Chapter 2
Grasp Recognition by Fuzzy Modeling and Hidden Markov Models

Rainer Palm, Boyko Iliev, and Bourhane Kadmiry

Abstract Grasp recognition is a major part of the approach for Programming-by-Demonstration (PbD) for five-fingered robotic hands. This chapter describes three different methods for grasp recognition for a human hand. A human operator wearing a data glove instructs the robot to perform different grasps. For a number of human grasps the finger joint angle trajectories are recorded and modeled by fuzzy clustering and Takagi-Sugeno modeling. This leads to grasp models using time as input parameter and joint angles as outputs. Given a test grasp by the human operator the robot classifies and recognizes the grasp and generates the corresponding robot grasp. Three methods for grasp recognition are compared with each other. In the first method, the test grasp is compared with model grasps using the difference between the model outputs. The second method deals with qualitative fuzzy models which used for recognition and classification. The third method is based on Hidden-Markov-Models (HMM) which are commonly used in robot learning.

2.1 Introduction

The field of human-like robotic hands has attracted significant research efforts in the last two decades aiming at applications like service robots, prosthetic hands and also industrial applications. However, due to the lack of appropriate sensor systems and some unsolved problems with the human-robot interaction such applications are relatively few so far. One particular reason is the difficult programming procedure due to the high dimensionality of grasping and manipulation tasks. An approach to solve this problem is *Programming-by-Demonstration (PbD)* which is used in complex robotic applications such as grasping and dexterous manipulation. That is, the

R. Palm (✉), B. Iliev, and B. Kadmiry
Department of Technology, Orebro University, 70182 Orebro, Sweden
e-mail: rub.palm@t-online.de; boyko.iliev@tech.oru.se; bourhane.kadmiry@tech.oru.se

H. Liu et al. (eds.), *Robot Intelligence,*
Advanced Information and Knowledge Processing,
DOI 10.1007/978-1-84996-329-9_2,

operator performs a task while the robot captures the data by a motion capture device or a video camera and analyzes the demonstrated actions. Then the robot has to recognize these actions and replicate them in a framework of a complex application. One of the most complicated tasks is the recognition procedure because of the ambiguous nature of a human grasp. Different techniques for grasp recognition have been applied in PbD. Kang et al. [1] describe a system which observes, recognizes and maps human grasps to a robot manipulator using a stereo vision system and a data glove. Zoellner et al. [2] use a data glove with integrated tactile sensors where the recognition is based on support vector machines (SVM). Ikeuchi et al. [3] apply Hidden Markov Models (HMMs) to segment and recognize grasp sequences. Ekvall and Kragic [4] use also HMM methods and address the PbD-problem using the arm trajectory as an additional feature for grasp classification. Li et al. [5] 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. Aleotti and Caselli [6] describe a virtual reality-based PbD-system for grasp recognition where only final grasp postures are modeled based on the finger joint angles. Palm and Iliev presented two methods based on fuzzy models [7] and [8]. Having a look at the rich variety of cited methods it is evident that they do not provide equally successful results. Moreover, the experimental setups often include different sensor suits which makes the comparison of results very difficult.

Therefore, in this article we compare three methods. The first two methods are described in detail in [7] and [8], while the third approach is a hybrid method of fuzzy clustering and HMM-methods. We choose to compare our methods with a HMM-approach since the latter is widely used in robot learning and considered as state-of-the-art. All three methods start with fuzzy time clustering. The 1st method, which is the simplest one, classifies a given test grasp using the distances between the time clusters of the test grasp and the time clusters of a set of model grasps [7]. The 2nd method, which is more complex, is based on qualitative fuzzy recognition rules and solves the segmentation problem and the recognition problem at once [8, 9]. The 3rd method deals with fuzzy time clustering and grasp recognition using HMM's [10]. All three methods are tested on the same set of grasp data in order to provide a fair comparison of the methods. This chapter is organized as follows: Sects. 2.2 and 2.3 describe the experimental platform consisting of a data glove and a hand simulation tool. Section 2.4 discusses the learning of grasps by time-clustering and the training of model grasps. Section 2.5 describes the three recognition methods. Section 2.6 presents the experimental results and gives a comparison of the three methods. Finally Sect. 2.7 draws some conclusions and directions for future work.

2.2 An Experimental Platform for PBD

Robotic grasping involves two main tasks: *segmentation* of human demonstrations and *grasp recognition*. The first task is to partition the data record into a sequence

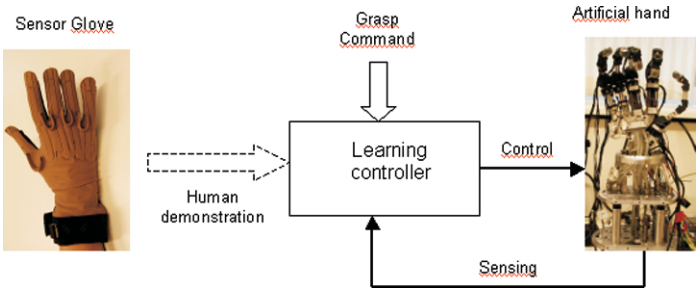

Fig. 2.1 Learning grasp primitives from human demonstrations

of episodes, where each one contains a single grasp. The second task is to recognize the grasp performed in each episode. Then the demonstrated task is (automatically) converted into a program code that can be executed on a particular robotic platform (see Fig. 2.1). If the system is able to recognize the corresponding human grasps in a demonstration, the robot will also be able to perform the demonstrated task by activating the respective grasp primitives.

The experimental platform consists of a hand motion capturing device and a hand simulation environment. The motions of the human operator are recorded by a data glove (CyberGlove) which measures 18 joint angles in the hand and the wrist (see [11]). Since humans mostly use a limited number of grasp types, the recognition process can be restricted to a certain grasp taxonomy, such as those developed by Cutkosky [12] and Iberall [13].

To test the grasp primitives, we developed a simulation model of a five-fingered hand with 3 links and 3 joints in each finger. The simulation environment allows us to perform a kinematic simulation of the artificial hand and its interaction with modeled objects (see Fig. 2.3).

Moreover, we can simulate recorded demonstrations of human operators and compare them with the result from the execution of corresponding grasping primitives. Inspired by the grasp taxonomy of Iberall 15 different grasps have been tested (see Fig. 2.2). These grasps are special cases of the following general classes [13]:

1. cylindrical grasp (grasps 1, 2, 3, 14)
2. power grasp (grasps 4, 5)
3. spherical grasp (grasps 6, 7, 12, 13)
4. extension grasp (grasps 10, 11)
5. precision grasp, nippers pinch (grasps 8, 9)
6. penholder grasp (writing grip) (grasp 15)

The advantage of this selection is that the quality of a grasp recognition both between classes and within a class can be analyzed (see Sect. 2.6: Experiments and Simulations).

We tested the grasping of 15 different objects some of them belonging to same class in terms of an applied type of grasps. For example, *cylinder* and *small bottle* correspond to cylindrical grasp, *sphere* and *cube* to precision grasp, etc.

2.3 Simulation of Grasp Primitives

For the purpose of PbD we need a model of the human hand which allows the simulation of demonstrated grasps. In order to test the grasp primitives a hand simulation was developed with the help of which one can mimic the hand poses recorded by the data glove. This hand model allows to compute the trajectories of both the finger angles and the fingertips. Since the size of the object to be grasped determines the starting and end points of the fingertips to a great extend we used fingertip trajectories instead of finger angles for modeling.

