Proc. 2002 Australasian Conference on Robotics and Automation
Auckland, 27-29 November 2002

Human-to-Robot Skill Transfer Through Haptic Rendered Environment

Y. Chen, and F. Naghdy
School of Electrical, Computer and Telecommunication Engineering,
University of Wollongong, NSW, 2522, Australia.
Fax : +61 42 21 3236.
{yc12,f.naghdy}@uow.edu.au

Abstract

A new paradigm for programming of robotics manipulator is being studied. The teaching of the machine will begin with the necessary skills being demonstrated by the human operator in a virtual environment with tactile sensing (haptics). Position and contact force/torque data generated in the virtual environment combined with a priori knowledge about the task will be used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The peg-in-hole insertion problem is used as a case study. The overall concept is described. The methodologies developed to build the virtual environment and to learn the basic skills are presented. The results obtained so far are reported.

1 Introduction

Robots are widely used as an automation tool to improve productivity in industry. Force sensitive manipulation is a generic requirement for a large number of industrial tasks, especially those associated with assembly. One of the major factors preventing greater use of robots in assembly tasks to date has been the lack of availability of fast reliable methods of programming robots to carry out such tasks. Hence robots have in practice been unable to economically replicate the complex force and torque sensitive capability of human operators.

Teaching a robot to perform a task by demonstrating the task carried out by a human operator is an ideal method of programming a manipulator. Employment of a haptic-rendered virtual environment to demonstrate the task simplifies the teaching by showing process, as the training data can be directly extracted from the haptic system [Mussa-Ivaldi, 1985]. Such approach takes advantage of recent developments in virtual reality and computer simulation. The data used by the machine to acquire basic manipulation skills is generated through a haptic-rendered virtual environment. This approach offers a number of advantages compared to other methods of obtaining training data, including:

1. The training data (e.g. velocities, angles, positions, forces and torque) can be extracted and recorded directly which simplifies the data collection process [Mussa-Ivaldi, 1985].
2. The environment can be easily modified and changed as the manipulation process and its requirements are changed.
3. The risk of breakdown and breakage of the system is very low.
4. Dangerous and costly environments can be easily constructed and simulated.
5. A user-friendly environment for the human operator can be developed.

This approach is studied in the work reported in this paper. The concepts and methodologies are developed in the context of the peg-in-hole insertion process which represents a typical manipulation task in automatic assembly. However, the algorithms developed for the system are as generic as possible. The developed algorithms acquire the basic manipulation skills performed in an automatic assembly and translate them to trajectories and task schedules for a particular application and robotics manipulator.

2 Peg-in-Hole Insertion Virtual Environment

The haptic rendering in the virtual environment is provided through a 3 degrees of freedom (DOF) generic device called Phantom manufactured by Sensable. It allows users to directly interact with digital objects as they do in the real world. GHOST® SDK, the software coming with Phantom, can handle complex computations and allow developers to deal with simple, high-level objects and physical properties like location, mass, friction and stiffness [SensAble Technologies, online].

Developed virtual environment is shown in Fig. 1. The peg is coupled with the phantom (ie the manipulation point) through a spring-damper system. The peg is a dynamic rigid object in the virtual environment. The force and torque reacted to the peg are transferred to PHANTOM through the spring-damper system. The hole is static in the environment while the peg can be translated and rotated. The rotation is controlled by pressing keys at this stage. A 6 DOF Phantom will be employed soon.

The haptic rendering model of the peg-in-hole insertion generating the force data is based on the PointShell method [Renz et al., 2001] [McNeely et al., 1999]. However, the points in the PointShell and force computation time per loop are reduced by only considering the dotted points on the peg and the black points on the hole as illustrated in Fig. 2(a). The direction of the force vector at dotted and black points are shown in Fig. 2(b).

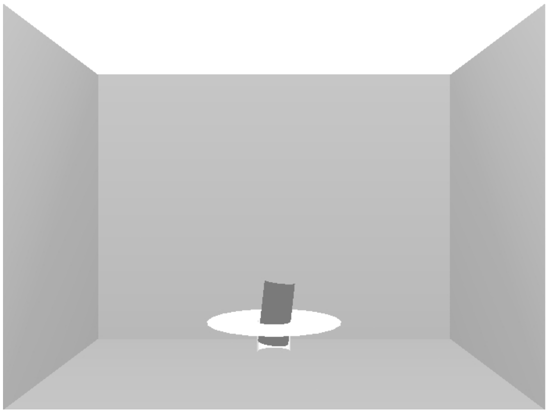

Fig. 1. Peg-in-hole Insertion Virtual Environment

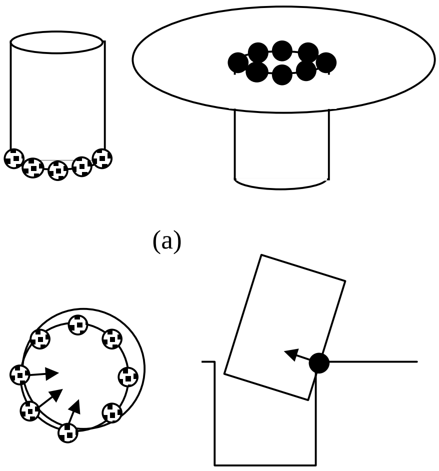

Fig. 2. The method used in haptic rendering

The force generated at each point is the sum of the Coulomb force and the friction force as shown in Fig. 3.

