Modeling and Classifying Six-Dimensional Trajectories for Teleoperation Under a Time Delay.

Vytas SunSpiral(1)(2), Kevin R. Wheeler(3), Mark B. Allan(1), Rodney Martin(3)
(1)QSS Group Inc., NASA Ames Research Center, MS 269-3, Moffett Field, CA, USA 94035
Email: Vytas@email.arc.nasa.gov
(2)Formerly published as Thomas Willeke
(3)Intelligent Systems Division, NASA Ames Research Center, MS 269-1, Moffett Field, CA, USA 94035

1.0 ABSTRACT

Within the context of teleoperating the JSC Robonaut humanoid robot under 2-10 second time delays, this paper explores the technical problem of modeling and classifying human motions represented as six-dimensional (position and orientation) trajectories. A dual path research agenda is reviewed which explored both deterministic approaches and stochastic approaches using Hidden Markov Models. Finally, recent results are shown from a new model which represents the fusion of these two research paths. Questions are also raised about the possibility of automatically generating autonomous actions by reusing the same predictive models of human behavior to be the source of autonomous control. This approach changes the role of teleoperation from being a stand-in for autonomy into the first data collection step for developing generative models capable of autonomous control of the robot.

2.0 INTRODUCTION

The motivating problem for this research has been how to best control the JSC (NASA Johnson Space Center) Robonaut platform given a 2-10 second time delay. Robonaut is a humanoid robot designed to have dexterous manipulation capabilities similar to those of a suited astronaut, enabling it to perform many of the construction and repair functions that would currently require an EVA by a human astronaut [8]. Besides reducing the number of dangerous EVA's that must be performed, Robonaut would also be able to perform emergency repairs. Currently it takes an astronaut 4-5 hours to suit up for an EVA. If there is a critical emergency which must be repaired quickly this delay may doom the astronauts. Thus, if the Robonaut were able to deploy immediately it might be able to handle the emergency repair before a human could even start the EVA. So, the next question is, how should this robot be controlled? Currently, fully autonomous control for such a high degree of freedom robot that is verifiably robust and safe is a long term research project and is unlikely to be flown soon. Thus, Robonaut will be controlled through some form of teleoperation. While success has been shown in the lab with fully immersive teleoperation, difficulties emerge when a time delay is introduced. The 2-10 second time delays that this project is concerned with can occur if there is a human controller on the ground and the Robonaut is in LEO (up to 2 second delay) or out at the moon (8-10 second communications delay). While these distances are not so great, it is the nature of the communications networks and topology which introduce such large time delays. When a multi-second time delay is introduced, direct teleoperation slows down tremendously since, for safety reasons, the operator is forced to adapt a "bump and wait" strategy. This entails making a small movement (the "bump") and then waiting to get feedback on the results of that motion. Controlling the robot with a "bump and wait" teleoperation strategy is slow and tedious, and does not optimize the speed advantage of deploying the robot for a critical emergency repair.

Thus, in order to deal with the time delay a sliding scale of autonomy is proposed. This is a hybrid control scheme that keeps a human in the loop (and ultimately in control for safety reasons), yet allows for and greatly increases operation speeds. This is done by allowing small, well understood and conditioned tasks (which are often the routine tasks) to be performed autonomously. The granularity of these autonomous tasks can vary from none (full teleoperator control), to simple motions such as positioning the hand close to (but still a safe distance away from) an object of interest, to compound tasks such as reaching towards a known object, and then grabbing and retrieving it.

Once capabilities for these autonomous tasks exist, one must ask how they will be commanded. As will be shown below, the operator currently controls the Robonaut by operating directly in a richly expressed virtual environment. The goal of this work is to be able to recognize and predict when the operator is engaging in one of the pre-defined autonomous tasks so that the action can be triggered on the robot. The control of these autonomous actions is purposefully kept in physical context (as opposed to developing some command grammar) to facilitate the seamless transition between direct tele-operation control and the varying levels of autonomous actions.

