hybrid automaton in which nodes represent continuous phases and arcs the discrete transitions (switches) between them. Figure 2 shows an example of such a hybrid automaton. The hybrid process is event-controlled (see [13]) and is assumed to be stable both within the individual phases and with respect to the switching behavior between them. On the other hand the purpose of segmentation is to identify the discrete instants of time for the switches to occur in order to cut the whole skill into phases during demonstration.

The following subsection deals with fuzzy time-modeling in general that is used both for phase modeling and for segmentation. In the same context the training of time cluster models using new data is described. After that the segmentation procedure is presented.

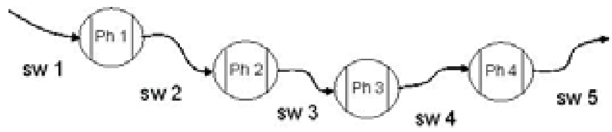

Figure 2: Hybrid automaton of a skill

3.1 Fuzzy time-modeling

Let us for the time being only concentrate on the modeling of a skill phase. The recognition of a skill phase is achieved by a model that reflects the *behavior of the robot end-effector in time* during the episode considered. Each demonstration is repeated several times to collect enough samples of every particular skill phase. From those data, models for each individual phase are developed using fuzzy clustering and Takagi-Sugeno fuzzy modeling ([14, 11]). We consider time instants as model inputs and end-effector coordinates as model outputs. Define the end-effector coordinate by

$$\mathbf{x}(t) = \mathbf{f}(t) \tag{1}$$

where $\mathbf{x}(t) \in R^3, \mathbf{f} \in R^3$, and $t \in R^+$. Further linearize (1) at selected time points t_i

$$\mathbf{x}(t) = \mathbf{x}(t_i) + \frac{\Delta \mathbf{f}(t)}{\Delta t}|_{t_i} \cdot (t - t_i) \tag{2}$$

which is a linear equation in t,

$$\mathbf{x}(t) = \mathbf{A}_i \cdot t + \mathbf{d}_i \tag{3}$$

where $\mathbf{A}_i = \frac{\Delta \mathbf{f}(t)}{\Delta t}|_{t_i} \in R^3$ and $\mathbf{d}_i = \mathbf{x}(t_i) - \frac{\Delta \mathbf{f}(t)}{\Delta t}|_{t_i} \cdot t_i \in R^3$. Using (3) as a local linear model one can express (1) in terms of a Takagi-Sugeno fuzzy model [15]

$$\mathbf{x}(t) = \sum_{i=1}^{c} w_i(t) \cdot (\mathbf{A}_i \cdot t + \mathbf{d}_i) \tag{4}$$

$w_i(t) \in [0, 1]$ is the degree of membership of a time point t to a cluster with the cluster center t_i, c is the number of clusters, and $\sum_{i=1}^{c} w_i(t) = 1$.

Let $\mathbf{x} = [x_1, x_2, x_3]^T$ be the 3 end-effector cordinates and t the time. The general clustering and modeling steps are described as follows

- Select an appropriate number of local linear models (data clusters) c
- Find c cluster centers $(t_i, x_{1i}, x_{2i}, x_{3i})$, $i = 1...c$, in the product space of the data quadruples (t, x_1, x_2, x_3) by Fuzzy-c-elliptotype clustering
- Find the corresponding fuzzy regions in the space of input data (t) by projection of the clusters of the product space into Gustafson-Kessel clusters (GK) within the input space [16]
- Calculate c local linear (affine) models (4) using the GK clusters from step 2.