2.3.1 Geometrical Modeling

In order to study grasp primitives and to develop specific grasp models a geometrical simulation of the hand is required (see Fig. 2.3). The hand model consists of 5 fingers which are linked to the wrist in such a way that the poses of a human operator can be displayed in a realistic way. The kinematic relations can be studied by means of the example of a single finger (see Fig. 2.4). Each finger is modeled with 3 links and 3 joints moving like a small planar robot. This turned out to be sufficient for the simulation of the grasp primitives in Fig. 2.2. The calculation of fingertip trajectories requires the formulation of transformations between the fingertips and

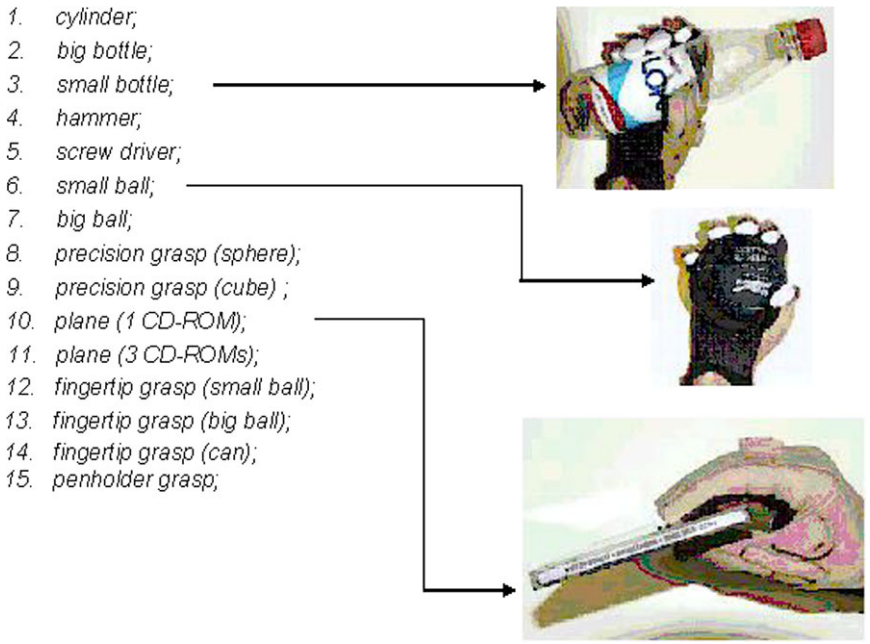

Fig. 2.2 Grasp primitives

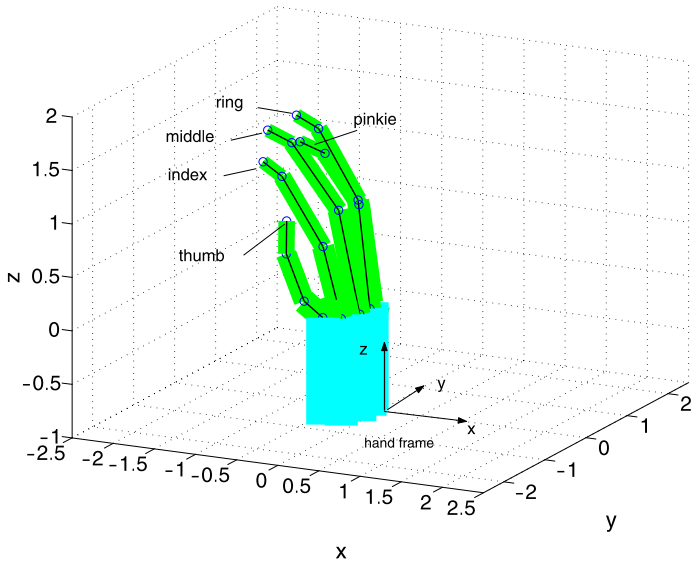

Fig. 2.3 Simulation of the hand

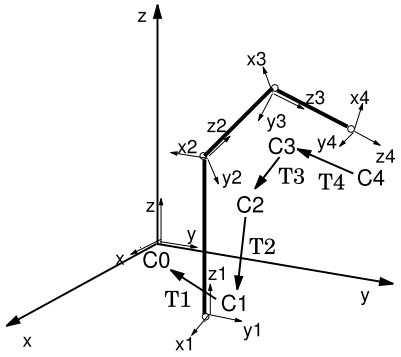

Fig. 2.4 Configuration of a single finger

the base frame of the hand. Translations and rotations between coordinate frames are calculated by homogeneous transformations with the help of which a point $P_{C4} = (x_4, y_4, z_4, 1)^T$ in local homogeneous fingertip coordinates can be transformed into the base frame C_0 by $P_{C0} = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot P_{C4}$. The transformation matrix T_i defines the transformation between the coordinate systems C_i and C_{i-1}.

2.3.2 Modeling of Inverse Kinematics

An important modeling aspect is the *inverse problem* which is crucial both for the simulation of grasps and the control of robotic hands as a feedforward component

of the control law [7]. Given the fingertip position vector $\mathbf{x}(t)$, compute the corresponding joint angle vector $\mathbf{q}(t)$. Let

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{q}); \qquad \mathbf{q}(t) = \mathbf{f}^{-1}(\mathbf{x}) \tag{2.1}$$

be the nonlinear direct and inverse transformation for a single finger where the inverse transformation is not necessarily unique for the existing finger kinematics. Therefore we deal with the differential kinematics which makes the computation of the *inverse* much easier. From (2.1) one obtains the differential transformations

$$\dot{\mathbf{x}}(t) = J(\mathbf{q})\dot{\mathbf{q}}; \qquad \dot{\mathbf{q}}(t) = J^{+}(\mathbf{q})\dot{\mathbf{x}} \tag{2.2}$$

where $J(\mathbf{q}) = \frac{\partial \mathbf{q}}{\partial \mathbf{x}}$ is the Jacobian and $J^{+}(\mathbf{q})$ is the pseudo inverse Jacobian. Assuming $\mathbf{x}(t)$ or $\dot{\mathbf{x}}(t)$ to be given from the task, i.e. from captured human demonstrations, the inverse kinematics in (2.2) remains to be computed. In order to avoid the time-consuming calculation of the inverse Jacobian at every time step the inverse differential kinematics is approximated by a TS fuzzy model

$$\dot{\mathbf{q}}(t) = \sum_{i=1}^{c_x} w_i(\mathbf{x}) J_{inv,i}(\mathbf{x}_i) \cdot \dot{\mathbf{x}} \tag{2.3}$$

where $w_i(\mathbf{x}) \in [0, 1]$ is the degree of membership of the vector $\mathbf{x}$ to a cluster C_{xi} with the cluster center $\mathbf{x}_i$. $J_{inv,i}(\mathbf{x}_i)$ are the inverse Jacobians in the cluster centers $\mathbf{x}_i$. c_x is the number of clusters. Due to the errors $\Delta\mathbf{x} = \mathbf{x}(t) - \mathbf{x}_m(t)$ between the desired position $\mathbf{x}(t)$ and the position $\mathbf{x}_m$ computed by the forward kinematics a correction of the angles is calculated via the analytical forward kinematics $\mathbf{x}_m(t) = \mathbf{f}(\mathbf{q}(t))$ of the finger. This changes (2.3) into

$$\dot{\mathbf{q}}(t) = \sum_{i=1}^{c_x} w_i(\mathbf{x}) J_{inv,i}(\mathbf{x}_i) \cdot (\dot{\mathbf{x}} + K \cdot (\mathbf{x}(t) - \mathbf{x}_m(t))). \tag{2.4}$$

It has to be emphasized that the correction or optimization loop using the forward kinematics $\mathbf{f}(\mathbf{q}(t))$ is started at every new time instant and stops until either a lower bound $\|\Delta\mathbf{x}\| < \varepsilon$ is reached or a given number of optimization steps is executed. The gain K has to be determined so that the optimization loop is stable. This TS-modeling is based on a clustering algorithm whose steps are described in the next section in more detail. The degree of membership $w_i(\mathbf{x})$ of an input vector $\mathbf{x}$ belonging to a cluster C_{xi} is defined by a bell-shape-like function

$$w_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c_x} \left(\frac{(\mathbf{x}-\mathbf{x}_i)^T M_{xi} (\mathbf{x}-\mathbf{x}_i)}{(\mathbf{x}-\mathbf{x}_j)^T M_{xj} (\mathbf{x}-\mathbf{x}_j)}\right)^{\frac{1}{\tilde{m}_x - 1}}} \tag{2.5}$$

M_{xi} define the induced matrices of the input clusters C_{xi}, $(i = 1 \ldots c_x)$, $\tilde{m}_x > 1$ determines the fuzziness of an individual cluster. The complexity of the on-line calculation of (2.4) is much lower than the complexity of (2.2) because (2.4) avoids the on-line calculation of numerous trigonometric functions. The time consuming clustering algorithm leading to the inverse Jacobians $J_{inv,i}$ is computed off-line.