Currently the autonomous actions are hand coded. One of the basic questions that this group is researching is to see if the process of creating autonomous actions can be automated by modeling human behavior. If we can accurately understand and model the methods humans use to solve certain problems, then it is possible that those very same models can be used to control a humanoid robot like Robonaut to accomplish similar tasks. Thus, we view teleoperation not simply as a means to accomplish a task in the absence of robust autonomy, but rather as the first step in building models of human action for the purpose of developing robust autonomous control [9].

The current method of controlling Robonaut involves the operator wearing two data gloves that are used to measure finger joint positions, and two magnetic trackers used to measure the x-y-z position and roll-pitch-yaw orientation of each hand (end effector). The position and orientation information is then transmitted to the robot as end effector position commands. The number of degrees of freedom in the elbow and shoulder are constrained to enable this position while maximizing strength. For safety considerations, the rate of movement of the arms is limited; thus the operators are trained to match or move slower than this rate. Most of the feedback to the operator comes from the stereo cameras mounted in the head of Robonaut and transmitted back to the teleoperator's head mounted display. Thus an operator will reach for an object so that his view from the head mounted cameras is not obscured by the hand. This can result in some simple tasks taking a very long time to accomplish. For example, in grasping a hand rail (as a rung of a ladder) the operator must make sure that the fingers can wrap around the handle using the stereo visual cues. This action typically takes several seconds for an experienced operator.

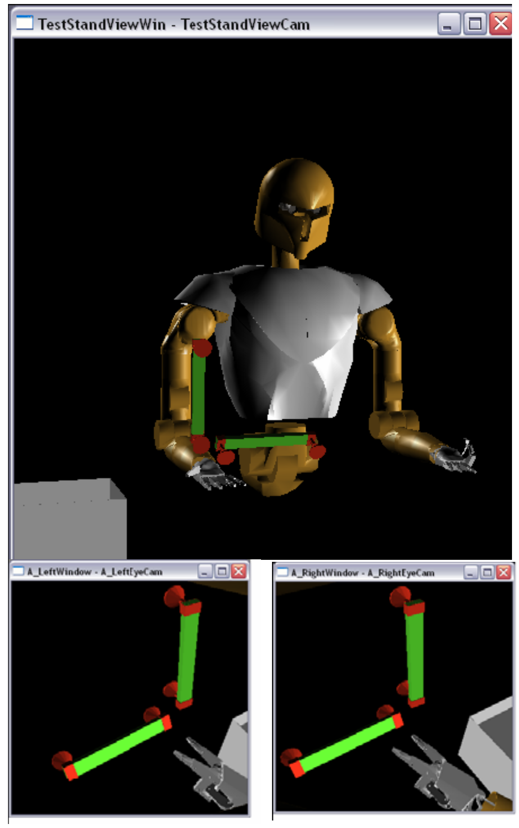

Fig. 1. Actual Simulation based experiment. The operator view from the left and right cameras of the simulated hand rails.

2.1 EXPERIMENTS

We chose two basic tasks, retrieving a hand rail mounted vertically and dropping it into a box, and retrieving a hand rail mounted horizontally and dropping it into a box. The hand rails are mounted with Velcro on a board, affixed to a stationary wall. The target box is a flexible cloth box that is open but is not within the same field of view as the hand rails. These tasks were chosen as a first step towards automating climbing on a space habitat.

The tasks consist of the following steps:

1. Start in initial position/state
2. Look down at hand (substitute for proprio-receptive feedback) and then at hand rails
3. Reach for specified hand rail (either vertical or horizontal according to plan)
4. Grasp hand rail
5. Remove hand rail from wall (pull)
6. Move hand rail over box
7. Drop hand rail into box
8. Return to initial position

The Robonaut can be operated via a simulated environment, so that the operators can perform tasks without regard for the time-delay normally associated with long distance operations. The motion commands generated in the simulated environment are then sent to the actual robot for execution. For this experiment, inexperienced operators tended to have greatly varying behaviors, whereas the variance in the data was negligible for the most experienced teleoperator. Fig. 1 shows the simulated environment in which the experiments discussed in this paper were conducted. These experiments were conducted on many different days over six months. Initial conditions varied noticeably from day to day.

