

Telemanipulation Enhancement through User's Motion Intention Recognition and Fixture Assistance

Wentao Yu¹, Rajiv Dubey¹, Norali Pernaleté²

¹Rehabilitation Robotics Laboratory, University of South Florida, USA

²Electrical and Computer Engineering, Western Michigan University, USA

wyu, dubey@eng.usf.edu, npernale@wmich.edu

Abstract

In telemanipulation system, assistance of variable position/velocity mapping or virtual fixture can improve manipulation capability and dexterity. This assistance provides aids not only for path following, but also for reaching target and avoiding obstacle. Conventionally, these assistances are based on the environment information, without knowing user's motion intention. In this paper, user's motion intention is combined with real-time environment information for applying appropriate assistance. If the current task is following a path, a hard virtual fixture orthogonal to the path is applied. Or if the task is position a target, an attractive force field is produced. In order to successfully recognize user's motion intention, a Hidden Markov Model (HMM) is developed to classify human actions, such as following path, positioning target and avoiding obstacle. The algorithm is tested on the simulation platform.

1. Introduction

The fundamental purpose of a telerobotic system is to extend operator's sensory-motor facilities and manipulation capabilities in remote environment [1]. This approach is guided by the philosophy that the human operator should remain in direct control of the slave at all times, with human-independent control parameters altered according to sensor information. However, manipulation tasks such as assembly are still difficult for telerobotic system. In many cases, the user's physical labor load of completing a task manually is replaced by mental labor of controlling the remote input device mentally. In the fields of rehabilitation robotics, this is the main hindering for the wide application of telerobot [2].

Our recent work in telemanipulation system for rehabilitation engineering motivated us to enhance manipulation accuracy and reduce fatigue [3,4,6]. These assistances, mostly about variable position/velocity mapping based on sensory data, in some extent reduced user's mental load and improve manipulation accuracy. But they still have limitations that the assistances are task related. Due to the unstructured environment of rehabilitation robotics and a variety of different tasks, a general assistance for telemanipulation system is necessary

to improve the manipulation capabilities. This paper is trying to combine the environment information with user's motion intention before applying appropriate assistance. Human motion intention is classified from movement velocities through Hidden Markov Model: following a path, posing a target and avoiding an obstacle. For each different motion, appropriate assistance is provided. For example, if the motion is following a path, a hard virtual fixture orthogonal to the path is applied along the path, just like a ruler. If the motion is positioning a target, an attractive force field will be applied.

2. Background

Assistance for performance in telerobotics field has been a familiar topic. The first kind of assistance is the position/velocity mapping based on sensory information or/and force feedback[6]. The other is virtual fixture. This approach has been recently developed and used as means of providing direct, physical assistance[5,7,8,9]. Just image drawing a straight line without a ruler, it is very difficult. Virtual fixture plays the same role as a ruler to enhance human's drawing a straight line. Both of these assistances can enhance human's performance accuracy for complex tasks execution and reducing time consuming. But the limitation is that they are related to some specific tasks. In order to provide general assistance, a specific task needs to be divided into several simple, general subtasks.

Hidden Markov Model(HMM) is a widely used for classification and recognition algorithm. Rich in mathematical structure, HMM is a trainable statistical model, with two appealing features: 1: no a priori assumptions are made about the statistical distribution of data to be analyzed, and 2: a degree of sequential structure can be encoded [3]. As such, they have been applied to a variety of stochastic signal processing. In robotics field, Yang and Xu applied HMM to teleoperation skill learning and gesture recognition, thus eliminating sluggish motion and correcting a motion command that the operator may mistakenly generate [11]. Hannaford and Lee [12] classified task structure in teleoperation based on HMMs. Actually, HMM is very powerful for classification [3]. If the user's motion in telemanipulation can be classified into several different and task-independent movements, the

appropriate assistance for each class of motion is general and can be applied.

