# Superpositioning of Behaviors Learned Through Teleoperation

Christina Louise Campbell, Richard Alan Peters, II, *Member, IEEE*, Robert E. Bodenheimer, *Member, IEEE*, William J. Bluethmann, Eric Huber, and Robert O. Ambrose

***Abstract*—This paper reports that the superposition of a small set of behaviors, learned via teleoperation, can lead to robust completion of an articulated reach-and-grasp task. The results support the hypothesis that a robot can learn to interact purposefully with its environment through a developmental acquisition of sensory-motor coordination. Teleoperation can bootstrap the process by enabling the robot to observe its own sensory responses to actions that lead to specific outcomes within an environment. It is shown that a reach-and-grasp task, learned by an articulated robot through a small number of teleoperated trials, can be performed autonomously with success in the face of significant variations in the environment and perturbations of the goal. In particular, teleoperation of the robot to reach and grasp an object at nine different locations in its workspace enabled robust autonomous performance of the task anywhere within the workspace. Superpositioning was performed using the *Verbs and Adverbs* algorithm that was developed originally for the graphical animation of articulated characters. The work was performed on Robonaut, the NASA space-capable humanoid at Johnson Space Center, Houston, TX.**

***Index Terms*—Dexterous manipulators, intelligent robots, nonlinear functions, robot programming, telerobotics.**

## I. Introduction

THE paper deals with the problem of enabling a robot to learn from experience by building models of the dynamics of its own sensory and motor interactions with objects and tasks [1]. This interaction is initially provided by fine-grained teleoperator inputs. Over time, information gleaned from teleoperator guidance is compiled into autonomous behaviors, so that the robot can perform tasks on its own, and so that the level of discourse between operator and robot can become more abstract.

### A. Approach

The approach described here builds on the self-organization of sensory-motor information in response to a robot's actions within a loosely structured environment. In [2], Pfeifer reported that sensory data and concurrent motor-control information recorded as a vector time series formed clusters in a sensory-motor state space. He noted that the state-space locus of a cluster corresponded to a class of motor action taken under specific sensory conditions. In effect, the clusters described a categorization of the environment with respect to sensory-motor coordination (SMC).

An exemplar of an SMC cluster corresponds at once to a basic behavior (as used implicitly by Brooks [3] and later defined by Matarić [4]) and to a competency module in a spreading activation network [5]. The latter is a specific example of a more general class of topological, action-map representations of an environment [4] which can controlled by discrete-event dynamical systems (DEDS) [6], [7], with transition probabilities given by Markov decision processes. If the state space is parameterized by time, the clusters are collections of trajectories, and an exemplar is a single representative trajectory through the space.

Thus, if a robot is controlled through an environment to complete a task while recording its SMC vector time series, the result is a state-space trajectory that is smooth during the execution of a behavior, but that exhibits a corner or a jump during a change in behavior (an *SMC event*). From this, a DEDS description of the task can be formed as a sequence of basic behaviors and the transitions between them. The task is learned in terms of the robot's own sensors, actuators, and morphology.

The autonomous execution of fixed motor trajectories by a DEDS controller that changes state in response to SMC events will fail if the operating environment differs significantly from the learning environment. On the other hand, if a set of trajectories is learned that bounds or covers the variations of the task, the task might be performed successfully under more conditions. In particular, a new situation might be successfully negotiated through a superpositioning of the bounding trajectories.

This paper reports the results of learning to reach toward and grasp a vertically oriented object at an arbitrary location within the robot's workspace by superpositioning a set of SMC state-space trajectories that were learned through teleoperation. The ideas behind the procedure are based on a number of assumptions: 1) when a teleoperator performs a task, it is her/his SMC that is controlling the robot. So controlled, the robot's sensors detect its own internal states and those of the environment as it moves within it. Thus, the robot can make its own associations between coincident motor actions and sensory features as it is teleoperated; 2) in repeating a task several times, a teleoperator will perform similar sequences of motor actions whose dynamics will depend on his/her perception of similar sensory events that occur in a similar sequence. As a result, the robot

Manuscript received December 31, 2003; revised December 13, 2004. This paper was recommended for publication by Editor F. Park upon evaluation of the reviewers' comments. This work was supported in part by DARPA-IPTO under Grant DASG60-99-1-0005, and in part by NASA-JSC under Grants NAG9-1428, NAG9-1446, and NAG9-1515.