2.2 TECHNICAL PROBLEM

While there are a number of other technical and design challenges to the larger problem we are solving, the core problem is the modeling and classification of six dimensional trajectories.

Each trial can be cut into a sequence of trajectory segments. In this work we focus on the first segment, which goes from the start position to one of the possible graspable objects. These trajectories are a time-stamped sequence of the X, Y, Z, location of the operator's hand and the associated rotation matrix, which we cast into roll, pitch, yaw representation. Thus, the trajectories are a time sequence of six-dimensional data points. It is important to note that the data is generated at a variable rate and will often include multiple repetitions of the same data point. Another property of these trajectories that is important to be aware of is the messy initial conditions. Since the operator starts in approximately the same location for every motion, the initial segments of all the trajectories are heavily overlapped, making it difficult to clearly separate them in the early part of the motion (Fig. 2). Furthermore, this is compounded by a large day-to-day variance on the operator's start position. (Within a single day's trials the trajectories tend to cluster fairly nicely). This non-stationarity means that there is a wide variance on the motion and location of the trajectories, and that in some cases the initial motions towards different objects end up looking very similar (Fig. 3).

Our task is: given a set of example six-dimensional trajectories to certain known points in space, develop meaningful models of the different trajectories, and then use those models to classify a new trajectory in real time, as soon as possible (*i.e.* while having seen only some initial segment of the trajectory), and with no false alarms (*i.e.* it is better to not classify than to give the wrong classification).

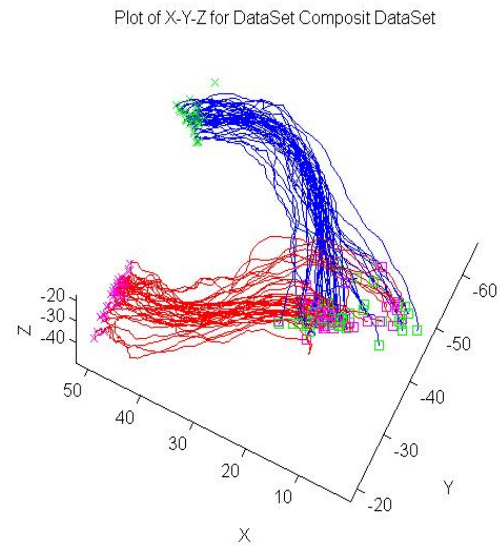

Figure 2 - This is a plot of the XYZ position information of many trajectories to two different handles. The start location on the right shows how the initial conditions for the different trajectories are heavily overlapped.

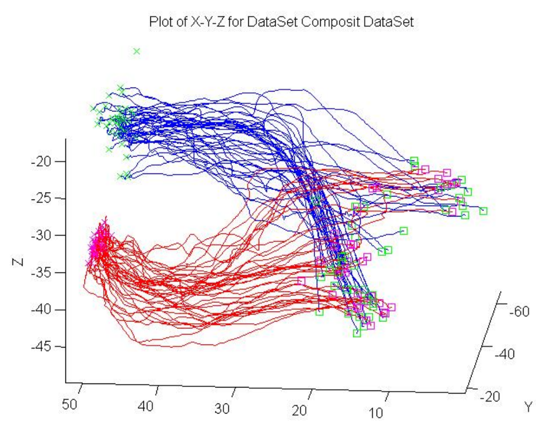

Figure 3 - This is the same data as figure 2, but viewed at a different angle to highlight the spatial variance of the start positions.