ISBN: 978-989-95079-6-8

The membership degree $w_i(t)$ of an input data point t in an input cluster C_i is then calculated by

$$w_i(t) = \frac{1}{\sum_{j=1}^{c} \left(\frac{(t-t_i)^T M_{i\,pro}(t-t_i)}{(t-t_j)^T M_{j\,pro}(t-t_j} \right)^{\frac{1}{\tilde{m}_{proj}-1}}} \qquad (5)$$

The projected cluster centers t_i and induced matrices M_{ipro} define the input clusters C_i $(i = 1 \ldots c)$. The parameter $\tilde{m}_{pro} > 1$ determines the fuzziness of an individual cluster. The spheres in Fig. 3 covering the trajectories represent the local fuzzy models. The stripes along the time coordinate represent projections of the local fuzzy models onto the time line.

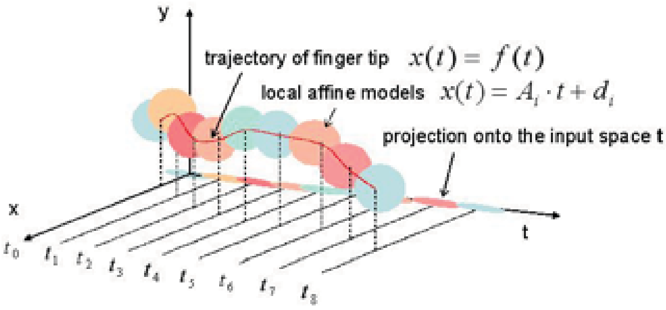

Figure 3: Time-clustering principle for the end-effector and its motion in (x,y)

3.1.1 *Training of time cluster models using new data*

A skill model can be built in several ways

- A single user trains the model by repeating the same skill n times

- m users train the model by repeating the same skill n times

The 1st model is generated by the time sequences $[(t_1, t_2, ..., t_N)_1...(t_1, t_2, ..., t_M)_n]$ and the output (end-effector position) sequences $[(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)_1 \cdots (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M)_n]$.

The 2nd model is generated by the time sequences $[((t_1, t_2, ..., t_N)_1^1...(t_1, t_2, ..., t_M)_n^1)$ ... $((t_1, t_2, ..., t_N)_1^m...(t_1, t_2, ..., t_M)_n^m)]$ and the output sequences $[((\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)_1^1...(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M)_n^1)$ ... $((\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)_1^m...(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M)_n^m)]$ where m is the number of users in the training process, N, M are lengths of time sequences where $N \approx M$.

Once a particular skill model has been generated it might be necessary to take new data into account. These data may originate from different human operators to cover several ways of performing the same type of skill. Let for simplicity the old model be built by a time sequence $[t_1, t_2, ..., t_N]$ and a respective output sequence $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$. The old model is then represented by the input cluster centers t_i and the output cluster centers $\mathbf{x}_i$ $(i = 1...c)$. It is also described by the parameters $\mathbf{A}_i$ and $\mathbf{d}_i$ of the local linear models. Let $[\tilde{t}_1, \tilde{t}_2, ..., \tilde{t}_M]$, $[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_M]$ be new training data. A new model can be built by "chaining" old and new training data leading for the time sequences to $[t_1, t_2, ..., t_N, \tilde{t}_1, \tilde{t}_2, ..., \tilde{t}_M]$, and for the output sequences to $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_M]$. The result is a model that involves properties of the old model and the new data. If the old sequence of data is not available, a corresponding sequence can be generated by running the old model with the time instants $[t_1, t_2, ..., t_N]$ as inputs and the end-effector positions $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ as outputs.

3.2 *Segmentation principle*

Let for simplicity the signals of a skill be the end-effector coordinates $\mathbf{x}(t) \in R^3$ and the forces $\mathbf{f}(t) \in R^3$ in the end-effector. In order to generate the 'events' determining the time bounds for the phases, $\mathbf{x}(t)$ is differentiated twice and $\mathbf{f}(t)$ only ones by time. The absolute values of the resulting vectors are collected in

$$\mathbf{X}(t) = [|\ddot{\mathbf{x}}(t)|^T, |\dot{\mathbf{f}}(t)|^T]^T. \qquad (6)$$