2.4 Modeling of Grasp Primitives

2.4.1 Modeling by Time-Clustering

The recognition of a grasp type is achieved by a model that reflects the *behavior of the hand in time*.

In the following an approach to learning of human grasps from demonstrations by *time-clustering* [7] is shortly described. The result is a set of grasp models for a selected number of human grasp motions. According to Sect. 2.2 experiments were performed in which time sequences for 15 different grasps were collected using a data glove with 18 sensors (see [11]).

Each demonstration has been 10 times repeated by the same test person to collect enough samples of every particular grasp. The time period for a single grasp is about 3 seconds. From those data models for each individual grasp have been developed using fuzzy clustering and Takagi-Sugeno fuzzy modeling [14]. We consider the time instants as model inputs and the 3 finger joint angles as model outputs. Let the angle trajectory of a finger be described by

$$\mathbf{q}(t) = \mathbf{f}(t) \tag{2.6}$$

where $\mathbf{q}(t) \in R^3$, $\mathbf{f} \in R^3$, and $t \in R^+$. Linearization of (2.6) at selected time points t_i yields

$$\mathbf{q}(t) = \mathbf{A}_i \cdot t + \mathbf{d}_i \tag{2.7}$$

where $\mathbf{A}_i = \frac{\Delta \mathbf{f}(t)}{\Delta t}|_{t_i} \in R^3$ and $\mathbf{d}_i = \mathbf{q}(t_i) - \frac{\Delta \mathbf{f}(t)}{\Delta t}|_{t_i} \cdot t_i \in R^3$. Using (2.7) as a local linear model one can express (2.6) in terms of a Takagi-Sugeno fuzzy model [15]

$$\mathbf{q}(t) = \sum_{i=1}^{c} w_i(t) \cdot (\mathbf{A}_i \cdot t + \mathbf{d}_i) \tag{2.8}$$

where $w_i(t) \in [0, 1]$ is the degree of membership of the 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 t be the time and $\mathbf{x} = [q_1, q_2, q_3]^T$ the finger angle coordinates. Then the general clustering and modeling steps are

- Choose an appropriate number c_t of local linear models (data clusters)
- Find c_t cluster centers $(t_i, q_{1i}, q_{2i}, q_{3i})$, $i = 1 \ldots c_t$, in the product space of the data quadruples (t, q_1, q_2, q_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 first into so-called Gustafson-Kessel clusters (GK) and then onto the input space [16]
- Calculate c_t local linear (affine) models (2.8) using the GK clusters from step 2.

The degree of membership $w_i(t)$ of an input data point t to an input cluster C_{ti} is determined by

$$w_i(t) = \frac{1}{\sum_{j=1}^{c_t} \left(\frac{(t-t_i)^T M_{ti} (t-t_i)}{(t-t_j)^T M_{tj} (t-t_j)} \right)^{\frac{1}{\bar{m}_t - 1}}}. \tag{2.9}$$

The projected cluster centers t_i and the induced matrices M_{ti} define the input clusters C_{ti} $(i = 1 \ldots c_t)$. The parameter $\tilde{m}_t > 1$ determines the fuzziness of an individual cluster. A detailed description of this very effective clustering method can be found in [14]. In this way for each of the 15 grasp primitives in Fig. 2.2 a TS-fuzzy model is generated. These so-called *model grasps* are used to identify demonstrated grasps from a test sequence of a given combination of grasps.

2.4.2 Training of Time Cluster Models Using New Data

A grasp model can be built in several ways

– A single user trains the grasp model by repeating the same grasp n times
– m users train the grasp model by repeating the same grasp n times

The 1st model is generated by the time sequences

$$[(t_1, t_2, \ldots, t_N)_1 \ldots (t_1, t_2, \ldots, t_M)_n]$$

and the finger angle sequences

$$[(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N)_1 \ldots (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_M)_n].$$

The 2nd model is generated by the time sequences

$$[((t_1, t_2, \ldots, t_N)_1^1 \ldots (t_1, t_2, \ldots, t_M)_n^1) \ldots ((t_1, t_2, \ldots, t_N)_1^m \ldots (t_1, t_2, \ldots, t_M)_n^m)]$$

and the finger angle sequences

$$[((\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N)_1^1 \ldots (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_M)_n^1) \ldots$$
$$((\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N)_1^m \ldots (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_M)_n^m)]$$

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 grasp 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 grasp type. Let for simplicity the old model be built by a time sequence $[t_1, t_2, \ldots, t_N]$ and a respective finger angle sequence

$$[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N].$$

The old model is then represented by the input cluster centers t_i and the output cluster centers $\mathbf{q}_i$ $(i = 1 \ldots c)$. It is also described by the parameters $\mathbf{A}_i$ and $\mathbf{d}_i$ of the local linear models. Let

$$[\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_M], [\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \ldots, \tilde{\mathbf{q}}_M]$$

be the new training data. A new model can be built by "chaining" the old and the new training data leading for the time sequences to

$$[t_1, t_2, \ldots, t_N, \tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_M]$$

and for the finger angle sequences to

$$[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N, \tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \ldots, \tilde{\mathbf{q}}_M].$$

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

$$[t_1, t_2, \ldots, t_N]$$

as inputs and the finger angles

$$[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N]$$

as outputs.

2.5 Recognition of Grasps—Three Methods

In the previous section we showed that TS fuzzy models can be successfully used for modeling and imitation of human grasp behaviors. Now, we will show that they can also be used for classification of grasps in data from recorded human demonstrations. If we just observe captured motions of a human arm while executing several grasp actions it is difficult to identify the exact moment when a grasp sequence starts and ends. Related research shows that this task can be solved efficiently only by fusion of additional information sources such as tactile sensing and vision (see [3] and [4]). Since the scope of this chapter is only the recognition we assume the segmentation already to be finished. In the following we present three different recognition methods all of them being based on the time clustering of human grasps [10]. The first method classifies a test grasp by comparing the time clusters of the test grasp and a set of model grasps. The second method uses fuzzy recognition rules for segmentation and recognition. The third method classifies a test grasp using HMM which are applied to the output cluster centers of the grasp models. It should be stressed that methods 1 and 2 are related with each other both of them using distances between fuzzy clusters for recognition. Method 3 is a completely different approach using a probabilistic approach for recognition and classification.