C. L. Campbell, R. A. Peters II, and R. E. Bodenheimer are with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235-1824 USA (e-mail: christina.l.campbell@vanderbilt.edu; Alan.Peters@vanderbilt.edu; bobbyb@vuse.vanderbilt.edu).

W. J. Bluethmann, E. Huber, and R. O. Ambrose are with the Lyndon B. Johnson Space Center, National Aeronautics and Space Administration, Houston, TX 77058 USA (e-mail: william.j.bluethmann1@jsc.nasa.gov; eric@roboteyes.com; robert.o.ambrose@nasa.gov).

Digital Object Identifier 10.1109/TRO.2005.861485

will detect a similar set of SMC events during each trial. Therefore, each trial can be partitioned into *SMC episodes*, demarcated by the common SMC events; 3) sensory events that are salient to the task will occur in every trial; sensory signals that differ across trials are not significant for the task and can be ignored. By averaging the time series for each episode pointwise over the trials, a canonical representation of the motor-control sequence can be constructed. As a result of the averaging, true events in the sensory signals will be enhanced, and those that are random will be suppressed.

This approach does not form an approximation of the inverse kinematics (IK) of the manipulator. Rather, it learns six-axis spatial end-effector trajectories that are sent as position commands to the robot, which computes its own IK. This somewhat higher level approach extracts Cartesian motion and pose trajectories, finger position trajectories, and sensory state information to create a sensory-motor (or, perhaps more accurately, a sensory-motion) description of the task.

### B. Related Work

This work extends that reported by Peters *et al.* in [8], where a single trajectory was learned over six trials that could later be performed autonomously with success in the face of small variations in the environment or perturbations of the goal. In addition to Pfeifer [2], many others have studied the extraction of SMC parameters, including Cohen [9], [10], Grupen [11], and Peters [12]. Like Pfeifer, Cohen concentrates on learning categories from random behaviors. However, he manually designates the episode boundaries, and uses categorization techniques to find similarities between the episodes. While such clustering may become important as more tasks are incorporated, the behaviors in this paper can be automatically clustered by their locations in the task sequence.

Grupen experimented with DEDS of parallel controllers that are, in some respects, quite similar in theory, to the *autonomous* parts of the work described here. His systems use, but do not learn, low-level SMC trajectories for motion control, and have mainly focused on grasping and dexterous manipulation. Grupen uses collections of closed-loop controllers preselected and preprogrammed for variations on a task. The controllers are invoked through a DEDS implemented as a Markov decision process to complete a task. The approach enables an articulated manipulator to perform more complex manipulation tasks than that described in this paper. However, the controllers must be designed manually, they are not learned.

In many respects, however, our work is fundamentally different from that of Grupen. We create exemplar trajectories of the end-effector (hand point-of-reference (POR), six-axis pose) and finger joint positions from a small number of repeated, teleoperated trials of the same task performed at different locations in the workspace. The trajectories exist in a sensory-motor state space, since the target position is specified by vision, and the grasp elicits a response from force-torque and haptic sensors. After offline analysis, these trajectories are recombined in real time to perform the same task at any location within the workspace. With regard to manipulation, there is little doubt that this approach is not as robust as that of Grupen; ours, however, requires no controller design. Grupen's work appears to be compatible with that described here, as discussed in Section VI.

The use of motion data to plan robotic motion is a problem that has been studied by Matarić [13], Jenkins and Matarić [14], Ude *et al.* [15], Pollard *et al.* [16], and Atkeson *et al.* [17]. Matarić and Jenkins have enabled a simulated humanoid to learn unconstrained motion patterns from human motion-capture data. (Our work modifies one of their segmentation and data normalization procedures.) Jenkins has further studied the creation of new motions through the interpolation of learned trajectories using Isomap [18]. In [15]–[17], Ude, Pollard, Atkeson, and their coauthors describe a system that learns from demonstration through vision. They enabled the robot DB (from the Kawato dynamic brain project) to watch a person perform a task several times and then do the task itself. Through various analysis techniques, the person's motions as perceived by the robot vision system are mapped onto a set of motion primitives. These are similar in concept to Matarić's basis behaviors [4], but contain only motor components. The motion primitives themselves do not contain sensory information. A task is learned by selecting those primitives that combine to best match the image of the perceived motion. The combination is constrained by a model of human motion and some regularization terms. Thus, perception itself becomes an optimization process that tries to find an underlying motor program to mimic the motion. Through these approaches, DB can learn a number of complex full-body articulated motion tasks.