For a segmentation we need the time-discrete case

$$\tilde{\mathbf{X}} = [\mathbf{X}(t_1)...\mathbf{X}(t_n)] \in R^{6\times n}. \qquad (7)$$

Further define a vector of bounds $\mathbf{B} > 0 \in R^6$ above which $\mathbf{X}(t_i)$ are counted as 'events'. Then a vector $\mathbf{I} = [I_1...I_k...I_m]^T$ is generated where I_k are discrete time stamps t_i for which at least one component of $\mathbf{X}(t_i)$ lies above the corresponding component of the vector of bounds $\mathbf{B}$.

$$I_k = t_i \quad if \quad \mathbf{X}(t_i) > \mathbf{B}. \qquad (8)$$

The next step is to select the number of time clusters c and find the clusters by time clustering for the data $\mathbf{Y} = ([\mathbf{X}(I_1); I_1]...[\mathbf{X}(I_k); I_k]...[\mathbf{X}(I_m); I_m])$. $\mathbf{Y}$ is a combination of 'events' $\mathbf{X}(t_i)$ and their corresponding time instants I_k. Once the time clusters are found the skill could be cut at these time intants into phases. However, for complicated skills the number of phases c might not be known in advance. Therefore we choose a higher number c and merge those cluster centers who are located close to each other into one.

4 Recognition of robot skills

In this section the recognition of phases or sub-skills ist discussed first. In a second step on the basis of this knowledge - number and type of phases - the construction of the full skill is presented.

4.1 *Recognition of phases using the distance between fuzzy clusters*

Let the model of each phase have the same number of clusters $i = 1...c$ so that each duration T_l $(l = 1...L)$ of the l-th phase is divided into $c - 1$ time intervals Δt_i, $i = 2...c$ of the same length. Let the phases be executed in an environment comparable with the modeled phase in order to avoid calibration and re-scaling procedures. Furthermore let

$$V_{modell} = [\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_c]_{model,l}$$
$$\mathbf{X}_i = [x, y, z, f_x, f_y, f_z]_i^T$$

where matrix V_{modell} includes the output cluster centers $\mathbf{X}_i$ for the l-th phase model.

A model of the phase to be classified is built by the matrix

$$V_{test} = [\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_c]_{test,l} \qquad (9)$$

ISBN: 978-989-95079-6-8

A decision about which phase is present is made by applying the Euclidean matrix norm

$$N_l = ||V_{model\,l} - V_{test}|| \tag{10}$$

Once the unknown phase is classified to the phase model with the smallest norm $min(N_l)$, $l = 1...L$ then the recognition of the phase is finished.

4.2 Recognition of skills using phase models

Once the phases of a test skill are recognized (identified) one should be able to recognize the skill as a whole and finally to reconstruct the hybrid automaton that represents the skill (see Fig. 2). For this purpose a list of possible skills and their phases should be produced. In the following we will discuss the following robot skills
- handling
- contour following
- assembly

The corresponding phases can be found in table 1. Figures

Table 1: Skills and phases

Phases Skills:	handling	contour	assembly
1. Grasp object	x		x
2. Free motion	x		
3. Search contact	x	x	x
4. Keep contact	x		
5. Follow with contact	x	x	x
6. Peg-in hole			x
7. release object/contact	x	x	x

4,5, and 6 show the correspondence between the skills and their individual phases.

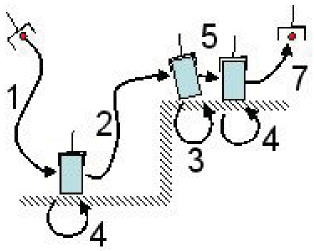

Figure 4: Handling skill

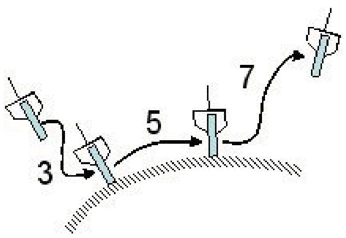

Figure 5: Contour following skill

5 Experiments and simulations

In this section an experimental evaluation of the recognition of phases will be presented. The experimental platform comprises a data glove with diodes mounted at the fingertips and links (see Fig. 7). A system of 4 stereo cameras takes records