2.5.1 Recognition of Grasps Using the Distance Between Fuzzy Clusters

Let the model of each grasp have the same number of clusters $i = 1 \ldots c$ so that each duration T_l $(l = 1 \ldots L)$ of the l-th grasp is divided into $c - 1$ time intervals Δt_i, $i = 2 \ldots c$ of the same length. Let the grasps be executed in an environment

comparable with the modeled grasp in order to avoid calibration and re-scaling procedures. Furthermore let

$$\begin{aligned} V_{model\,l} &= [V_{index}, V_{middle}, V_{ring}, V_{pinkie}, V_{thumb}]_l, \\ V_{index\,l} &= [\mathbf{q}_1, \ldots, \mathbf{q}_i, \ldots, \mathbf{q}_c]_{index,l}, \\ &\vdots \\ V_{thumb\,l} &= [\mathbf{q}_1, \ldots, \mathbf{q}_i, \ldots, \mathbf{q}_c]_{thumb,l}, \\ \mathbf{q}_i &= [q_1, q_2, q_3]^T \end{aligned} \tag{2.10}$$

where matrix $V_{model\,l}$ includes the output cluster centers $\mathbf{q}_i$ of every finger for the l-th grasp model. $\mathbf{q}_i$ is the vector of joint angles of each finger.

A model of the grasp to be classified (the test grasp) is built by the matrix

$$V_{grasp} = [V_{index}, V_{middle}, V_{ring}, V_{pinkie}, V_{thumb}]_{grasp}. \tag{2.11}$$

A decision on the grasp is made by applying the Euclidean matrix norm

$$N_l = \| V_{model\,l} - V_{grasp} \|. \tag{2.12}$$

The unknown grasp is classified to the grasp model with the smallest norm $\min(N_l)$, $l = 1 \ldots L$ and the recognition of the grasp is finished.

2.5.2 Recognition Based on Qualitative Fuzzy Recognition Rules

The goal of this method is to recognize and classify all individual grasps types in a data sequence containing a combination of several grasps. That is, it also performs the segmentation of the data sequence. The identification of a grasp from a combination of grasps is based on a recognition model. This model is represented by a set of recognition rules using the model grasps mentioned in the last section. The generation of the recognition model is based on the following steps:

1. Computation of distance norms between a test grasp combination and the model grasps involved.
2. Computation of extrema along the sequence of distance norms.
3. Formulation of a set of fuzzy rules reflecting the relationship between the extrema of the distance norms and the model grasps.
4. Computation of a vector of similarity degrees between the model grasps and the grasp combination.

2.5.2.1 Distance Norms

Let, for example, $grasp_2$, $grasp_5$, $grasp_7$, $grasp_{10}$, $grasp_{14}$ be a combination of grasps taken from the list of grasps shown in Fig. 2.2. In the training phase a time series of these grasps is generated using the existing time series of the corresponding

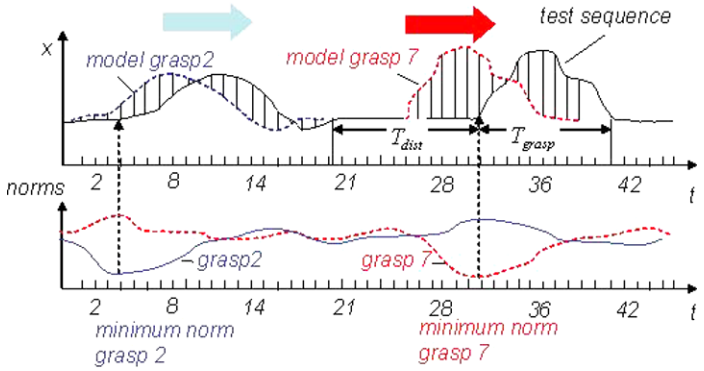

Fig. 2.5 Overlap principle

grasp models. Then each of the model grasps $i = 2, 5, 7, 10, 14$ is shifted along the time sequence of the grasp combination and compared with parts of it while taking the norm2 $\|\mathbf{Q}_{ci} - \mathbf{Q}_{mi}\|$ between the difference of the finger angles

$$\mathbf{Q}_{mi} = (\mathbf{q}_m(t_1), \ldots \mathbf{q}_m(t_{n_c}))^T{}_i$$

of a $grasp_i$ and the finger angles of the grasp combination

$$\mathbf{Q}_{ci} = (\mathbf{q}_{ci}(\tilde{t}_1), \ldots, \mathbf{q}_c(\tilde{t}_{n_c}))^T.$$

The vectors $\mathbf{q}_m$ and $\mathbf{q}_c$ include the 3 finger angles for each of the 5 fingers. Because of scaling reasons the norm of the difference is divided by the norm $\|\mathbf{Q}_{mi}\|$ of the model grasp. Then we obtain for the scaled norm

$$n_i = \frac{\|\mathbf{Q}_{ci} - \mathbf{Q}_{mi}\|}{\|\mathbf{Q}_{mi}\|} \tag{2.13}$$

where n_i are functions of time. With this for each grasp $i = 2, 5, 7, 10, 14$ a time sequence $n_i(t_1)$ is generated. Once the model grasp starts to overlap a grasp in the grasp combination, the norms n_i reach an extremum at the highest overlap which is either a minimum or a maximum (see Fig. 2.5).

2.5.2.2 Extrema in the Distance Norms and Segmentation

Using a model $grasp_i$ for comparison with a sequence of M test grasps the norm n_i forms individual patterns at M distinct time intervals of the norm sequence. Within each of these time intervals the norm sequence n_i reaches an extremum, i.e. either a local minimum or a maximum. In order to find the local extrema in n_i the total time interval T_{n_i} of n_i is partitioned into l time slices within which the search takes place (see Fig. 2.6). To be able to identify all relevant extrema, the lengths T_{slice} of the time slices have to be bounded by

$$T_{grasp,min} < T_{slice} < T_{dist,min}/2 \tag{2.14}$$

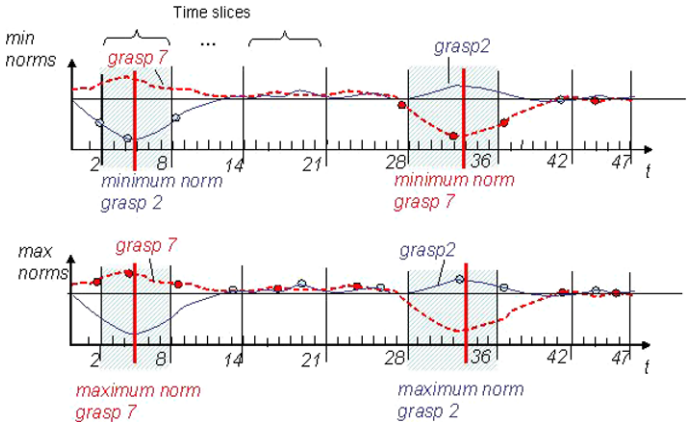

Fig. 2.6 Time slices

where $T_{grasp,min}$ is the minimum time length of a grasp. $T_{dist,min}$ is the minimum time distance between the end of a grasp and the starting point of a new grasp which is equal to the length of the pause. $T_{grasp,min}$ and $T_{dist,min}$ are supposed to be known. This search yields two pairs of vectors

$$\begin{aligned} \mathbf{z}_{mini} &= (z_{min,1i}, \ldots, z_{min,li})^T, \\ \mathbf{t}_{mini} &= (t_{min,1i}, \ldots, t_{min,li})^T \end{aligned} \tag{2.15}$$

and

$$\begin{aligned} \mathbf{z}_{maxi} &= (z_{max,1i}, \ldots, z_{max,li})^T, \\ \mathbf{t}_{maxi} &= (t_{max,1i}, \ldots, t_{max,li})^T \end{aligned} \tag{2.16}$$

