

2005

Manipulation skills acquisition through state classification and dimension decrease

S. Dong

University of Wollongong, shen@uow.edu.au

F. Naghdy

University of Wollongong, fazel@uow.edu.au

Publication Details

This article was originally published as: Dong, S & Naghdy, F, Manipulation skills acquisition through state classification and dimension decrease, 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 05), 14-16 Nov. 2005, 392-396. Copyright IEEE 2005.

Manipulation skills acquisition through state classification and dimension decrease

Abstract

The work carried out to explore the feasibility of reconstructing human constrained motion manipulation skills is reported. This is achieved by tracing and learning the manipulation performed by a human operator in a haptic rendered virtual environment. The peg-in-hole insertion problem is used as a case study. In the developed system, force and position variables generated in the haptic rendered virtual environment combined with a priori knowledge about the task are used to identify and learn the skills in the newly demonstrated task. The data obtained from the virtual environment is classified into different cluster sets using Fuzzy Gustafson-Kessel Model (FGK). Principal Component Analysis (PCA) is applied to each cluster to reduce the dimension of the data. The clusters in the optimum cluster set are tuned using Locally Weighted Regression (LWR) to produce prediction models for robot trajectory performing the physical assembly based on the force/position information received from the rig.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Dong, S & Naghdy, F, Manipulation skills acquisition through state classification and dimension decrease, 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 05), 14-16 Nov. 2005, 392-396. Copyright IEEE 2005.

Manipulation Skills Acquisition through State Classification and Dimension Decrease

Shen Dong*
Fazel Naghdy*

(*) *School of Electrical, Computer and Telecommunication Engineering,
University of Wollongong, NSW, 2500, Australia*

Abstract

The work carried out to explore the feasibility of reconstructing human constrained motion manipulation skills is reported. This is achieved by tracing and learning the manipulation performed by a human operator in a haptic rendered virtual environment. The peg-in-hole insertion problem is used as a case study. In the developed system, force and position variables generated in the haptic rendered virtual environment combined with a priori knowledge about the task are used to identify and learn the skills in the newly demonstrated task. The data obtained from the virtual environment is classified into different cluster sets using Fuzzy Gustafson–Kessel Model (FGK). Principal Component Analysis (PCA) is applied to each cluster to reduce the dimension of the data. The clusters in the optimum cluster set are tuned using Locally Weighted Regression (LWR) to produce prediction models for robot trajectory performing the physical assembly based on the force/position information received from the rig.

1. Introduction

The effectiveness of computer simulation can be augmented using haptic rendering. A haptic interface, or force feedback device increases the quality of human-computer interaction by accommodating the sense of touch in computer simulation. It provides an attractive augmentation to visual display and significantly enhances the level of immersion in a virtual environment.

In this work, haptic rendered virtual modeling is used as part of a new paradigm for programming robotics manipulator to perform complex constrained motion tasks. The peg-in-hole assembly process is used as a platform to study the concept, as it concisely

represents a constrained motion force sensitive manufacturing task with all the attendant issues of jamming, tight clearances, and the need for quick assembly times, reliably, etc. Human operator demonstrates both good and bad examples of the desired behaviour in the haptic virtual environment. Position and contact force and torque data as well as orientation generated in the virtual environment combined with a priori knowledge about the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The robot evaluates the controller's performance and thus learns the best way to produce that behaviour.

The data obtained from the virtual environment is classified into different cluster sets using Fuzzy Gustafson–Kessel Model. Each cluster represents a contact state between the peg and the hole. The data in the clusters consists of fourteen different parameters including force and position variables. In order to reduce the dimension of the data, Principal Component Analysis (PCA) is applied to each cluster. PCA is a classical statistical method usually used to find patterns in high dimensional data. The method has been successfully applied in the past to face recognition and image compression. A reduced dimension and high feature related vector is simpler to interpret and understand than the original raw data. It will be easier and less time consuming for machine learning algorithms to learn human skills based on these highly focused and lower dimension data. The compression level is determined according to the error produced in the compression process.