Notice that the solution to this problem, which was motivated by teleoperation and user intent prediction, can be used in other ways. Specifically, it really is about classifying and predicting human gestures. Thus, if a robot were working side by side with a human, and assuming there was a good solution for visually capturing the motion of the human's hand or arm, this trajectory classification method could be used to comprehend the intent behind the human's motions, such as which object the human was reaching for. And, as mentioned earlier, we hope that the same models could be reused to generate autonomous control.

3.0 DEVELOPOMENT APPROACH

To ensure that there was no ideological bias towards algorithms developed by the group during earlier research, we explored in parallel different approaches to solving this problem. One track continued with previous work by applying stochastic methods, such as using Hidden Markov Models to classify the trajectories. The other track looked at more deterministic methods and ended up exploring different spatial representations of the data. This research approach has been very successful, with the two different tracks starting to converge and borrow techniques from each other. Hopefully this research approach of starting from multiple points of the solution space is more likely to lead to a globally optimal solution to the problem. In contrast, research programs which start out ideologically tied to certain classes of algorithms will, at best, find only a locally optimal solution. Due to academic specialization, this last case is, unfortunately, reasonably common.

In the following section we will try to present the evolution of our thoughts on these two different tracks, and show how they have influenced each other.

3.1 DETERMINISTIC APPROACH

The initial approach to this problem was simply to calculate the distance between the operator's hand and the possible target objects. It was quickly found that a simple distance threshold was not sufficient, as one might pass reasonably near one object while reaching for the other one. Next, the first derivative of motion was examined to see which object the operator was "moving towards the most". The major lesson from this approach was realization of how important orientation was. We found examples in which the operator was positioning the hand to be in the correct orientation to grab one handrail, and the motion to do this did not involve moving towards that handrail, and even sometimes looked like it was moving directly towards some other object. Yet, these false alarms are evident when orientations of the hand and objects are considered because the back of the hand was moving towards the "false" object. This means that while the motion vector looked good, the relative orientation between the hand and the object was such that a grasp was very unlikely.

After further work, it turned out that merely looking at the instantaneous motion vector and the relative orientation was insufficient because the operator does not follow a minimum-distance path in either position or orientation space. The operators have been trained to move in such a way as to avoid singularities, self contact and other hazards for the robot. Also, since control is purely visual (they use no force-feedback information), they always move so as to maintain visual contact with both the object and the hand. This leads, for example, to a preference for underhand grasps so that the operator can see when the object is properly grasped. An overhand grasp would occlude the object being manipulated. All these constraints result in motion paths that are not necessarily the shortest distance or most direct motion towards the goal.

It was at this point that we really started to look at the data as a six-dimensional trajectory through position and orientation space. By doing this we capture the overall shape of the motion, not just the instantaneous motion. We also needed to recast the data into a time independent manner since the raw data was very dependant upon the speed at which an operator moved. To do this we decided to normalize along the length of the trajectory itself. In other words, we recast the data into a new set of data points which were a known distance apart along the piecewise linear direction of travel. The overall shape of the plotted trajectory stayed the same, but the actual data points which were used to represent it are now equidistant on the trajectory itself. Using this new representation an "average" trajectory was computed for each graspable object. A method of comparing two trajectories was developed by calculating the average distance error and then novel trajectories were compared to these canonical averages. The weakness of this is that the high degree of variance and overlap, especially in the early part of the trajectories, caused many false alarms (i.e. matching to the wrong trajectory). In order to capture this variance a memory based approach was experimented with where all the trajectories in the training set were kept in memory and the novel trajectory was compared to all of them to find the best match. Under certain conditions this was found to work fairly well, but was fragile to the inclusion of outlier examples in the training set. There is certainly a large body of literature that studies these sorts of memory systems and how to condition and clean the training data. But at this point it was decided that such a path did not lead towards our goal of creating reversible models of human behavior which could be used to generate autonomous control. Thus, we turned again to statistical approaches to create flexible models of these clusters of trajectories. We'll look at the stochastic methods next and then discuss the new converged algorithms in section 3.3.