3. HMM-based User's Motion Recognition

For the unstructured environment, it may be not possible to predefine the desired trajectory for a task. But the movement direction at a certain stage can be known. For example, if the computer vision locates the target or obstacle, the movement at this stage is moving toward the target or going around the obstacle. Obviously, the trajectory of moving toward target can be defined as a straight line. The trajectory of going around an obstacle can be defined as the movement orthogonal to the direction. So if we can recognize user's intention, appropriate assistance can be applied to the task execution. Even in 3D space, we decompose velocity vector into two parts, v_p , velocity component along the path tangent direction and v_o , velocity component orthogonal to the path tangent. When users are following a path, $v_p \gg v_o$ (See figure 1); when avoiding obstacle, $v_p \ll v_o$ (Figure 2); when positioning, both v_p and v_o (Figure 3), when stopping, both velocity components are close to zero. Based on the velocity information, Hidden Markov Model is used to classify the user's intention into three types: following a path, positioning a target and avoiding an obstacle. Since these two velocity components are orthogonal, they are independent. In order to apply HMM to model these two velocities components, a 2-dimension HMM is used.

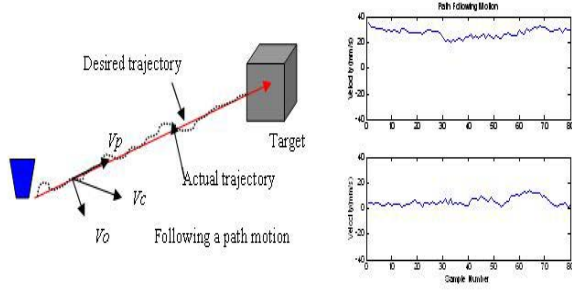


Figure 1 Path-following motion, $v_p \gg v_o$

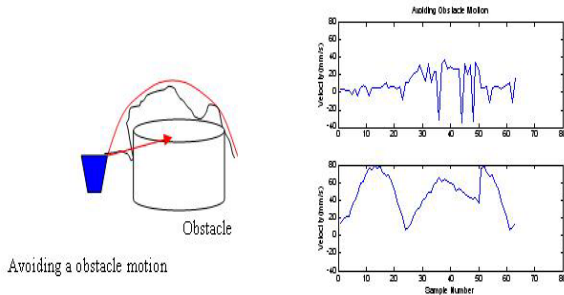


Figure 2 Avoiding obstacle motion, $v_p \ll v_o$

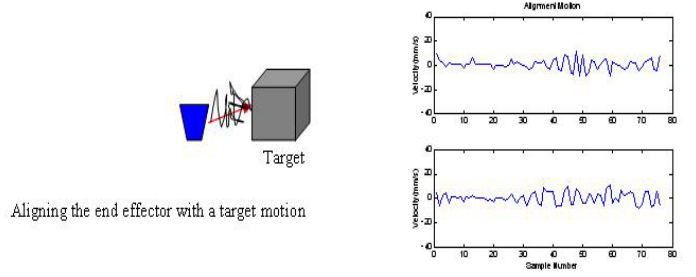


Figure 3 Motion of aligning end-effector with target, $v_p \cong v_o$

3.1 Data Preprocessing

The velocity of the input device is sampled at 100Hz rate. The data is represented as:

$$X = [V_p, V_o] \quad (1)$$

where V_p and V_o are the set of v_p and v_o samples.

$$\begin{cases} V_p = [v_{1,p} & v_{2,p} & \dots & v_{n,p}]^T \\ V_o = [v_{1,o} & v_{2,o} & \dots & v_{n,o}]^T \end{cases} \quad (2)$$

where n is the sample number. Since V_p and V_o plays the same role, we just demonstrate the data processing of one of them, say V_p . From the sampled data, a 16-point width window with 50% overlap is used to select data. Prior to spectral conversion, in order to minimize spectral leakage, a hamming window is first used to filter each frame. The Hamming window maps a k -length ($k=16$ in this case) real vector $V = [v_1, v_2, \dots, v_{16}]^T$ to a k -length real vector $V_h = [v_{1,h}, v_{2,h}, \dots, v_{16,h}]$. Then FFT analysis is applied for every windowed data. The FFT transform maps a k -length vector V_h to a k -length complex vector $Z = [z_1, z_2, \dots, z_{16}]$. Finally, define the power spectral density (PSD) estimates for the hamming-Fourier output \vec{z} . The PSD estimates is given by,