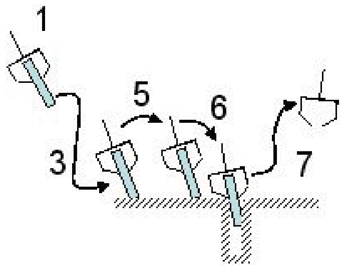

Figure 6: Assembly skill

of the positions of the diodes so that the position of the hand and its fingers can be tracked. In addition, tactile sensors are mounted at each finger tip in order to detect the contact between the fingertips and an object or a surface, respectively. In the experiment only the tip of the index finger is tracked in order to identify the contour performed by this finger. The experiments described here cover only contour following examples but using different contours running at different speeds. Each example consists of three phases: the approach phase, the contour following phase, and the retract phase. The experiment starts with the index fingertip being in contact with a defined start location at a distance from the contour. Then it follows the approach phase without any contact to an object ending at the begin of the contour to be followed. In the contour following phase the index finger follows the contour while the contact is preserved until the end of the contour is reached and the retract phase starts. During the retract phase there is no contact until the index finger reaches the start location again. The experiments can be divided into 3 groups:

A Straight lines
1: slow speed (see Fig. 8)
2: fast speed
3: ramp downhill slow
4: ramp downhill fast
5: ramp uphill slow
6: ramp uphill fast

B Meander
7: meander slow (see Fig. 9)
8: meander fast

C Loops
9: loop slow 1 (see Fig. 10)
10: loop slow 2
11: loop fast 1

Three modeling examples are shown in Figs. 8 - 10. The blue curves represent the modeled phases whereas the red curves represent the original data. Each phase of a skill is modeled by 15 cluster centers. The crosses depict the cluster centers. It can be observed that the modeling/approximation quality of the fuzzy time models is excellent. The partition of the skill into phases has been done by means of the forces applied to the tip of the index finger. Figure 11 shows the time plots for the meander experiment 7. By means of the force f applied to the fingertip and its derivative df a segmen-