3.2 STOCHASTIC APPROACH

The research group has a strong body of experience in applying Bayesian techniques to gesture recognition problems [10]. The Bayesian approach has is roots solidly grounded in formal probabilistic graphical modeling. Using the junction tree algorithm, DBN's (dynamic bayes nets) can easily meet the needs of any number of inferential or parameter learning problems. In this case, the parameters we would like to learn are those of a Gaussian mixture model, or multi-modal Gaussian distribution which would describe a cluster of example trajectories. We believe that human motion is Markovian in nature, and can be broken down into discrete chunks. In turn, we hypothesize that each of these chunks can be characterized probabilistically by the aforementioned Gaussian mixture model. This is very similar to what is done in voice/speech recognition, where words are broken down into phonemes. The use of the Hidden Markov Model (HMM) encapsulates all of these ideas (DBN's, Markov chains, Gaussian mixture emissions), and is ubiquitous in the voice/speech recognition community. However, these techniques have also been applied by other researchers interested in the field of motion classification and detection for video sequences. This field is related closely to our problem of motion trajectory classification, and they use similar intuitive arguments for applying formal probabilistic methods [3,4,5,6,7].

Much of the Bayesian approach applied to the current problem of modeling and classifying 6D trajectories for tele-operation under a time delay has been documented in previous work [1,2]. The common theme has been the use of the HMM as the fundamental modeling paradigm. A typical example of a tied-mixture HMM of the sort used in this work is shown in Fig. 4, in the context of a formal graphical model.

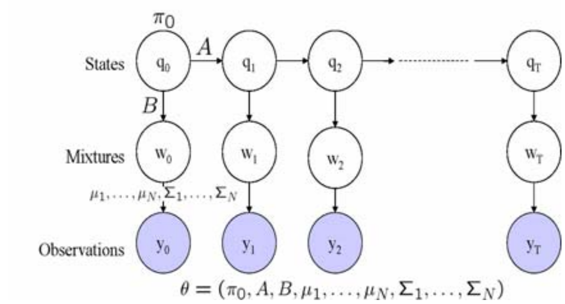

Figure 4

A tied-mixture model is used in order to allow us to represent the output observations as mixtures of Gaussians but yet still allows the model to scale in a reasonable fashion, providing us with sufficient free parametric flexibility. The shaded nodes shown in Figure 4 indicate that they are observed, while the unshaded nodes are "hidden," and need to be inferred from our observations. Some of the other important quantities from Fig. 4 can be described as follows: q_t - the state value at time t, w_t - the mixture value at time t, y_t - the observation vector at time t having dimension n, corresponding to the number of elements in the feature vector. The HMM parameters which are learned by Baum-Welch iteration (an optimization/training algorithm) are grouped together as θ, and are defined as follows:

$$
\begin{aligned}
\text{Prior (initial) probability} & \\
\text{distribution} &: \pi_0 \\
\text{Transition probability matrix} &: A \\
\Rightarrow a_{ij} &= p(q_{t+1} = j | q_t = i) \\
\text{Mixture weights} &: B \\
\Rightarrow b_{ij} &= p(w_t = j | q_t = i) \\
\Rightarrow \pi_0(i) &= p(q_0 = i) \\
\text{Mean of Gaussian distribution} & \\
\text{for mixture j} &: \mu_j \\
\text{Covariance matrix of Gaussian} & \\
\text{distribution for mixture j} &: \Sigma_j
\end{aligned}
$$

Some other important probabilities with regards to the tied mixture HMM shown in Fig. 4 are as follows:

$$
\begin{aligned}
\text{Emission (output) probability} &: \\
p(y_t | w_t = j) &= \mathcal{N}(y_t; \mu_j, \Sigma_j) \\
\text{Gaussian mixture probability} &: \\
p(y_t | q_t = i) &= \sum_{j=1}^{N} b_{ij} \mathcal{N}(y_t; \mu_j, \Sigma_j)
\end{aligned}
$$