$$P(\vec{z}) = [P_0(\vec{z}) \ P_1(\vec{z}) \ \dots \ P_{k/2}(\vec{z})] \quad (3)$$

where $P_0(\vec{z}) = \frac{1}{H_{ss}} |z_0|^2$

$$P_i(\vec{z}) = \frac{1}{H_{ss}} (|z_i|^2 + |z_{k-i}|^2), \quad i \in \{1, 2, \dots, k/2 - 1\},$$

$$P_{k/2}(\vec{z}) = \frac{1}{H_{ss}} |f_{k/2}|^2, \quad \text{and}$$

$$H_{ss} = k \sum_{q=1}^k H_k^2$$

We train the VQ codebook by those vectors and the codebook was produced by LBG algorithm [13]. The LBG VQ(vector quantization) technique maps these 9-dimensional vectors into a finite set of vectors $Y = \{y_i; i = 1, 2, \dots, L\}$, where L is the length of the codebook(it is determined 32 in our case). Each vector y_i is called a code vector or a

codeword and the set of all the codewords is called a *codebook*. Associated with each codeword, y_i , is a nearest neighbor region called *Voronoi* region, and it is defined by:

$$V_i = \{x \in R^k: \|x-y_i\| \leq \|x-y_j\|, \text{ for all } j \neq i\} \quad (4)$$

The 256 9-dimensional vectors in the codebook are 256 symbols in the output probability distribution functions for discrete HMM. Similarly, a codebook for the velocity component v_o vector and the 256 symbols are also obtained in the same way. The computation of the data preprocessing part is illustrated in Figure 4.

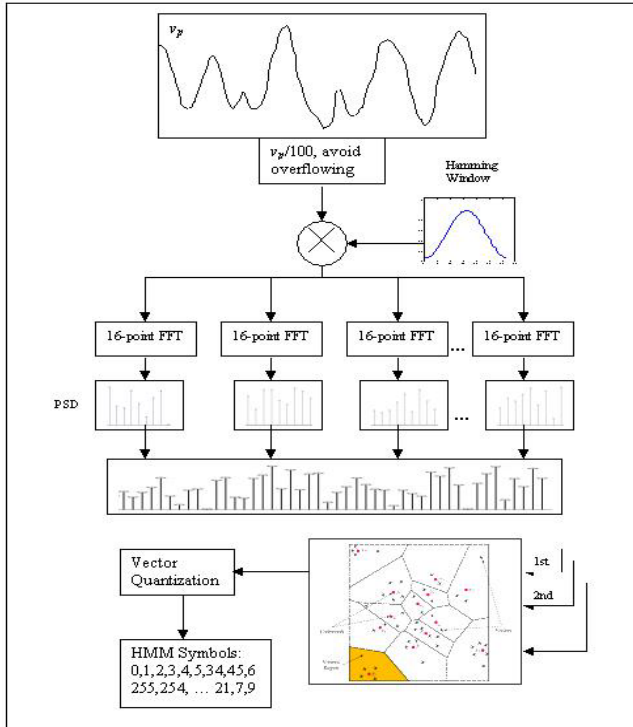


Figure 4 Conversion of continuous velocity data to a sequence of discrete symbols

3.2 Hidden Markov Model Construction

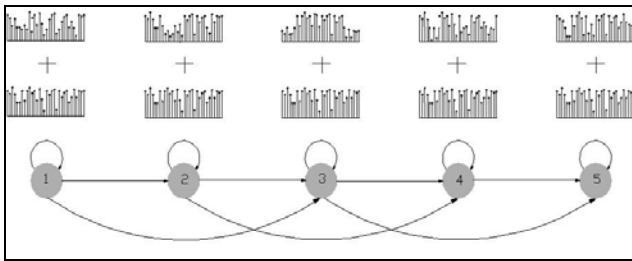


Figure 5 Left-Right HMM with 5 States and each state has two dimensional observations, each of which has 256 symbols

A HMM consists of:

- 1) States and the corresponding observations
- 2) State transition probability matrix $A=\{a_{ij}\}$;

- 3) Observation probability matrix $B=\{b_j(k)\}$;
- 4) Initial state distribution $\pi=\{\pi_i\}$.