The direction of the Coulomb force is perpendicular to the contact surface and points to the moving object. The magnitude of the Coulomb force generated at each point is calculated by

$$f_c = k*d + c*d_a + b*v \qquad (1)$$

where,

d is the depth of the point in the contacting static object;

d_a is the accumulated depth during a continuous contact between the point and the static object;

v is the velocity of the object and is calculated by the current *Depth* minus the last *Depth* divided by the sampling time;

k is the stiffness coefficient;

b is the damping coefficient;

c is the coefficient for the accumulated depth.

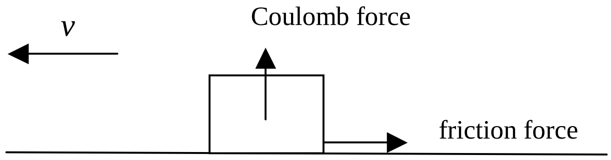

Fig. 3. Coulomb force and friction force

The direction of the friction force is along the contact surface and opposite to the moving direction. The magnitude of the friction force generated at each point is calculated by

$$f = \sigma * z \qquad (2)$$

where

z is the strain describing micromovements between the two objects, which is not allowed to exceed a small value called the breakaway distance z_{max};

σ is the stiffness relating force to strain.

Assume x_i is a point fixed on the moving object, and y_i is an adhesion point on the static object as shown in Figure 4. The following relationship is used to calculate z_i by

$$z_i = x_i - y_i$$

$$\begin{cases} y_i = x_i \mp z_{\max}, & if\ |x_i - y_{i-1}| > z_{\max} \\ y_i = y_{i-1}, & otherwise \end{cases} \qquad (3)$$

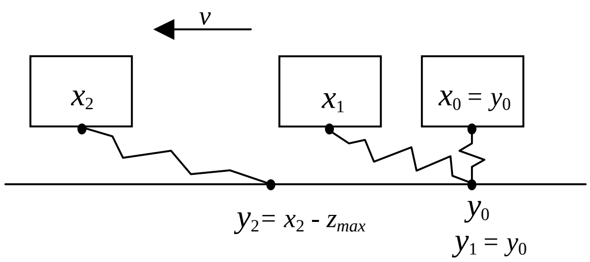

Fig. 4. The definition of the strain z

The depth d in equation (1) is found by locating the intersection point between the surface of Object A and a line defined by the current and previous locations of a point on Object B, and then projecting the vector which points from the current point to the intersection point along the normal of the surface of Object A at the intersection point as shown in Fig. 5(a).

In the virtual environment, the black points are static since the hole is static, while the peg is moving. Hence, the position of a previous point is estimated based on the current position of the peg (Fig. 5(b)). It is assumed that coordinates of the estimated previous point in the coordinate frame of the current peg position are the same as the coordinates of the current point in the coordinates frame of the previous peg position. Thus the depth of the black point in the peg can be determined as shown in Fig. 5(a).

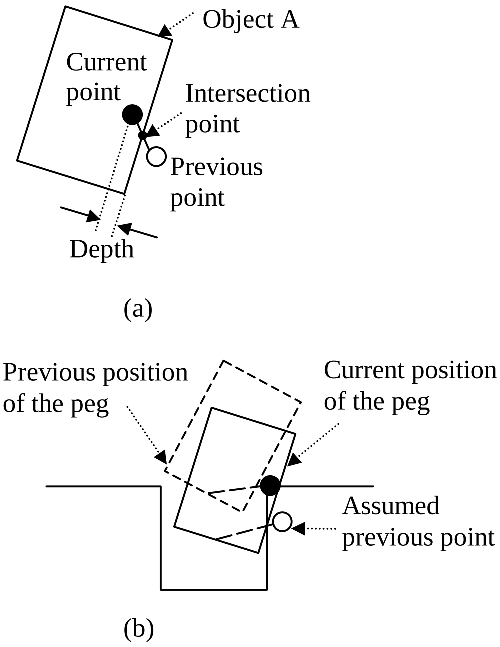

Fig. 5. Depth Calculation

3 Learning the Task

The overall approach pursued in this work is presented in Fig. 6. As illustrated in this diagram, the robotics manipulator mimics the behaviour of the human operator by acquiring the skills and producing the machine control action $u^m(t)$ from $y^h(t)$ as illustrated in Fig. 6.