The number of the cluster sets is determined based on minimum compression error and maximum dimension removed. The clusters in the optimum cluster set are tuned using Locally Weighted Regression (LWR) to produce prediction models for robot trajectory performing the physical assembly

based on the force/position information received from the rig.

The rest of the paper is organized as follows. Section 2 briefly provides the development platform employed in the work to perform the peg-in-hole insertion with a 6 DOF haptic device. The assembly state classification training by Fuzzy Gustafson–Kessel Model and dimension reducing by Principal Component Analysis is then described. This will be followed by the employment of the locally weighted regression (LWR) on skill acquisition from the virtual environment in Section 3. Physical manipulation results are presented in Section 4. Finally, conclusion of the work and suggestions for future work are given in Section 5.

2. Haptic rendered virtual environment

The data used by the robot to acquire basic manipulation skills is generated from a haptic rendered virtual environment. The haptic rendered virtual manipulation environment consists of a six degree-of-freedom (6DOF) haptic device PHANToM Premium 1.5 [1] and Reachin API [2] along with VRML and Python [2], which are used to construct the virtual haptic manipulation environment of the peg-in-hole insertion.

Figure 1 shows the developed virtual environment. The peg is a dynamic rigid object in the haptic rendered virtual environment. The force and torque reacted to the peg are transferred to PHANToM Premium 1.5 through the spring damper system. The hole is static in the environment while the peg can be translated and rotated.



Figure 1. 6DOF Peg-in-hole haptic virtual environment

The haptic rendered model of the peg-in-hole assembly insertion generating force and torque data is constructed using the virtual proxy method [3]. The

virtual rigid peg is defined as a virtual proxy and controlled by the physical ReachIn probe in the haptic rendered virtual environment. The position of the virtual proxy is changed according to alteration in the probe's position.

Figure 2 shows the position, force and torque data as well as change in the rotation angle from the last step to the current, obtained from the haptic rendered virtual environment.

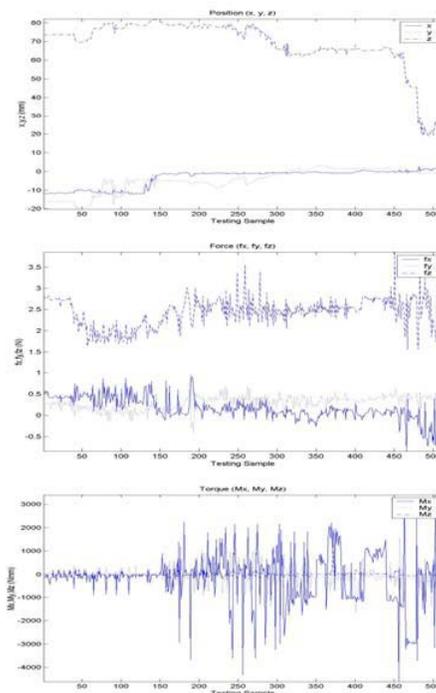


Figure 2. The training data obtained from the virtual environment

3. Skill acquisition

Fuzzy Gustafson–Kessel algorithm (FGK) is applied to the trajectory associated with the manipulation performed by the operator in the haptic virtual environment. The algorithm classifies the trajectory into a number of clusters, each representing a state. In the first stage of the process the constraints on the state variables are learned. These constraints determine the corresponding desired state and choose the best trajectory to the next state. The Fuzzy Gustafson–Kessel classifier, obtained by training the data from the haptic rendered virtual environment, will be used to estimate assembly states in on-line analysis. Principal Component Analysis (PCA) is then applied to each cluster for dimension reduction and feature extraction. The number of clusters is determined based

on a flatness index of the clusters combined with PCA compression rate [10]. Actions that determine the best trajectory from current state to next state are computed using knowledge of the system dynamics, acquired through Locally Weighted Regression (LWR) method and encoded as the on-line nonlinear function approximator for the trajectories.