Figure 5 shows a 5-state HMM with 2-dimensional 256-length observations. Each state is characterized by two sets of probabilities: a transition probability, and an output probability [11]. The three model using problems are:

- 1) Likelihood of a sequence of observations given a model;
- 2) The most likely sequence of states the model traverses given an observation;
- 3) Adjusting the model parameters to best account for a set of observation sequences (training).

In this paper, we are trying to classify three different motions based on partial observation sequence. For each class of motion M_i , we design a separate HMM. The observations are sequences of symbols obtained in LBG VQ algorithm. The answer of the third problem is to set the parameters of the model for each class of motion. For a given model, let $\alpha_{t+1}(j)$ represent the likelihood of observing vector o_1 to o_t and being in state j at time t . The partial likelihood of each model can be computed by the forward procedure:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix} \quad (5)$$

where $\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N$

The probability of the partial observation sequence is:

$$P(O|M) = \sum_{i=1}^N \alpha_T(i) \quad (6)$$

The computation of Eq. (6) requires summing over N^T possible sequences. Instead, a forward-backward procedure is invented to do this[10]. Once the motion models have been built, the model which produces the largest likelihood $P(O|M)$ given an observation is the classification result.

Since we have 2-dimensional observations, a 2-dimensional HMM is used. A multidimensional HMM is an HMM that has more than one observable symbol at each time t for each state. In this case, the A matrix structure of a multidimensional HMM is the same as a one-dimensional HMM. But a multidimensional HMM has more B matrix, each one for characterizing stochastic distribution of each dimensional signal [11].

The two dimension data in our case is the spectral power vector of velocity components sample in path tangent and orthogonal directions. For state q_k , there are 256×2 distinct output symbols can be observed, where $O_k = [O_k(1), O_k(2)]$, $k = 1, 2, \dots, 256$. As mentioned in last subsection, these two-dimension symbols sequence are independent, the output probability can be computed as the product of the output probability of each dimension. That is, the forward procedure can be modified as [11]:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] \prod_{l=1}^2 b_j(O_{t+1}(l)). \quad (7)$$

4. Design of Assistances

Since user's motion intentions have been recognized, appropriate assistance can be designed for each different motion. Most of the movement of this task is to follow a path. We define the path curve as $p(s)$, the target position as t . When the goal during task execution is to move to a target, we assume that the desired trajectory is a straight line which connects the current Cartesian position of the end-effector and the target. A preferred reference direction d , can be defined for each current point of the end-effector x_c as

$$d(x_c) = \frac{x_t - x_c}{\|x_t - x_c\|} \quad (8)$$

Where x_t and x_c are the target position the current position of the end-effector respectively. We decompose the v_i into two orthogonal components:

$$v_p = (v_i \cdot d)d \quad (9)$$

$$v_o = v_i - (v_i \cdot d)d \quad (10)$$

where V_p is the velocity component along the path curve tangent, v_o is the velocity component orthogonal to the curve tangent. The desire of the path following is that the velocity tangent the curve is large and velocities component in orthogonal direction is small. If the desired trajectory of a sub-task is a straight line, a virtual fixture can provide the same assistance as a ruler helps us to draw a line.

4.1 Fixture assistance

Fixture assistance is always applied for path following except that it is removed when the user is trying to positioning an object or avoiding an obstacle. So the stiffness coefficient k_d along the curve tangent is set to be zero. The stiffness orthogonal to the curve tangent is defined as:

$$k_o = \begin{cases} k & d > r \\ 0 & d \leq r \end{cases} \quad (11)$$

where k is the fixture coefficients maximum (it is determined 0.5N/mm for this experiment), d is the distance between the end-effector and the obstacle center position, r is the force fields radius. This means that once the end-effector goes into the force field, path following fixture is gone.