Feature selection has proven to be a very important factor in how well the models characterize the experimental motion trajectory-based data. Additionally, it has played an important role in how consistently the models and real-time recall thresholds can be optimized to achieve the goals of no false alarms, minimum missed detections and time to prediction. Feature selection refers to the choice of several different combinations of feature vectors that can be used (i.e. what comprises y_t - the observation vector at time t having dimension n). These feature vectors act as templates for the observation sequences used to train and recall the hidden Markov models. "Recall" is a term that often refers to the use of the Viterbi algorithm, but can also be used to describe any algorithm that is used during the model testing phase, after the models have already been trained. During real-time recall of the models, HMMs trained on all tasks of interest (reaching for a particular object) are arbitrated based upon an algorithm to determine the "winning model," or which model best describes the streaming data.

Example feature vectors include subsets of the pose vector, which provides point of resolution (POR) data, a 4x4 homogeneous transform matrix representing the commanded position and orientation of the back of the robot's hand decomposed into position (x-y-z) and orientation (roll-pitch-yaw). Euclidean distances to the objects of interest being reached for can be used to form the feature vector as well. In addition to feature selection, the tradeoffs, advantages, and disadvantages of applying different recall methods were studied in detail, including their optimizations. Concerning the feature vectors, initially we considered looking only at Y and Yaw, which were discovered to provide good discrimination between the different types of trajectories to distinct objects. However, there were some problems with inconsistent initial conditions across multiple tele-operation sessions and multiple operators that biased the final error statistics. Optimization has not even completely resolved this problem, which may in part be due to the small size of the validation sets.

We'd also like to determine whether we can maximum the robustness of our final error statistics (% false alarms and missed detections) to minor variations in the experimental setup. In doing so, we've found that processing and testing new datasets based upon HMM prediction models trained on previous sessions are not sufficiently robust to changes in the experimental setup to yield reasonable error statistics. As a result, this gives us incentive to converge to a hybrid solution between the Bayesian approach and the deterministic approach that incorporates the best features of both. The first step in this process is to study and test a new approach that takes advantage of the spatial characterization of the trajectories rather than their time dependence. As such, we become much more reliant on distance to the objects of interest as not only an element within the feature vector, but as a method for discretizing the trajectory space, as one would discretize pixels in a video sequence.

3.3 CONVERGED METHOD

It is at this point that the convergence of the two research tracks began in earnest. The new method, which we call the approach manifold method, takes an object-centric view of an approaching hand. While the hand and object position information was encoded in a robot centric frame of reference before, the approach manifold method casts the hand position into an object centered frame composed of three values of position (XYZ) and 4 values of orientation represented by quaternions. Next., instead of explicitly comparing trajectories, this approach statistically models the distribution of approaching hand trajectories. It does this by discritizing the world by distance from the object. At a given distance, a mixture of Gaussians is found which represents the distributions of trajectories at that distance in the object centric frame. Intuitively, what this creates is a "fuzzy" multi-dimensional manifold shaped somewhat like a funnel. This is somewhat difficult to visualize, but in figures 5 and 6 we show a histogram based distribution on two of the axis, Roll and X, for the horizontal handrail target.

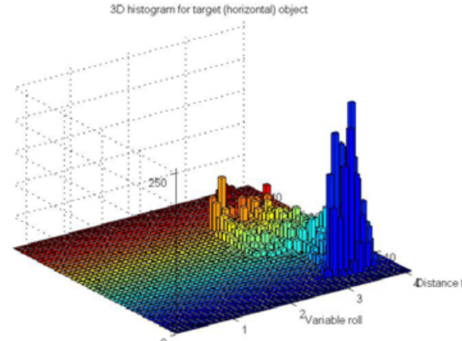

Figure 5: Distribution of Roll values versus distance.