Figure 3 illustrates the skill acquisition procedure. The first row represents the offline training procedure. 30 group of data obtained from the haptic virtual environment are used in Fuzzy Gustafson–Kessel classifier training. Thin lines represent the flow of data in the training progress. The second row represents the online data analysis. Thick lines represent the sensing data in the physical assembly procedure from the experimental rig. Dotted lines represent the use of previously trained module.

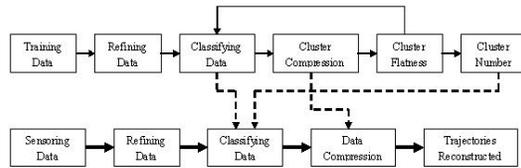


Figure 3. Skill acquisition procedure

3.1 Data Refinement

The noise in the data is removed before analysis. The following algorithm [4] is applied to refine the data, in which u is the action vector:

1. Remove irrelevant actions that do not enhance an action. This is defined by $\|u\| \leq \delta_s, \delta_s \geq 0$, where δ_s is an application specific threshold.
2. Remove the operator's rough control. If the differences between two continuous actions are too large, their average value will be used instead.

3.2 State Classification by FGK

The Fuzzy Gustafson-Kessel clustering algorithm has been widely used for fuzzy data analysis and pattern recognition [7], because it is flexible in automatically determining the optimal hyper-ellipsoidal cluster from the training data set. This algorithm uses statistic analysis of the data generated from haptic rendered virtual environment, and divides them into groups according to their common characteristic.

A separate norm matrix M_i is calculated to determine the shape of the clusters. These norm matrices are updated together with the centres of the

corresponding clusters. Therefore the prototypes of the clusters are a pair (v_i, F_i) .

$$M_i = (\det F_i)^{1/n} F_i^{-1} \quad (1)$$

where v_i is the centre of the cluster and F_i is the covariance matrix, which defines the shapes of the clusters. v_i and F_i are defined in equation 4.

The FGK computes the distance to the prototypes as:

$$d^2(z_j, v_i) = (z_j - v_i)^T M_i (z_j - v_i) \quad (2)$$

where z_j represents each data set.

FGK searches for the partition matrix and the cluster prototypes in order to minimize the following function:

$$J(Z, V, U) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m d^2(z_j, v_i) \quad (3)$$

where u_{ij} are membership degrees of the data vector z_j in the i th cluster.

The prototypes are updated according to the following equations [8]:

$$v_i = \frac{1}{N_i} \sum_{j=1}^N u_{ij}^m z_j, \quad F_i = \frac{1}{N_i} \sum_{j=1}^N u_{ij}^m (z_j - v_i)(z_j - v_i)^T \quad (4)$$

3.3 Dimension Decrease by PCA

The dimension of the recorded data from the haptic rendered virtual environment is large, but the components of the data are highly redundant. It is useful in this situation to reduce the dimension of the input vectors. Principal Component Analysis (PCA) is an effective procedure for performing this operation. It generates the following outcomes:

It orthogonalizes the components of the input vectors so that they are uncorrelated with each other.

It orders the resulting orthogonal components, principal components, so that those with the largest variation come first.

It eliminates those components that contribute the least to the variation in the dataset.

Principal Component Analysis is based on the statistical representation of a random variable. The aim is to find a set of M orthogonal vectors in data space that account for as much data variance as possible. Projecting the data from their original N -dimensional space onto the M -dimensional subspace spanned by these vectors results in a dimensionality reduction that often retains most of the intrinsic information in the data [9].

Given a set of training vectors x from the n dimensional input space \mathbb{R}^n ,

$X(t_k) = (x_1(t_k), x_2(t_k), \dots, x_n(t_k))$, where t_k is the k th observation time.

The Principal Component Analysis looks for a $y_k(t_k)$, which is a linear combination of the components of $X(t_k)$

$y_1(t_k) = X(t_k) \times \lambda_1$, so that the approximation to X , $\hat{X} = y_1(t_k) \times \lambda_1$, $e = \{\|X - \hat{X}\|^2\}$, e is minimized.

The residuals $(\lambda_2, \dots, \lambda_n)$ can be found using the same method.