where $l = \lceil T_{ni}/T_{slice} \rceil$. The elements of $\mathbf{z}_{mini}$ and $\mathbf{z}_{maxi}$ contain l absolute values of local minima and maxima of n_i, respectively. $\mathbf{t}_{mini}$ and $\mathbf{t}_{maxi}$ contain the corresponding l time stamps of the local minima and maxima. Usually there are more elements (extrema) included in (2.15) and (2.16) than grasps exist in the grasp sequence $l \geq M$. The segmentation task is to find the time slices that include the beginnings of grasps. To deal with an unknown number of grasps solutions different strategies are possible. A 'soft' solution requires a variable number of time clusters and a repetitive search for the most likely starting points of grasps. A mixed hardware-software solution is to utilize sensor information about the established contact between the fingers and the object to be grasped. In the following we assume the number M of grasps in a grasp sequence to be known. A segmentation procedure finds those extrema that indicate the starting points of the grasps. The segmentation is done by time clustering where the time vectors $\mathbf{t}_{maxi}$ and $\mathbf{t}_{mini}$ are the model inputs, $\mathbf{z}_{mini}$ and $\mathbf{z}_{maxi}$ are the model outputs. We expect the elements of $\mathbf{z}_{maxi}$ and $\mathbf{z}_{mini}$ to form M clusters $\mathbf{t}_{seg} = (t_{seg,1} \ldots t_{seg,M})^T$. The result of the clustering procedure is a vector of M time cluster centers pointing to the starting points of the grasps. For each time point $t_{seg,r}$ there is a pair $(z_{min,ij}, z_{max,ij})$. Index j denotes a $grasp_j$ in the

grasp sequence executed at time $t_{seg,r}$. Index $i = 2, 5, 7, 10, 14$ denote the model $grasps_i$. This finalizes the segmentation procedure.

2.5.2.3 Set of Fuzzy Rules

The two sets of vectors $\mathbf{z}_{min\,i}$ and $\mathbf{z}_{max\,i}$ build '*fingerprint patterns*' for each grasp in a specific grasp combination. On the basis of these patterns a set of rules decides whether a special combination of minima and maxima by consideration of their absolute values belong to a certain grasp or to another one. Obviously, for a selected grasp these patterns change with the change of a grasp combination. For example, the pattern for $grasp_2$ in the grasp combination 2, 5, 7, 10, 14 differs significantly from the pattern in grasp combination 1, 2, 3, 4, 5 etc. This is taken into account by the formulation of an individual set of rules for each grasp combination. In order to recognize a model $grasp_i$ from a specific grasp combination a set of 5 rules is formulated, one rule for each grasp in the combination.

A general recognition rule for $grasp_i$ to be identified from the combination reads:

$$\begin{aligned} &\text{IF } (n_j \text{ is } ex_{ji}) \text{ AND} \ldots \\ &\text{AND } (n_k \text{ is } ex_{ki}) \\ &\text{THEN } grasp \text{ is } grasp_i \end{aligned} \tag{2.17}$$

Rule (2.17), for example, can be read

"IF (norm n_2 of model $grasp_2$ is $max_{2,5}$)

...

AND (norm n_{14} of model $grasp_{14}$ is $max_{14,5}$)

THEN grasp is $grasp_5$".

The full rule to identify $grasp_5$ reads

$$\begin{aligned} &\text{IF } (n_2 \text{ is } max_{2,5}) \\ &\text{AND } (n_5 \text{ is } min_{5,5}) \\ &\text{AND } (n_7 \text{ is } max_{7,5}) \\ &\text{AND } (n_{10} \text{ is } min_{10,5}) \\ &\text{AND } (n_{14} \text{ is } max_{14,5}) \\ &\text{THEN } grasp \text{ is } grasp_5 \end{aligned} \tag{2.18}$$

$j = 2 \ldots k = 14$ are the indexes of the grasps in the grasp combination. i is the index of $grasp_i$ to be identified. ex_{ji} indicate fuzzy sets of local extrema which can be either minima min_{ji} or maxima max_{ji}. Extrema appear at the time points $\tilde{t} = t_j$ at which *model* $grasp_i$ meets $grasp_j$ in the grasp combination with a maximum overlap.

Let the total extremum $z_{ex_{tot}}$ either be a total minimum $z_{min_{tot}}$ or a maximum $z_{max_{tot}}$ over all 5 rules and all time slices (see Fig. 2.7)

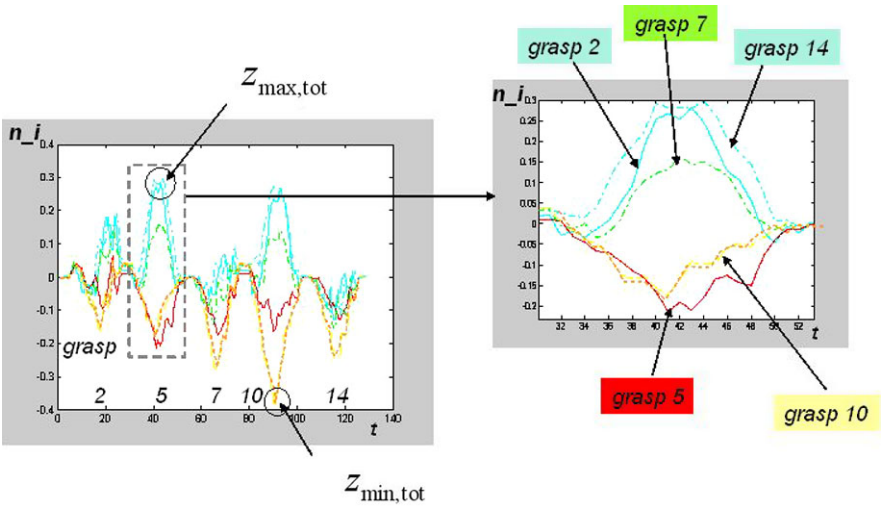

Fig. 2.7 Norms of a grasp sequence

$$z_{min_{tot}} = \min(z_{j_{min,i}}), \qquad z_{max_{tot}} = \max(z_{j_{max,i}}),$$
$$j = 1, \ldots, l; \; i = 2, 5, 7, 10, 14. \tag{2.19}$$

Then a local extremum $z_{j_{ex,i}}$ can be expressed by the total extremum $z_{ex_{tot}}$ and a weight $w_{ji} \in [0, 1]$

$$z_{j_{ex,i}} = w_{ji} \cdot z_{ex_{tot}}, \qquad z_{j_{min,i}} = w_{ji} \cdot z_{min_{tot}},$$
$$z_{j_{max,i}} = w_{ji} \cdot z_{max_{tot}}, \qquad j, i = 2, 5, 7, 10, 14. \tag{2.20}$$

2.5.2.4 Similarity Degrees

The special form of data requires the design of a specific similarity degree and a regarding membership function. From the time plots in Fig. 2.7 of the norms n_i for the training sets the analytical form of (2.18) for the identification of $grasp_i$ is chosen as follows

$$\mathbf{a}_i = \prod_j \mathbf{m}_{ji}, \quad j = 2, 5, 7, 10, 14,$$
$$\mathbf{m}_{ji} = \mathrm{Exp}(-|w_{ji} \cdot \mathbf{z}_{ex_{tot}} - \mathbf{z}_{ex\,i}|), \; \mathbf{z}_{ex\,i} = (z_{1ex,i}, \ldots, z_{lex,i})^T, \tag{2.21}$$

$\mathbf{a}_i = (a_{1,i}, \ldots, a_{l,i})^T$; $a_{m,i} \in [0, 1]$, $m = 1 \ldots l$ is a vector of *similarity degrees* between the *model* $grasp_i$ and the individual grasps 2, 5, 7, 10, 14 in the grasp combination at the time point t_m. The vector $\mathbf{z}_{ex\,i}$ represents either the vector of minima $\mathbf{z}_{min\,i}$ or maxima $\mathbf{z}_{max\,i}$ of the norms n_i, respectively.