4.2 Force Field Design for targets and obstacles

Since grasping object and obstacle avoidance are small operation, it is usually not easy to execute them. Force fields generated from the center position of target or obstacle can provide some extent of assistance. Based on this concept, force fields are designed around targets and obstacles. We call the radius of force filed as r . In this project, the force filed is defined using spring force. The k

is determined as 0.02N/mm. For object approaching, the force is defined as:

$$f = \begin{cases} 0 & d > r \\ k(r - d) & d \leq r \end{cases} \quad (12)$$

For obstacle avoidance, the force is defined as:

$$f = \begin{cases} 0 & d > r \\ -k(r - d) & d \leq r \end{cases} \quad (13)$$

So once the end-effector goes into the radius r for object grasping, the attractive force originated from object center position can provide assistance. The force clues generated by position and approach fixtures are shown in Figure 6. Payandeh et al used this kind of virtual fixture as a task-dependent telemanipulation aid [5]. However, the origin of the force fields needs to be determined from sensory data. In addition, r should be larger than the size of target or obstacle.

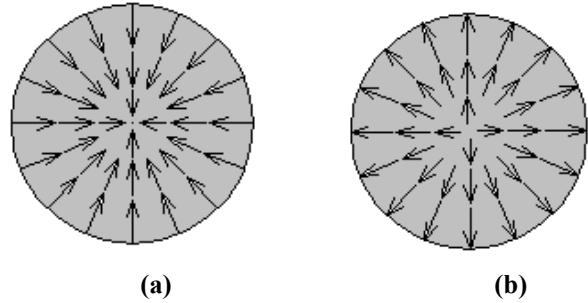


Figure 6 Force fields illustration (a: Attractive force, b: Repulsive force).

5. Experiments

5.1 Experiments

Our telemanipulation simulation system is composed of a visualization scene and a haptic device. The visualization component, simulation scene, is realized through the `gstScene` class of GHOST package. In this experiment, the task is to move the end-effector from the origin (0,0,0) to go get the object(-80,50,0)(this means the end-effector must reside in the object sphere for a short time) and then avoid the obstacle (0,45,0) and then put the object in the destination place (80,50,0) and go back the origin. The object and the destination place are simulated as 8mm radius sphere. The obstacle and the end-effector are simulated as a 15mm and 5mm radius sphere respectively. User is told to move the end-effector as fast and smooth as possible. The task execution environment can refer to Figure 7. In order to avoid confusing, the operator is allowed to move on a planar surface since a plane constraint has been added to the haptic device. In this experiment, we just concern about the straight-line path since it is relatively easy to get from environment information. Of course, this

algorithm can extend to complex trajectory application if we can define the trajectory.

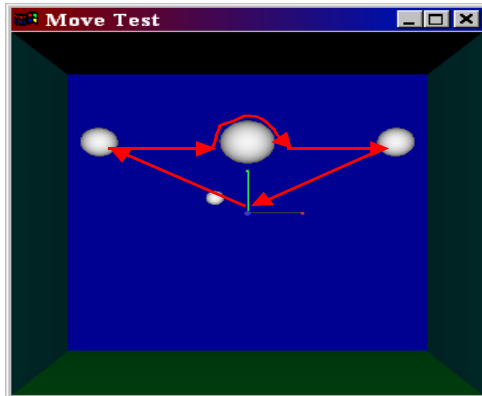


Figure 7 Simulation of the Task Execution

5.2 Experimental Results without Assistance

First, an expert user completed the task several times without assistance. In the first a couple of tests, the common performance of the system is shown in figure 8, 9. As expected, the free motion has much difficulty for positioning target and following path. The velocity component orthogonal the path is not small compared with the useful velocity components, tangent to the path. Table1 summarizes the results, including path following error (mm) and execution time(s).