Without being able to rotate the figure around, it is somewhat tricky to see, but in the Roll distribution there is actually a multi-modal distribution at greater distance, which collapses to a single clean distribution as the hand gets closer to the object. The X distribution is interesting because is stays tightly focused over all distances, which is not surprising because X is the major contributor to the distance metric.

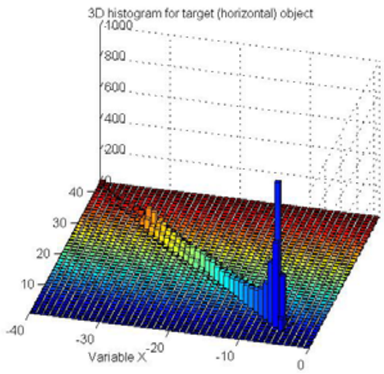

Figure 6: X versus distance

When a novel trajectory is encountered, it is compared against the manifolds associated with all valid objects, and the object with the best score is considered the goal of the motion. Having described the algorithm, it is important to note that this is under active development and that, for example, we are still using a simple histogram approach instead of the mixture of Gaussians described above. Despite this simplification, this approach is producing good results that are fairly robust to minor changes in handrail position.

4.0 RESULTS

As noted above, we have not yet fully implemented the Approach Manifold Method, and already we are getting some good results. Since this is still under development, we have not yet done a fully rigorous variational test, but we will show the results we do have. Not only are the results good on their own terms (100% correct predictions), they also compare favorably against previous results using the optimized HMM's.

Table 1 shows the results of a number of different tests against different datasets. In all these results, the models were trained on other data sets, and then shown either the 6/16 or the 10/7 data set. The Avg. Time Delta metric is the number of seconds from when the classification was made until the handrail was grabbed. Thus, the larger the value the better the algorithm did. For example, if the Delta were zero that would mean the classification was made only after the operator had made the entire gesture and was grabbing the handrail. This metric is good for comparing tests against the same data set, but because the operator might move quicker on some days (as they seem to be doing for 10/7) it is not a reliable measure across data sets. The tests shown in Table 1 are as follows:

HMM 6/16: The Optimized Hidden Markov Models described in the Stochastic section, and in earlier papers, is tested against the 6/16 data set.

AMM-Bin: The Approach Manifold Method, using binning to approximate a distribution.

AMM-Gaus: The Approach Manifold Method, using a Gaussian representation of trajectory distribution. It should be noted that this method was just completed and may improve over the next few days.

	HMM 6/16	AMM-Bin 6/16	AMM-Gaus 6/16	AMM-Gaus 10/7
Total Trials	28	28	28	20
Classified Correct	26	28	28	20
Failed to Classify	2	0	0	0
Classified Wrong	0	0	0	0
Avg. Time Delta	6.9	7.9	7.2	5

Table 1: Results Metrics

As can be seen from this chart, the new Approach Manifold Method does a better job of classifying the trials, and does so quicker than the previous HMM method. What the chart doesn't show is even more impressive. The 10/7 data is special because the handrail orientations are reversed: the vertical and horizontal handrails have swapped position. The models were trained on earlier data, and then shown the 10/7 data and they were still able to correctly classify all the trials. Since one of the goals of the Approach Manifold Method was to let the models be more independent of position and orientation, these early results show that we are working in the right direction.