ISBN: 978-989-95079-6-8

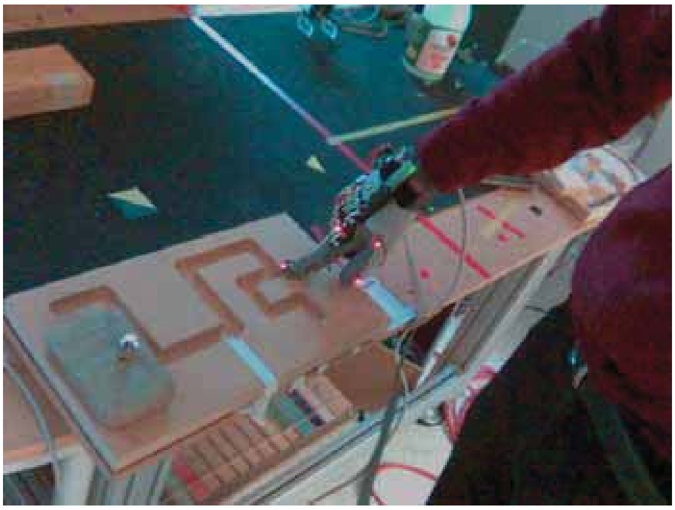

Figure 7: Experiment with a meader-like contour

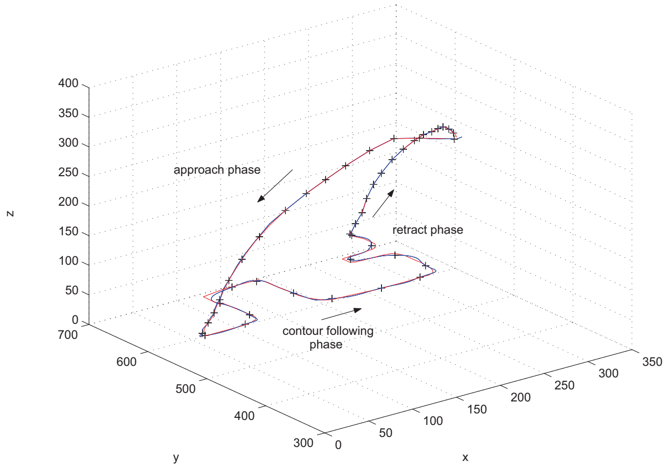

Figure 9: contour following, meander

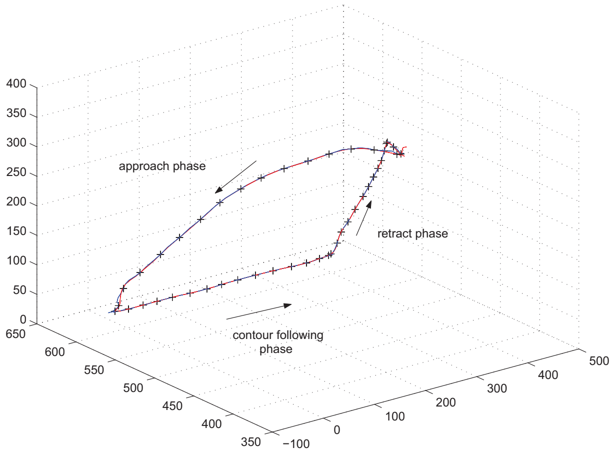

Figure 8: contour following, line

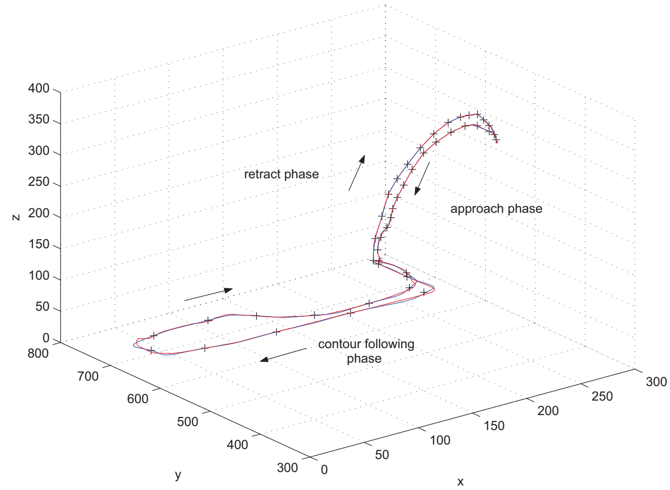

Figure 10: contour following, loop

tation can be done very easily because of the distinct derivatives of the force signals df. The recognition of skills has been done by comparing each skill with all other skills using the method described in the last section. Since we only deal with contour following experiments the norms over the whole skill have been taken into account instead of considering the phases seperately. The results are shown in Table 2. To explain the matrix $M(i,j)$ $(i,j = 1...11)$ in Table 2, let us consider experiment 3, *ramp downhill slow*, as an example (3rd row), $M(3,i)$ $(i = 1...11)$. Compared to itself the norm of the differences between model and test skill is zero, $M(3,3) = 0$. The next higher norm difference can be observed for experiment 4, *ramp downhill fast*, $M(3,4) = 0.3$. This corrresponds completely with the idea that a similarity of trajectories should lead to small norm differences. Going through all 11 experiments it turns out that almost all contour following skills can be indentified. One exception is experiment 6 where either skill 1 or 2 are identified instead of skill 5 as expected, $M(6,1) \wedge M(6,2) < M(6,5)$. It can also be noticed that the groups "A: Straight lines", $M(i,j)$ $(i,j = 1...6)$, "B: Meander" , $M(k,l)$ $(k,l = 7,8)$, and "C: Loops" , $M(m,n)$ $(m,n = 9...11)$, can be significantly distinguished from each other. Furthermore, one can see that, from the recognition point of view, the groups A and B are more related with each other than B and C or A and C, respectively.