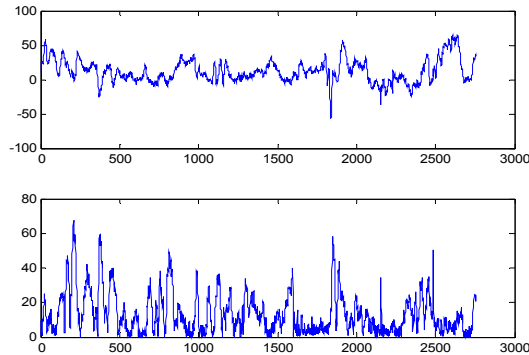


Figure 8 Velocity Components without Assistance

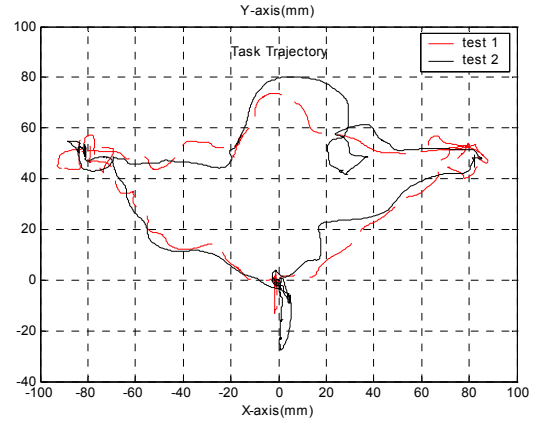


Figure 9 Trajectories without Assistance

Table1 Performance Summary without Assistance

Subject	Path Error (mm)		Execution Time(s)	
	Mean	Stdev	Mean	Stdev
1	10.1	2.4	21.5	1.9
2	8.9	1.5	20.2	3.3
3	11.8	2.6	22.1	3.4
4	10.3	2.5	20.7	2.8

Table 2 Motion Recognition Rate

5.3 Motion Recognition

An expert user completed the task 10 times. From them, we collect 250 samples of data for each motion, the first 200 for training and the rest of the sample for testing. For a total 50 testing samples of four motions, the system successfully recognized 43 samples. The accuracy is 86%. Definitely, the size of the training set influences the recognition accuracy. After we include more samples into the training set, say 500, the system can recognize 92 samples from 100 testing samples. Most of these recognition error happened between positioning target and following path. For a complete task execution which needs all the three motions, the motion recognition performance is shown in Table 2.

5.4 Experimental Results with Assistance Based on

Motion	correct rate	Incorrect rate			
		to 1	to 2	to 3	to 4
1: Following	90.5%	----	4.0%	2.3%	3.2%
2: Aligning	89.1%	6.4%	----	2.3%	2.2%
3: Avoiding	88.3%	7.7%	2.0%	----	2.0%
4: Stopping	98.7%	0.0%	1.3%	0.0%	----

Motion Recognition

As mentioned before, the corresponding assistance is applied to a specific class of task. If the task at a certain stage is path following, a hard fixture is applied so that the

end-effector can move along the path. Once the motion has been changed into positioning motion, hard fixture is replaced by an attractive force field. For avoiding an obstacle, a repulsive force field is applied. Generally, the shape of an obstacle is difficult to build from sensors information in real world. So creating a desired path for object avoidance is not a good way. In this paper, this repulsive force field just provides some assistance for operator to go around obstacle. With these assistances, the expert user also executed the same tasks for several times. One task performance is shown below (figure 10, 11). The hard fixture provides assistance for path following, primarily due to the fact that reaction force applied to the PHANToM tool tip can make it back once there is some deviation from the path. Most of time, the velocity component is much small compared with the velocity components tangent the path. Large orthogonal velocities only happen when the user is positioning target or avoid obstacle.

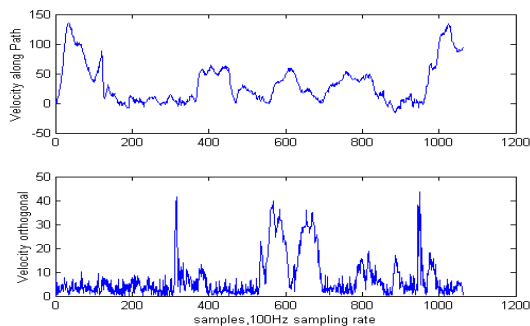


Figure 10 Velocity Components with Assistance

Table 3 Performance Summaries with Assistance

subject	Path Error		Execution Time	
	Mean	Stdev	Mean	Stdev
1	5.1	0.5	12.6	0.7
2	4.6	0.8	11.8	0.6
3	5.3	0.9	12.9	1.2
4	4.8	1.1	13.4	1.2