The product operation in (2.21) represents the AND-operation in the rules (2.18). The exponential function in (2.21) is a *membership function* specially designed for

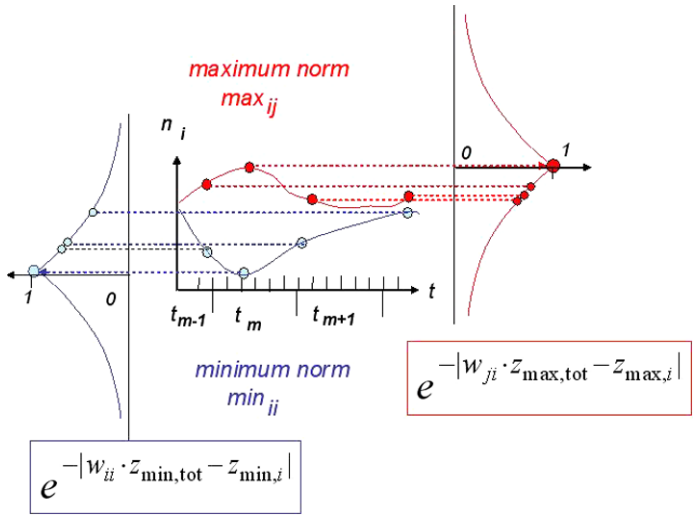

Fig. 2.8 Grasp membership functions

the indication of the distance of a norm n_i to a local extremum $w_{ji} \cdot z_{ex_{tot}}$. With this the exponential function reaches its maximum at exactly that time point t_m when $grasp_i$ in the grasp combination has its local extremum (see, e.g., Fig. 2.8).

If, for example, $grasp_5$ occurs at the time point t_m in the grasp combination then we obtain for $a_{m,5} = 1$. All the other grasps lead to smaller values of $a_{k,i}$, $k = 2, 7, 10, 14$. With this the type of grasp is identified and the grasp recognition is finished.

2.5.3 Recognition Based on Time-Cluster Models and HMM

The task is to classify an observation sequence of a test grasp given a set of observation sequences of model grasps using HMM. The HMM used here are of discrete nature which requires the formulation of a of number discrete states and discrete observations. One condition for the use of HMM in our approach is that all model grasps and the test grasp to be recognized are modeled by time-clustering described before.

The elements of a discrete HMM can be described in the compact notation [17]

$$\lambda = (A, B, \pi, N, M) \tag{2.22}$$

where N is the number of states S, M is the number of observations O, $A = \{a_{ij}\}$ is the matrix of state transition probabilities, $B = \{b_{jk}\}$ is the observation symbol probability of symbol O_k in state j, π is the initial state distribution vector. As an example, Figs. 2.9 and 2.10 show for grasp 1 (cylinder, only *close operation*) the graphs of the initially chosen state transitions $\{a_{ij}\}$ and the state transitions after

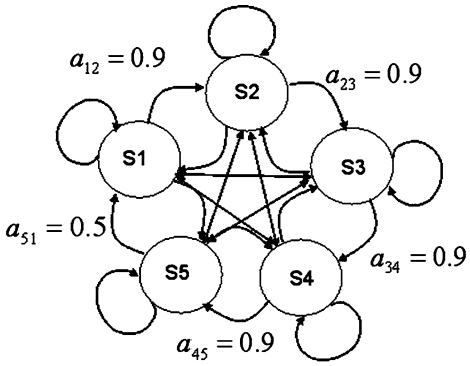

Fig. 2.9 Initial state transitions

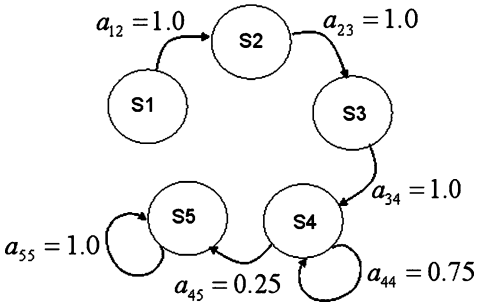

Fig. 2.10 Computed state transitions

the computation via HMM, respectively. Connections in Fig. 2.9 without explicit transition probabilities are denoted as $a_{ij} = 0.1$. Observe that after the computation most of the connections in the initial graph can be cut because of $a_{ij} = 0$.

To prepare the HMM for the recognition a number of steps has to be done:

Step 1: Determine a number N of states S. The states need not necessarily to be directly connected with a physical meaning, but it is of high advantage to do so. Therefore, $M = 5$ states are chosen getting the following labels:

state $S1$: open hand
state $S2$: half open hand
state $S3$: middle position
state $S4$: half closed hand
state $S5$: closed hand

Step 2: Generate a number M of discrete observations O. To generate discrete observations one has to deal first with the continuous observations, meaning the output cluster centers in V_{grasp} and the corresponding joint angles $\mathbf{q}_i$. It should be mentioned that the clustering process leads to vectors of cluster centers whose elements are, although being 'labeled' by a time stamp, not sorted in an increasing order of time. Since clustering of several grasps is done independently of each other the orders of time stamps of the cluster centers are in general different. This makes a comparison of test clusters V_{grasp} and model clusters V_{model} impossible. Therefore after time-clustering has been performed the output clusters have to be sorted in an

increasing order of time. In the following, cluster centers are assumed to be sorted in that way. Next, one has to transform the continuous output cluster centers $V_{model}(i)$, $i = 1 \ldots 10$ of the model into discrete numbers or 'labels'. If one would attach each cluster center an individual label one would obtain $M = 10 \times 15 = 150$ observation labels, 10—number of clusters, 15—number of grasps. This number of observations is unnecessarily high because some of the cluster centers form almost the same hand poses. Therefore two observations are reserved for the starting pose and end pose of all grasps since it can be assumed that every grasp starts and ends with nearly the same pose. Then, three poses for each grasp are chosen at the cluster numbers $(3, 5, 6)$ which makes $M = 3 \times 15 + 2 = 47$ observations. The result is obviously a set of possible observations labeled by the numbers $1 \ldots 47$ representing 47 poses of 15 time-clustering models of grasps. In order to label a specific pose of a given grasp one finds the minimal norms

$$I_j(i) = \min(\| V_{grasp\,j}(i) - out_1 \|, \ldots, \| V_{grasp\,j}(i) - out_{47} \|), \quad i = 1 \ldots 10 \tag{2.23}$$

where $I_j(i) \in [1 \ldots 47]$ is the observation label, $i \in [1 \ldots 10]$ is the number of a time cluster for test grasp $j \in [1 \ldots 15]$, $O(k) = V_{model\,m}(l)$, $k \in [1 \ldots 47]$ is the k-th observation, $m \in [1 \ldots 15]$ is a corresponding model grasp, $l \in [2, 3, 5, 6, 9]$ is a corresponding number of a time cluster in model grasp m. This procedure is done for all model grasps V_{model} with the result of 15 sequences $I_j(i)$ of 10 observations each, and for the test grasp V_{grasp} to be recognized.

Step 3: Determine the initial matrices $A \in R^{M \times M}$, $B \in R^{N \times M}$ and the initial state distribution vector $\pi \in R^{1 \times N}$. Since in the experiments the hand always starts to move with almost the same pose and keeps on moving through the states defined above we can both estimate the initial matrices A, and B, and the initial state distribution vector π easily.

Step 4: Generate 15 observation sequences $O_{train} \in R^{10 \times 15}$ for the 15 model grasps according to *step 2*.

Step 5: Generate 1 observation sequence $O_{test} \in R^{10 \times 1}$ for the test grasp according to *step 2*.