In our opinion, learning from observation is an important and difficult problem. The approach described here is, in some respects, simpler than those described above, and thereby avoids some of their difficulties. Since our robot learns sensory-motor trajectories through teleoperation, it learns its own motions and sensory responses directly. It does not have to map remote observations of a far more dexterous machine, a person, into its own limited degrees of freedom (DOFs). Moreover, the authors cited above have enabled their systems to learn large-scale, relatively unconstrained motions such as dancing, punching, and playing ping pong. Our approach enables learning of the more highly constrained motions required for grasping. Another difference is that they learn only the motion, not the accompanying sensory percepts.

## II. Robot Hardware

The experiments for this paper were performed on Robonaut, NASA's space-capable, dexterous humanoid robot (see Fig. 1). Robonaut was developed by the Dexterous Robotics Laboratory (DRL) of the Automation, Robotics, and Simulation Division of the NASA Engineering Directorate at Lyndon B. Johnson Space Center in Houston, TX [19]. In size, the robot is comparable to an astronaut in an extra-vehicular activity (EVA) suit. Each 7-DOF Robonaut arm is approximately the size of a human arm, with similar strength and reach, but with a greater range of motion. Each arm mates with a dexterous end-effector, a 12-DOF hand, to produce a 19-DOF upper extremity. The robot has manual dexterity sufficient to perform a wide variety of tasks requiring the intricate manipulation of tools and other objects.


Fig. 1. Robonaut, NASA's space-capable humanoid robot.

TABLE I
SIGNALS RECORDED FROM ROBONAUT

| Signal | Dimension |
|---|---|
| End-effector position, actual | 3 |
| End effector rotation mat, actual | 9 |
| Arm orbit angle, actual | 1 |
| End-effector position, requested | 3 |
| End effector rotation mat, req'd | 9 |
| Arm orbit angle, requested | 3 |
| Arm 3-axis force on wrist | 3 |
| Arm 3-axis torque on wrist | 3 |
| Arm 3-axis force on shoulder | 3 |
| Arm 3-axis torque on shoulder | 3 |
| Arm joint positions | 7 |
| Arm joint torques | 7 |
| Hand force on fingers | 5 |
| Hand joint positions | 12 |
| Hand joint torques | 12 |
| Hand tactile sensors | 19 |
| Visual object name | 1 |
| Visual object pose | 6 |
| Teleoperator voice command | 1 |

Robonaut has many sensors. These include a color, stereo camera platform embedded in a head mounted on a 3-DOF neck, and binaural microphones located on opposite sides of the head, parallel to the stereo camera baseline. The two hand/wrist modules contain 84 sensors for feedback and control, 60 of which are analog and require signal conditioning and digitization. Each DOF has a motor position sensor, a joint force sensor, and a joint absolute position sensor. The two arm modules contain 90 sensors, 80 of which are analog. Each actuator contains a motor incremental position sensor, redundant joint torque sensors, redundant joint absolute position sensors, and four temperature sensors distributed throughout the joint. Each arm employs relative optical encoders in five of its joints. The encoders reside on the motor side of the gear train and have resolutions ranging between 200 and 1000 counts per degree of arm motion. The two wrist joints employ resolvers integrated into the motor assemblies (see [20] for a more detailed description of the Robonaut mechatronics.) A variety of data signals are recorded from Robonaut during teleoperation. These are listed in Table I. They are recorded at a nominal rate of 50 Hz, but some signals, such as those produced by vision, are slower.

Robonaut is physically capable of autonomous operation. At this time, however, it is most often controlled directly by a person via teleoperation. In this mode, every motion made by Robonaut reflects a similar motion made by the operator, who perceives the robot's workspace through full-immersion virtual reality. The operator wears a helmet that enables her or him to see through the robot's stereo camera head and to hear through the robot's binaural microphones.[1] Sensors in gloves worn by the operator determine Robonaut's finger positions. Six-axis Polhemus sensors [21] on the helmet and gloves determine the robot's arm and head positions. An operator guides the robot using only vision; there is neither direct haptic nor direct force feedback from robot to person. That is, the robot's haptic sensors do not transmit touch sensations to the operator, nor do the force sensors project forces onto the operator's gloves.[2]

Each 7-DOF arm is commanded independently by specifying the 6-D Cartesian pose (position and orientation) of its hand's POR. The POR is located on the back of the hand so that it corresponds to the location of the Polhemus sensor on the corresponding teleoperator glove. Usually, the seventh DOF is computed by an IK algorithm that minimizes joint velocities. The operator can, by specifying an angle, command the elbow orbit position if the IK algorithm computes one that is problematic for the desired motion in the current environment. The elbow position is specified by the angle between the plane formed by the shoulder, elbow, and wrist, and the vertical plane of right–left symmetry of the robot.