$\hat{X}(t_k) = \sum_{j=1}^c [X(t_k) \times \lambda_j] \times \lambda_j + \varepsilon_k$, where ε_k are the residuals.

3.4 Determining the Number of Clusters

The number of cluster must be specified before clustering. The higher the number of clusters, the finer will be the approximation of the nonlinear. This, however, but also more parameters with higher variance have to be estimated. If no prior knowledge on the number of clusters is available, then automatic procedures can be applied.

A new approach for automatic determining of the number of the clusters, based on a flatness index of the cluster [10] and Principal Component Analysis (PCA), has been developed in the this paper. In this approach, the eigenvalues are sorted of the cluster covariance matrix C_i in a descending order, $\lambda_{i,1} \geq \lambda_{i,2} \geq \dots \geq \lambda_{i,n}$.

When approximating the regression surface, the obtained hyperellipsoidal clusters are flat, i.e., one of the axes is much shorter than the others. Consequently, the smallest eigenvalues $\lambda_{i,n}$ is significantly smaller than the remaining ones. In order to efficiently approximate the regression surface by hyperplanes, the clusters should be as flat as possible. The flatness index t_i of a cluster is defined as a ratio between the smallest and the largest eigenvalue, that is $t_i = \lambda_{i,n} / \lambda_{i,1}$. An aggregate measure called the average cluster flatness is defined as $t_A = \frac{1}{c} \sum_{i=1}^c \frac{\lambda_{i,n}}{\lambda_{i,1}}$

The data cluster sets obtained from the Fuzzy Gustafson-Kessel Model consists of fourteen different parameters including force and position variables and could be more in the future research. In order to reduce the dimension of the data, Principal Component Analysis (PCA) is applied to each cluster. The compression level is determined according to the error produced in the compression process, that is $p_{i,n} = d_{i,n} / D_{i,n}$. A compression rate $p_{i,n}$ is defined by the ratio of the most significant eigenvalue number $d_{i,n}$ (or the remain dimension numbers after compression) the and the number of original eigenvalue $D_{i,n}$. The

eigenvalue number $d_{i,n}$ is decided by an error percentage e_i , $e_i = \{\|X_i - \hat{X}_i\|^2\}$, which is the Frobenius

norm between the compressed matrix and original data matrix. The error tolerance E must be set before the calculation of $p_{i,n}$. The number of clusters and the compression rate of each cluster is calculated by

$PT = \frac{1}{c} \sum_{i=1}^c (\frac{\lambda_{i,n}}{\lambda_{i,1}} \times p_i)$. The optimal values of $\lambda_{i,n}$ and $d_{i,n}$,

are found by minimising PT .

As a result, five clusters are determined. The compression rate p and the error percentage e for each cluster are shown in Table. 1.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
p (e< 10%)	28.6%	21.4%	35.7%	35.7%	28.6%

Table 1. Error percentage below 10% the maximum compression rate achieved

3.5 Trajectory Reconstruction

Locally Weighted Regression (LWR) method is encoded as the approximator for the trajectories in each state during physical manipulation. In most learning methods, a single global model is used to fit all the training data, while local models attempt to fit the training data only in a region around the location of the query point. Locally weighted regression is one of the methods, which uses a distance weighted regression to fit nearby points, giving them a high relevance. Locally weighted regression is a form of lazy and memory based learning, since it stores the training data in memory and finds relevant data from the database to answer a particular query point [5]. When a locally weighted linear model is computed, the stored data points are weighted according to the distance from the query point.

The recorded training data is classified into several groups according to contact states between the peg and the hole. Specific indexes are assigned to each group. The index speeds up the search process in the physical manipulation by looking for the actions associated with a particular state in a sub-training data set rather than the whole database.