5.0 FUTURE WORK & DISCUSSION

One feature that all these approaches share is that a single model is not general for any arbitrary location of the object. This makes sense since a trajectory is fundamentally the path *from* some known point in space, *to* some known point in space. That being said, it is clear that the different models being produced are valid for some amount of deviation from the points they were trained on. The approach manifold method explicitly attempts to get away from this limitation and encode the approach manifold of a hand towards the object from any arbitrary location. Yet even in this case there are limitations to how general a single model can be because unusual approaches, while valid, may also look like outlier data and will be given low score. Thus, for all these approaches it can be said that, at some granularity, there is a region for which a specific model is valid. Thus, in order to have a complete solution which could recognize grasp attempts anywhere in the workspace of the robot, one would need very many models to account for all the possible positions of objects. This is a reasonable approach, though it comes at the expense of requiring large amounts of example data. One thing we would like to look at in the future is how to automatically generate appropriate models for any arbitrary object location.

A common first thought is, why not create a path planner which can generate paths to any arbitrary location. The problem with this approach is that it is not clear that a path planner will generate a trajectory that looks like what a human would decide to do. The goal here is not optimal path generation, but rather the goal is interpreting human gesture. Thus, what we would like to do is to develop enough models of human motion that we can start automatically generating motion models that mimic human gestures.

Of course, this leads us back to one of our original questions: Is it possible to automatically generate autonomous behavior by modeling human action? Ideally this could be done directly from the teleoperation example data and would never require the creation of an explicit path planner. An important, and unanswered, question is: given two different models which both predict the human action equally well, will one of these models be better at generating autonomous control? What are the properties of a good generative model? And, finally, how can you verify that a model is robust enough that its autonomous control will be safe and reliable? This is a long term research agenda, which, if successful, could have a huge impact on lowering the barrier to the creation of autonomous control.

We would like to close by thanking the members of the Dexterous Robotics Laboratory at Johnson Space Center for their hard work and ongoing support of this research. Without them, we would have no data.

6.0 REFERENCES

[1] K. Wheeler, R. A. Martin, V. SunSpiral, and M. Allan. “Predictive interfaces for long-distance tele-operations”, In *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Munich, Germany, September 2005.

[2] R. A. Martin, K. Wheeler, M. Allan , and V. SunSpiral. “Optimized Algorithms for Prediction within Robotic TeleOperative Interfaces”, Submitted to *Human-Robot Interaction Conference (submitted for review, pending acceptance)*, Salt Lake City, UT, March 2006.

[3] Porikli, F.M., “Trajectory Distance Metric Using Hidden Markov Model Based Representation”, In *European Conference on Computer Vision (ECCV)*, May 2004.

[4] Christoph Bregler. “Learning and Recognizing Human Dynamics in Video Sequences” *IEEE Conf. on Computer Vision and Pattern Recognition,* 1997.

[5] Yue Zhou, Yihong Gong, and Hai Tao, “Background modeling using time dependent Markov random field with image pyramid,” in *Proc. IEEE Motion’05*, January 2005.

[6] D. Makris and T. Ellis, “Spatial and probabilistic modeling of pedestrian behavior,” in *Proc. Brit. Machine Vision Conf.* Cardiff, U.K., vol. 2, Sep. 2002, pp. 557-566.

[7] R. Fablet, P. Bouthemy. “Motion recognition using spatio-temporal Random Walks in Sequence of 2D Motion-Related Measurements.” In *IEEE Int. Conf. on Image Processing, ICIP'2001*, Pages 652-655, Thessalonique, Grèce, Octobre 2001.

[8] W. Bluethmann, R. O. Ambrose, M. A. Diftler, S. Askew, E. Huber, M. Goza, F. Rehnmark, C. Lovchik, and D. Magruder. “Robonaut: A robot designed to work with humans in space.” *Autonomous Robots*, 14(2-3):179–198, 2003.

[9] R. A. P. II, C. L. Campbell, W. Bluethmann, and E. Huber. “Robonaut task learning through teleoperation.” *In ICRA*, pages 2806–2811. IEEE, 2003

[10] Wheeler K., Jorgensen C., “Gestures as Input: Neuroelectric Joysticks and Keyboards”, *IEEE Pervasive Computing*, Vol. 2, No. 2, April-June, 2003.