Step 6: Train the HMM with every model sequence O_{train} separately using the iterative expectation-modification procedure (EM), also known as Baum-Welsch method. The training process evaluates a log-likelihood LL of the trained model during iteration and stops as the change of LL undergoes a certain threshold. Observe here that $LL \leq 0$.

Step 7: Classify the observation sequence O_{test} by evaluating the log-likelihood LL of the m-th trained HMM for a model grasp m given the test data O_{test}. In addition, the most probable sequence of states using the Viterbi algorithm is computed.

Step 8: Compute the most probable model grasp number m to be closest to the test model by computing $\max(LL_i)$, $i = 1 \ldots 15$. With step 8 the grasp recognition is completed.

2.6 Experiments and Simulations

In this section time clustering and fuzzy modeling results are presented first. Then an experimental evaluation of the three methods for grasp recognition follows together with a comparison of the three methods.

2.6.1 Time Clustering and Modeling

The choice of the numbers of clusters both for the fingertip models and for the inverse kinematics depend on the quality of the resulting TS fuzzy models. On the basis of a performance analysis for each grasp and finger, 10 fingertip position models with 10 cluster centers have been generated from collected data.

Furthermore, 3 inverse Jacobian models for each grasp primitive and finger with 3 cluster centers have been built which are 15 Jacobians to be computed off-line. Since there are 3 angles (q_1, q_2, q_3) and 3 fingertip coordinates (x, y, z) for a single finger the Jacobians and their inverses are 3×3 square matrices. The 3rd link of each finger (next to the fingertip) does not have a sensor in the data glove. Therefore the angle of this link gets a fixed value greater than zero so that neither ill-conditioned Jacobians nor their inverses can computationally occur. For the identification of inverse Jacobians small random noise excitation is added to the angles to prevent ill-conditioned Jacobians while modeling from data. The motion of a grasp lasts 3.3 s in the average which adds up to 33 timesteps $\Delta t = 0.1$ s. The time clustering procedure results in the cluster centers $t_i = 2.04, 5.43, 8.87, 12.30, 15.75, 19.19, 22.65, 26.09, 29.53, 32.94$ where the time labels are measured in steps of $\Delta t = 0.1$ s. The time cluster centers are then complemented by the corresponding cluster centers for the x, y, z coordinates of the fingertips. This equidistant spacing can be found for every individual grasp primitive as a result of the time clustering. Figures 2.11, 2.12, 2.13, 2.14 and 2.15 shows modeling results for grasp 10 (plane (1 CD-ROM)) for the x,

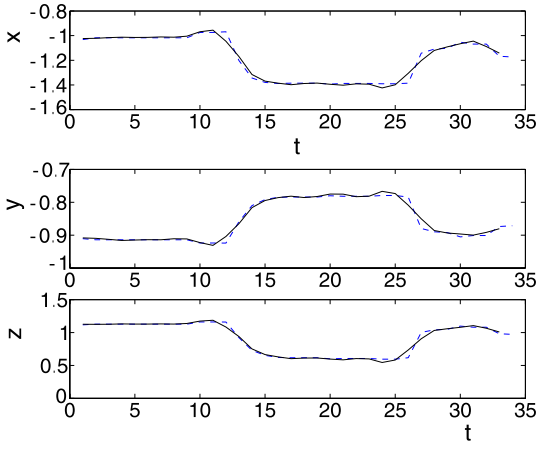

Fig. 2.11 Index finger, original:*solid*, model:*dashed*

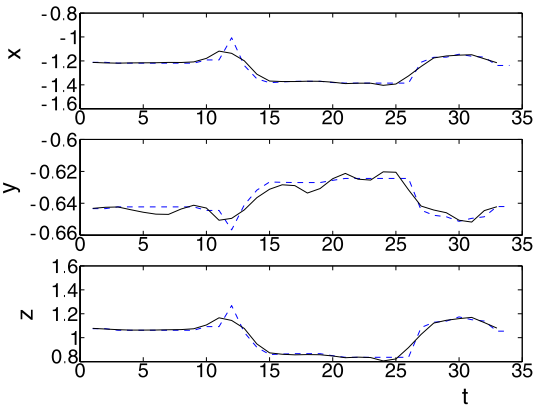

Fig. 2.12 Middle finger, original:*solid*, model:*dashed*

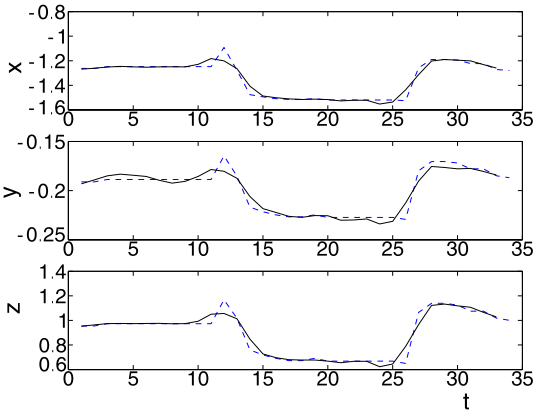

Fig. 2.13 Ring finger, original:*solid*, model:*dashed*

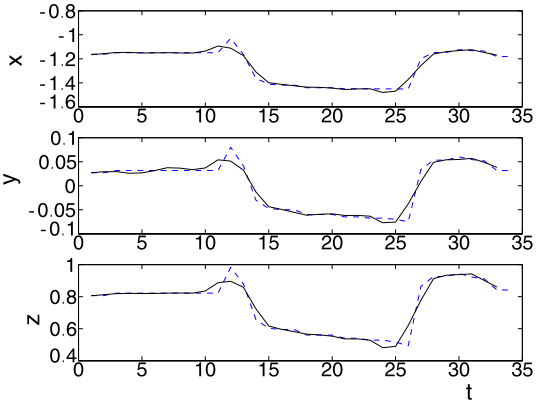

Fig. 2.14 Pinkie finger, original:*solid*, model:*dashed*

y, and z coordinates for the index, middle, ring, and pinkie finger plus the thumb. These results show a good or even excellent modeling quality.

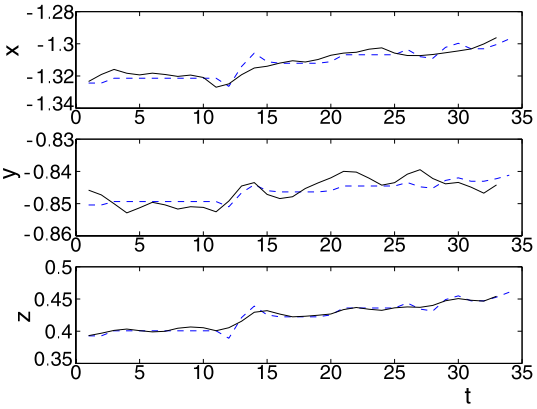

Fig. 2.15 Thumb finger, original:*solid*, model:*dashed*

2.6.2 Grasp Segmentation and Recognition

In this section an experimental evaluation of the three methods for grasp recognition is presented and a comparison of the methods is made. 10 test grasps for each of the 15 different grasp primitives have been tested according to Sect. 2.2. A grasp starts with an open hand and is completed when the fingers establish contact with the object. The experimental results are divided into 3 groups of recognition rates:

1. grasps with a recognition rate $\geq 75\%$
2. grasps with a recognition rate $< 75\% - \geq 50\%$
3. grasps with a recognition rate $< 50\%$.

In the following, the recognition rates of the three discussed methods are listed in Tables 2.1, 2.2, and 2.3. The experimental results confirm the assumption that distinct grasps can be discriminated quite well from each other while the discrimination between similar grasps is difficult. Therefore, merging of similar grasps and building of larger classes can improve the recognition process significantly. Examples of such classes are grasps (4, 5, 15), grasps (10, 11), and grasp (8, 9).

Table 2.1 shows the recognition rates for method 1.