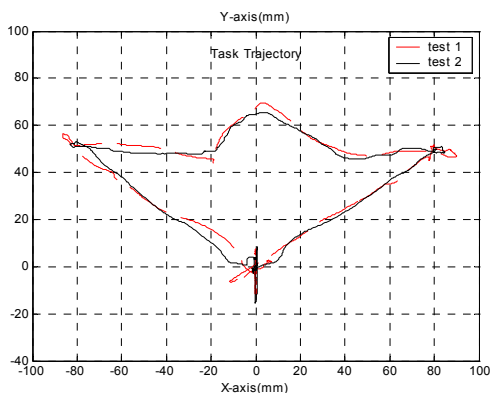


Figure 11 Trajectories with Assistance

6. Conclusions

Hidden Markov Model is effective for motions classification. As long as the training set is sufficiently large, the motion recognition accuracy is close to 100%. The selected assistance based on the recognized motion is appropriate for each type of motion. The experiment results without assistance verified that operator always has random errors which result in the difficulty in following a path and positioning a target. The experimental results with assistance showed that all those undesired random errors were removed or reduced. The HMM based assistance is useful for improving performance accuracy and decreasing execution time.

These results indicate that the appropriate assistance approach selection based on motion intention is possible. Conventionally, sensors can not tell us which object is the target/obstacle, but the teleoperator can tell computer this judgment based on his motion intention since he/she is in the loop. When he/she is trying to approach it, it's a target. When he/she is getting around it, it's an obstacle.

In order to improve the recognition accuracy, the dimension number of the Hidden Markov Model can be expanded. As long as they all are independent, the added dimensions will not bring too much computation complexity.

References

1. Pepper, R.L. and Kaomea, P.K. "Research Issues in Teleoperator Systems", 28th annual Human Factors Society Meeting, San Antonio, Tx, 1984.
2. Bolmsjo, G.; Neveryd, H.; Efring, H.; "Robotics in rehabilitation", Rehabilitation Engineering, IEEE Transactions, Volume: 3 Issue: 1, Mar 1995 Page(s): 77 -83.
3. S. E. Everett, R. V. Dubey "Human-Machine Cooperative Teleoperation of Fitts Task Using Uncertain Sensor and Model Data", IROS, 1998.
4. Dubey, R.V.; Everett, S.E.; Pernalet, N.; Manocha, K.A., "Teleoperation assistance through variable velocity mapping", IEEE Transactions on Robotics and Automation, Volume: 17 Issue: 5, Oct 2001.
5. Z. Stanisic, S. Payandeh, and E. Jackson. Virtual fixture as an aid for teleoperation., in 9th Canadian Aeronautic and Space Inst. Conference., 1996.
6. N. Pernalet, Wentao Yu, and R. V. Dubey W. Moreno, "Variable Position Mapping Based Assistance in Teleoperation for Nuclear Cleanup," in ICRA, 2002
7. Luc D. Joly and Claude Andriot, "Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism," ICRA 1994.
8. P. Marayong, A. Bettini and A. Okamura, "Effect of Virtual Fixture Compliance on Human-Machine Cooperative Manipulation", in IEEE/RSJ proceedings, Lausanne, Switzerland, Oct 2002.

9. Rosenberg, L.B.; "Virtual fixtures: Perceptual tools for telerobotic manipulation", Virtual Reality Annual International Symposium, IEEE, 18-22 Sep 1993 Page(s):
10. L. R. Rabiner, " A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition", Proceedings of the IEEE, Volume: 77 Issue: 2, Feb 1989.
11. Jie Yang, Yangshen Xu, Chiou S. Chen, " Human Action Learning via Hidden Markov Model", IEEE Transactions on Systems, Man, And Cybernetics-Part A: Systems and Humans, Volume: 27 No.1 , January 1997.
- 12.B. Hannaford and P. Lee, Hidden Markov Model Analysis of Force/Torque Information in Telemanipulation. The International Journal of Robotics Research, Oct.1991, Vol. 10,No.5, pp. 528-539.
13. R.M. Gray, " Vector quantization", IEEE ASSP Mag., vol.1,No.2. pp.4-29,1984.