6 Conclusions

In this paper modeling and recognition/identification of robot skills using fuzzy time modeling have been presented. Recognition of skills is a part of the teach-in process and the Programming-by-Demonstration (PbD) of robot tasks by a human operator. In this paper the focus is directed to the skills "handling", "contour following", and "assembly". In this context a partitioning of skills into phases by segmentation and their modeling using fuzzy time clustering is discussed. The recognition of phases and skills is done by comparing fuzzy time clusters of model skills and test skills. The experimental platform consists of a data glove with diodes mounted at the fingertips and links. 4 stereo cameras track the positions of the hand and its fingers. Tactile sensors are mounted at each fingertip to detect the contact between the fingertips and objects. In the experiments different groups of contour following skills are presented. In 11 experiments several contours like "straight line", "meander", or "loop", respectively,

ISBN: 978-989-95079-6-8

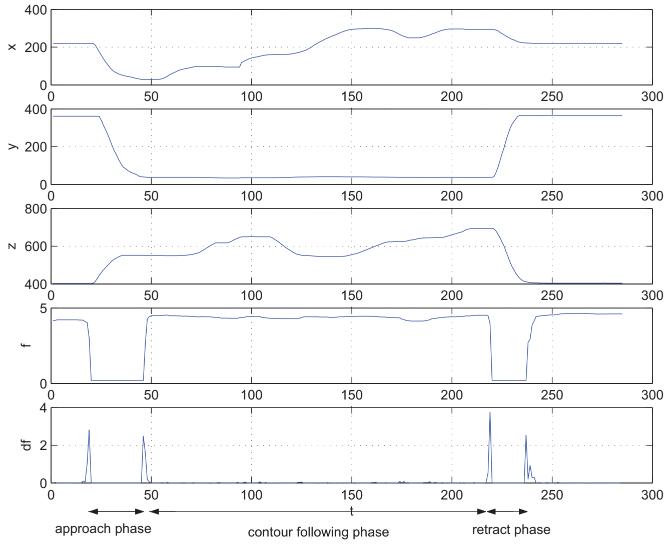

Figure 11: contour following, loop, time plots

Table 2: identification results

skill	1	2	3	4	5	6	7	8	9	10	11
1	0	0.2	0.6	0.5	0.6	0.4	0.7	0.9	1.8	1.8	1.8
2	0.2	0	0.7	0.6	0.6	0.5	0.7	0.9	1.8	1.9	1.8
3	0.5	0.7	0	0.3	0.8	0.6	0.9	1.0	1.9	1.9	1.8
4	0.5	0.6	0.3	0	0.8	0.6	0.8	0.9	1.8	1.9	1.8
5	0.6	0.6	0.8	0.8	0	0.5	0.9	1.0	1.9	2.0	1.9
6	0.4	0.5	0.6	0.6	0.5	0	0.8	1.0	1.8	2.0	1.9
7	0.7	0.7	0.9	0.8	0.9	0.8	0	0.4	1.4	1.4	1.4
8	0.9	0.9	1.0	0.9	1.0	1.0	0.4	0	1.5	1.4	1.3
9	1.8	1.8	1.9	1.8	1.9	1.8	1.4	1.5	0	0.6	0.4
10	1.8	1.9	1.9	1.9	2.0	1.9	1.4	1.4	0.6	0	0.5
11	1.8	1.8	1.8	1.8	1.9	1.9	1.4	1.4	0.4	0.5	0

are performed by the index fingertip. The segmentation has been done by differentiation of the tactile forces. The modeling and recognition results were very good to excellent. In the future more experiments will be done for handling and assembly tasks. The recognition of skills using fuzzy time modeling and hybrid automata will be further developed. A further focus will be the integration of high-level AI planning techniques with low-level robotic control.