The 1st group with a recognition rate $\geq 75\%$ is the largest one where 4 of 7 grasps show a recognition rate 100%. It follow the equally large groups 2 and 3. Table 2.2 shows the recognition rates for method 2. In this experiment 12 grasp combinations of 5 grasps each have been tested. It could be shown that grasps with distinct maxima and minima in their n_i patterns can be recognized better than grasps without this feature. Reliable grasps are also robust against variations in the time span of an unknown test grasp compared to the time span of the respective model grasp. Our results show that this method can handle a temporal difference up to 20%. By temporal difference we mean the difference in the length of the test grasp and the respective model grasp. The 1st group with a recognition rate $\geq 75\%$ is again the largest one where 3 of 8 grasps show a recognition rate 100%.

Table 2.3 shows the recognition rates for method 3. The 2nd group is here the largest one with a recognition rate $< 75\%$, $\geq 50\%$ followed by the 1st group where

Table 2.1 Recognition rates, method 1

Class	Grasp	Percentage
$\geq 75\%$	4. Hammer	100%
	8. Precision. grasp sphere	87%
	10. Small plane	100%
	11. Big plane	85%
	12. Fingertip small ball	100%
	14. Fingertip can	100%
	15. Penholder grip	85%
$< 75\%, \geq 50\%$	1. Cylinder	71%
	2. Big bottle	57%
	3. Small bottle	57%
	13. Fingertip big ball	71%
$< 50\%$	5. Screwdriver	0%
	6. Small ball	14%
	7. Big ball	28%
	9. Precision grasp cube	42%

Table 2.2 Recognition rates, method 2

Class	Grasp	Percentage
$\geq 75\%$	4. Hammer	100%
	5. Screwdriver	93%
	8. Precision. grasp sphere	80%
	9. Precision grasp cube	100%
	10. Small plane	100%
	11. Big plane	88%
	13. Fingertip big ball	75%
	15. Penholder grip	75%
$< 75\%, \geq 50\%$	1. Cylinder	55%
	2. Big bottle	60%
	3. Small bottle	66%
	6. Small ball	55%
$< 50\%$	7. Big ball	16%
	12. Fingertip small ball	33%
	14. Fingertip can	33%

2 of 5 grasps show a recognition rate 100%, and by the 3rd one. For more than half of the grasp primitives all three methods provide similar results. This is true for the grasps 1, 2, 3, 4, 7, 8, 10, and 15. However similarities between grasps may give space for misinterpretations which explains the low percentages for some grasps e.g. grasps 5 and 9 in method 1 or grasps 5 and 8 in method 3. Looking at groups 1,

Table 2.3 Recognition rates, method 3

Class	Grasp	Percentage
$\geq 75\%$	4. Hammer	100%
	9. Precision grasp cube	85%
	10. Small plane	85%
	14. Fingertip can	85%
	15. Penholder grip	100%
$< 75\%, \geq 50\%$	1. Cylinder	71%
	2. Big bottle	57%
	3. Small bottle	71%
	5. Screwdriver	71%
	6. Small ball	57%
	11. Big plane	57%
	12. fingertip small ball	57%
	13. Fingertip big ball	71%
$< 50\%$	7. Big ball	0%
	8. Precision. grasp sphere	28%

method 1 is the most successful one which is also a solution with the easiest implementation. Then it follows method 2 with a quite high implementation effort and finally method 3 based on HMM. It should be stated that the HMM principle may allow some improvement of the results especially in the case of an extended sensory suit in the experimental setup.

2.7 Conclusions

The goal of grasp recognition is to develop an easy way of 'programming by demonstration' of grasps for a humanoid robotic arm. In this chapter, three different methods of grasp recognition are presented. Grasp primitives are captured by a data glove and modeled by TS-fuzzy models. Fuzzy clustering and modeling of time and space data are applied to the modeling of the finger joint angle trajectories of grasp primitives. The 1st method being the simplest one classifies a human grasp by computing the minimum distance between the time-clusters of the test grasp and a set of model grasps. In the 2nd method a qualitative fuzzy model is developed with the help of which both the segmentation and grasp recognition can be achieved. The 3rd method uses Hidden Markov Models (HMM) for grasp recognition. A comparison of the three methods showed that the 1st method is the most effective one, followed by the 2nd and the 3rd method. In order to achieve a further increase of the recognition rates methods 1 and 2 could be combined because of their close relationship whereas method 3 is only connected with methods 1 and 2 via the time cluster modeling of the grasps. Therefore, the HMM principle may lead to better results using

more haptic sensors in the experimental setup. To improve the PbD-process in general, all 3 methods will be further developed for the recognition and classification of operator motions in a robotic environment using more sensor information about the robot workspace and the objects to be handled.

References

1. Kang, S.B., Ikeuchi, K.: Towards automatic robot instruction form perception—mapping human grasps to manipulator grasps. IEEE Trans. Robot. Autom. **13**(1), 81–95 (1997)
2. Zoellner, R., Rogalla, O., Dillmann, R., Zoellner, J.: Dynamic grasp recognition within the framework of programming by demonstration. In: Proceedings Robot and Human Interactive Communication, 10th IEEE International Workshop, Bordeaux, Paris, France, 18–21 September 2001. IEEE, New York (2001)
3. Bernardin, K., Ogawara, K., Ikeuchi, K., Dillman, R.: A sensor fusion approach for recognizing continuous human grasp sequences using hidden Markov models. IEEE Trans. Robot. **21**(1), 47–57 (2005)
4. Ekvall, S., Kragic, D.: Grasp recognition for programming by demonstration. In: Proceedings of International Conference on Robotics and Automation, ICRA 2005, Barcelona, Spain (2005)
5. Li, C., Khan, L., Prabhaharan, B.: Real-time classification of variable length multi-attribute motions. Knowl. Inf. Syst. (2005). doi:10.1007/s10115-005-0223-8
6. Aleotti, J., Caselli, S.: Grasp recognition in virtual reality for robot pregrasp planning by demonstration. In: International Conference on Robotics and Automation, ICRA 2006, Orlando, FL, USA, May 2006. IEEE, New York (2006)
7. Palm, R., Iliev, B.: 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, 16–21 July 2006. IEEE, New York (2006)
8. Palm, R., Iliev, B.: 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, 23–26 July 2007. IEEE, New York (2007)
9. Palm, R., Iliev, B., Kadmiry, B.: Recognition of human grasps by time-clustering and fuzzy modeling. Robot. Auton. Syst. **57**(5), 484–495 (2009)
10. Palm, R., Iliev, B.: 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, 1–5 July 2008. IEEE, New York (2008)
11. Asada, H.H., Fortier, J.: Task recognition and human-machine coordination through the use of an instrument-glove. Progress report No. 2-5, pp. 1–39, March 2000
12. Cutkosky, M.: On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Trans. Robot. Autom. **5**(3), 269–279 (1989)
13. Iberall, T.: Human prehension and dexterous robot hands. Int. J. Robot. Res. **16**, 285–299 (1997)
14. Palm, R., Stutz, C.: Open loop dynamic trajectory generator for a fuzzy gain scheduler. Eng. Appl. Artif. Intell. **16**, 213–225 (2003)
15. Takagi, T., Sugeno, M.: Identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **SMC-15**(1), 116–132 (1985)
16. Gustafson, D., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: Proceedings of the 1979 IEEE CDC, pp. 761–766 (1979)
17. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)

http://www.springer.com/978-1-84996-328-2

Robot Intelligence
An Advanced Knowledge Processing Approach
Liu, H.; Gu, D.; Howlett, R.J.; Liu, Y. (Eds.)
2010, XIV, 294 p., Hardcover
ISBN: 978-1-84996-328-2