A manipulation task consists of a sequence of basic skills. Identification of these basic skills and mapping them on to equivalent series of robot manipulation primitives, form the core of an algorithm for skill acquisition and transfer of those skills from human to a robotic manipulator. Such skill-based manipulation is an effective way for a robotic manipulator to execute a complex task.

The basic skills are defined according to the contact state transition of a task, independent from the configuration of a manipulator [Nakamura et al., 1996]. In a virtual manipulation environment, the basic skills can be also identified by the contact states and state changes [Takamatsu et al., 1999] [Onda et al., 1995]. Using this approach, the basic skills can be automatically extracted from the manipulation carried out in the virtual environment.

Skills can be classified at different levels according to difference between state changes. For example, the whole insertion progress can be divided into search and insertion phases. The search and insertion skills are two high level skills which result in critical state changes by driving the peg from the initial state to touch the hole and inserting peg after touching the hole respectively. Each high level skill can be divided into low level skills which result in minor state changes.

Two types of skills based on state changes are employed.

1. Skills based on both the current state and the next state
2. Skills only based on the current state

In this work, a simple method is used to identify the optimal state change sequence for insertion phase. Initially different states are classified using a fuzzy neural network. This is achieved according to the forces/torque and translation along the Z-axis. The classification is then used to recognize the state change sequence for each training data file, in which the outputs are actions such as rotating of the hole. The inconsistent or unintended actions should be identified and removed from the training data. The learning algorithm primarily learns the actions which result in a proper change of state.

The optimum sequence to perform the task is either identified from the training data or generated by combining different sequences according to a criterion such as the shortest time. If Sequence A takes shorter time from the initial state to a mid-state B but longer time from state B to the final state than Sequence B, then the first part of Sequence A is combined with the second part of Sequence B to generate a sequence with shorter implementation time than both Sequences A and B. The actions or outputs recorded for each state change in the optimum sequence with different initial state are the first type of skills. For the round peg-in-hole insertion, the data can be mapped symmetrically to reduce the amount of training data needed.

The two types of skills learned by the perception module or the learning module are stored in the skill database. In real insertion, the task planner module can decide which type of skills to be applied. If the current insertion state is among or near any state in the optimal space, the recorded actions for these optimum sequences are employed to perform the task. On the other hand, if the current state is not among or near any state in the optimum sequences stored in the skill database, a second type of skill generated through a fuzzy neural network is applied until a state among or near a state in the optimal space is produced.

The experimental rig with very small clearance consists 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 3 DOF (Degrees of Freedom) are sufficient to study the insertion phase. Some of the experimental results are illustrated in Figure 7. The variation of 9 normalized series of fx, fy, fz, Mx, My, z, Delta_z Step_x and Step_y are illustrated in this diagram, where

fx, fy, fz are forces,

Mx, My are torques,

z is the translation along Z-axis,

Delta_z is the change of the translation along Z-axis,

Step_x and Step_y are generated according to the acquired skills which indicate the steps taken by the step motors for turning the hole around x-axis and y-axis, respectively.

On the horizontal axis in Fig. 7, 1 indicates the first sampled state; 2 indicates the second sampled state and so on. The peg is jammed in the states with suddenly increased amplitudes of the forces and torque. Each time the peg is jammed or some force threshold is reached, the actions are taken immediately, ie, the peg stops moving down or even moves up a little bit and the stepper motors turn the hole accordingly. If the peg is not jammed, it is just moved down continuously. It can be seen in Fig. 7 that there are only 6 series actions taken in the insertion phase, which shows the effectiveness of the method.