The following issues are considered when locally weighted regression learning is applied [6]:

1. Distance function: the relevance between data points is measured using diagonally weighted Euclidean distance.

$$d_m(x, q) = \sqrt{\sum_j (m_j(x_j - q_j))^2} = \sqrt{(x - q)^T M^T M (x - q)} = d_E(M_x, M_q)$$

- m_j is the feature scaling factor for the j th dimension. M is a diagonal matrix with $M_{jj}=m_j$.
2. Separable criterion: the weights of training points are computed using Gaussian kernel, which has infinite extent:

$$K(d) = \exp(-d^2)$$
 3. Enough data: Sufficient data are needed to satisfy the statistical requirements.
 4. Labeled data: Each training data point should be linked to a specific output.
 5. Representations: Fixed length vectors are produced for a list of specified features.

4. Results

The developed algorithm was applied to an experimental rig consisting of a hole with two degrees of freedom (the pitch and yaw angles) controlled by two stepper motors, and a peg with one degree of freedom (the translation along the axis of the peg) controlled by a DC speed motor. These 3DOF are sufficient to study the insertion phase. The radius of the peg is 10mm and that of the hole is 10.05mm. The clearance between the peg and the hole is 0.05mm. The hole has 0.18° degree turning angle in each step.

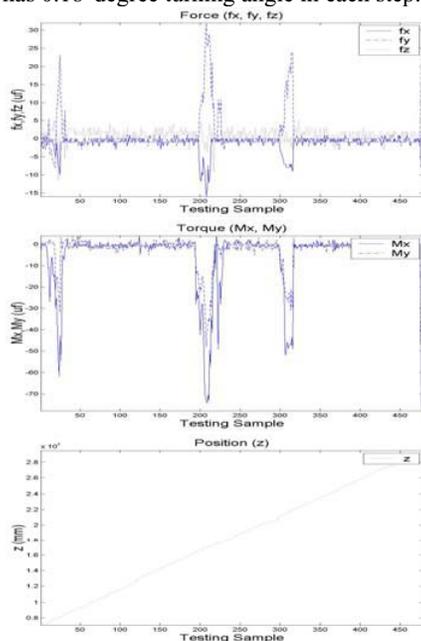


Figure 4. Experimental results

A typical performance of the approach is illustrated in Figure 4, which shows the forces: f_x, f_y, f_z , the torque around X, Y -axis: M_x, M_y and the position translation along Z -axis.

5. Conclusion

The development of a new paradigm for programming of robotics manipulators performing complex constrained motion tasks was reported in the paper. The manipulation skills are acquired from the demonstration of the task performed by a human operator in a haptic rendered virtual manipulation environment. The peg-in-hole assembly process is used as a platform to study the concept. The work explores how human psychomotor manipulation skills can be replicated by a machine.

6. References

- [1] SensAble Technologies, *GHOST SDK Programmer's Guide*, Cambridge MA, 1999
- [2] Reachin Technologies AB, *Programmer's Guide Reachin API 3.2*, 2003
- [3] Diego C. Ruspini, Krasimir Kolarov, Oussama Khatib, "The Haptic Display of Complex Graphical Environments", *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 345-352.
- [4] Kaiser, M. and Dillman, R., "Building Elementary Robot Skills from Human Demonstration," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, vol. 3, 1996, pp. 2700-2705.
- [5] Šuc, D. and Bratko, I., "Modelling of Control Skill by Qualitative Constraints", *Thirteenth International Workshop on Qualitative Reasoning*, Loch Awe, Scotland, 1999, pp. 212-220.
- [6] Atkeson, C. G., Moore, A.W., and Schaal, S., "Locally Weighted Learning", *Artificial Intelligence Review*, vol. 11, 1-5, 1997, pp. 11-73.
- [7] Babuska, R., *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1998.
- [8] Hoppner, F., Klawonn, F., Kruse, R., and Runkler, T., *Fuzzy Cluster Analysis*, Chichester, J. Wiley & Sons, 1999
- [9] Smith, L. I., "A tutorial on Principle Component Analysis", *Technical Report*, 2002, University of Otago, New Zealand.
- [10] Babuska, R. and Verbruggen, H. B., "New Approach to Constructing Fuzzy Relational Models From Data", *Proceeding of Third European Congress on Intelligent Techniques and Soft Computing*, Aschen, Germany, 1995, pp. 583-587.