References

[1] J. D. Morrow and P. K. Khosla. Manipulation task primitives for composing robot skills. *Proceedings of the 1997 IEEE Intemational Conference on Robotics and Automation, Albuquerque, New Mexico*, pages 3354–3359, 1997.

[2] M. Kaiser and R. Dillmann. Building elementary robot skills from human demonstration. *Proceedings of the 1996 IEEE Intemational Conference on Robotics and Automation, Minneapolis, Minnesota*, 1996.

[3] C. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krueger, and F. Woergoetter. Object action complexes as an interface for planning and robot control. In *Proc. IEEE RAS Int Conf. Humanoid Robots(Genova)*, Genova, Dec. 4-6 2006. IEEE, IEEE.

[4] Kwun Han and M. Veloso. Automated robot behavior recognition. *Proceedings of the 1999 ijcai Conference*, 1999.

[5] R. Zoellner, O. Rogalla, R. Dillmann, and J.M. Zoellner. Dynamic grasp recognition within the framework of programming by demonstration. In *Proceedings Robot and Human Interactive Communication,10th IEEE International Workshop*, Bordeaux, Paris, France, September 18-21 2001. IEEE.

[6] S. Ekvall and D. Kragic. Grasp recognition for programming by demonstration. In *Proceedings of International conference on robotics and automation, ICRA 2005*, Barcelona, Spain, 2005.

[7] Ch. Li, L. Khan, and B. Prabhaharan. Real-time classification of variable length multi-attribute motions. *Knowledge and Information Systems*, DOI 10.1007/s10115-005-0223-8, 2005.

[8] O. Bimber. Continuous 6d gesture recognition: A fuzzy-logic approach. *Proceedings of 7-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, WSCG'99:24–30, 1999.

[9] R. Palm and B. Iliev. Grasp recognition by time clustering, fuzzy modeling, and hidden markov models (hmm) - a comparative study. In *Proceedings FUZZ-IEEE 2008 - IEEE International Conference on Fuzzy Systems*, Hong Kong, July 1-5 2008. IEEE.

[10] Z. Ju, H. Liu, X.Y. Zhu, and Y.L. Xiong. Dynamic grasp recognition using time clustering, gaussian mixture models and hidden markov models. In *Proceedings Intern. Conf. on Intell. Robots and Appl.s, ICIRA-2008*, Wuhan,China, Oct. 15-17 2008. Springer.

[11] R. Palm and B. Iliev. Learning of grasp behaviors for an artificial hand by time clustering and takagi-sugeno modeling. In *Proceedings FUZZ-IEEE 2006 - IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, July 16-21 2006. IEEE.

[12] R. Palm and B. Iliev. Segmentation and recognition of human grasps for programming-by-demonstration using time clustering and takagi-sugeno modeling. In *Proceedings FUZZ-IEEE 2007 - IEEE International Conference on Fuzzy Systems*, London, UK, July 23-26 2007. IEEE.

[13] M.S.Branicky, V.S.Borkar, and S.K.Mitter. A unified framework for hybrid control: Background, model, and theory. *Proceedings of the 11th Intern. Conference on Analysis ond Optimization of Systems. INRIA. France*, June 15-17, 1994.

[14] R. Palm and Ch. Stutz. Open loop dynamic trajectory generator for a fuzzy gain scheduler. *Engineering Applications of Artificial Intelligence*, Vol. 16:213–225, 2003.

[15] T. Takagi and M. Sugeno. Identification of systems and its applications to modeling and control. *IEEE Trans. on Syst., Man, and Cyb.*, Vol. SMC-15. No.1:116–132, January/February 1985.

[16] D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. *Proceedings of the 1979 IEEE CDC*, pages 761–766, 1979.

ISBN: 978-989-95079-6-8