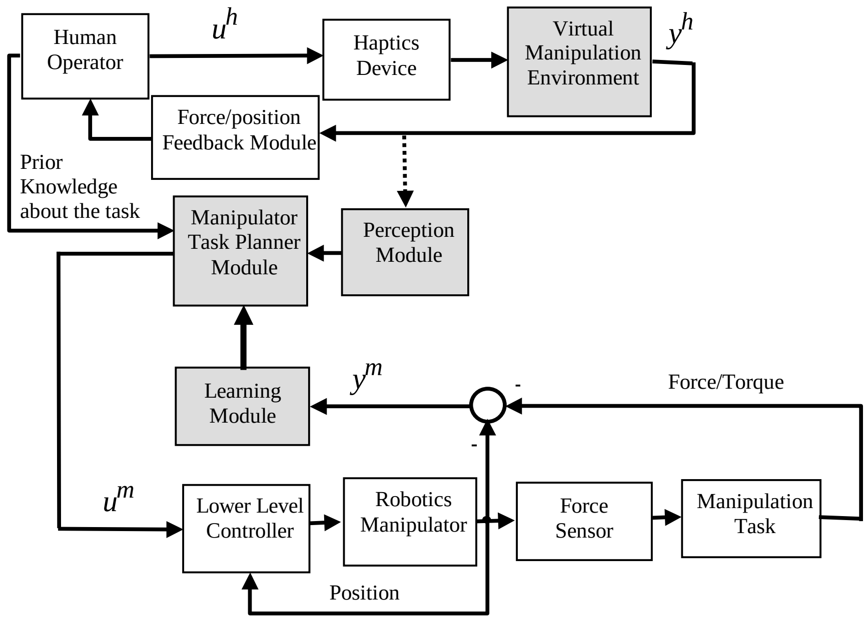

Fig. 6. Overall model of the system

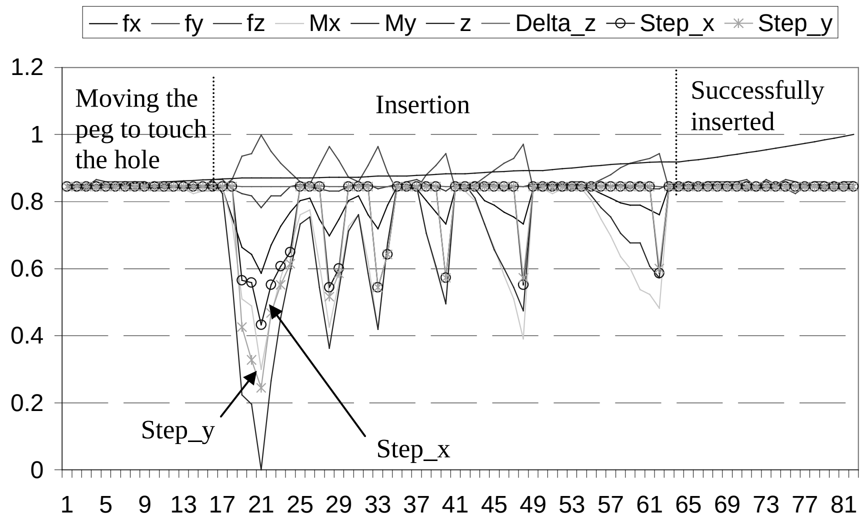

Fig. 7. Results

4 Conclusion

The work conducted to study the transfer of manipulation skills from human to a robotics manipulator through demonstrating the task in a haptic rendered virtual environment is reported. The overall concept has been presented. Broadly speaking, the project has a generic scope that is novel and innovative. It explores how human motor manipulation skills can be replicated by a machine. It also aims at providing a new insight into the nature of transfer of manipulation skills from human to machine.

The work at this stage is developed in the context of the peg-in-hole insertion process. The concepts and methodologies developed for this application will be expanded in the next stage of the project to offer more generic algorithms and models.

Acknowledgements

The support of ARC (Australian Research Council) and CRC (Cooperative Research Centre) for Intelligent Manufacturing and Systems Ltd for this project is acknowledged.

References

[McNeely et al., 1999] W. A. McNeely, K. D. Puterbaugh, & J. J. Troy. Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling, *Proc. ACM SIGGRAPH*, 401-408, 1999.

[Mussa-Ivaldi et al., 1985] F. A. Mussa-Ivaldi, N. Hogan, and E. Bizzi. Neural, mechanical and geometric factors subserving arm posture in humans. *J. Neurosci*., Vol. 5, pp.2732-43, 1985.

[Nakamura et al., 1996] A. Nakamura; T. Ogasawara; T. Suehiro; H. Tsukune, Skill-based back-projection for fine motion planning. *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, vol 2, pp 526 –533.

[Onda et al., 1995] H. Onda, H. Hirukawa, and K. Takase. Assembly motion teaching system using position/force simulator - extracting a sequence of contact state transition. *Proceedings, 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95, 'Human Robot Interaction and Cooperative Robots*', vol.1, pp: 9-16.

[Renz et al., 2001] M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, G. Hirzinger. Stable Haptic Interaction with Virtual Environments Using an Adapted Voxmap-PointShell Algorithm. *Proceedings of the Eurohaptics Conference*, Birmingham, UK, 2001.

[SensAble Technologies, online] SensAble Technologies - Haptics Research. avaible online, http://www.sensable.com/haptics/

[Takamatsu et al., 1999] J. Takamatsu; H. Kirnura; K. Ikeuchi. Classifying contact states for recognizing human assembly task . *Proceedings, 1999 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp: 177 -182, 1999.