Robonaut's arm–hand systems have a high-bandwidth dynamic response (the servo control loop operates at 50 Hz) that enable it to move quickly, if necessary, under autonomous operation. During teleoperation, however, the response of the robot is slowed to make it less susceptible to jitter in the arms of the teleoperator, and to make it safe for operation around people, either unprotected on the ground or in pressurized EVA suits in space. The slowdown is, effectively, to a 10-Hz loop with the teleoperator. The purposeful limitation of maximum joint velocity affects not only the motion analysis described below, but also the superposition of learned behaviors, especially with respect to the time warping of component behaviors (cf. Section III-C).

The Robonaut stereo vision system uses object shape to determine the six-axis pose of well-defined objects, such as wrenches and tables, as well as variable-form structures, such a human limbs [22]. The robot's field-of-view (FOV) is preprocessed using patch correlation on Laplacian-of-Gaussian (LoG)-filtered image pairs to generate 3-D range maps, as well as binary, 2-D range maps taken over a series of range intervals. Initially, four DOFs of a known object are estimated roughly

[1]The robot has microphones for terrestrial use; radio would be used in space.

[2]Several approaches to supplying such feedback have been tried. None of them improved the performance of the teleoperator acting through vision alone. Indirect feedback in the form of visual force displays are being tested.

through an efficient matching of large sets of 2-D silhouette templates against the 2-D range maps. This estimate is refined to give a more precise pose estimate in all six DOFs. The strongest silhouette matches are used to seed a process which matches 3-D sets of contour templates against 3-D range maps. Although considerably more expensive computationally than 2-D, a 3-D process is necessary for the robot to handle and manipulate objects. A high level of precision is obtained in real time, because most of the environment is filtered out by the much faster 2-D silhouette matching process.

After low-pass filtering, the outputs with a time constant of about 0.2 s (finite-impulse response (FIR) averaging), the vision system is accurate to within 0.003 m and 2.0° in the workspace of the robot. This is as measured relative to an object with a calibrated pose. The general accuracy of the system in deployment is within about 0.015 m and 5°. Currently, most estimation error is caused by the correlation mismatches on surfaces that are metallic (reflective) or low in texture (e.g., a black plastic drill handle). The system outputs the poses of recognized objects within its FOV at a rate of about 7 Hz. The overall latency through the system (photons hitting lens to vision system Ethernet output) is about 0.22 s. Latency from vision output to robot actuation is approximately 0.38 s.

Although the teleoperator may be unaware of most of it, all sensory data is available in real time for the robot's computers to analyze. In particular, from the data, the robot can learn the sensory characteristics of tasks performed via teleoperation. That information can, in turn, be used by Robonaut in autonomous operations.

## III. Behavior Superposition

There were four phases in the data gathering and analysis for this learning task.

1) A teleoperator controlled the robot through the tasks that would serve as examples. Five trials at each of nine locations were performed of a reach-and-grasp of a vertically oriented object (a wrench). As the teleoperator performed these example motions, Robonaut's sensory data and motor command streams were sampled and recorded as a vector time series or signal.
2) The SMC events common to all trials were found and used to partition the signal into episodes. The episodes were time-warped, so that the $j$th episode in the $k$th trial had the same duration (and number of samples) as the $j$th episode in every other trial (cf. Section III-C.)
3) The signals were averaged over all five trials at each location to produce a canonical, sensory-motor data, vector time series for each location. This approach is similar both to that of Jenkins and Matarić [14] and to those analyzed by Cohen [10].
4) These generalized motions were combined using the process described by Rose *et al.* [23], called *Verbs and Adverbs* (VaV).

When the process completed, the resulting set of parameters could be saved to file and then used to create a general representation of the task that was adaptable under real-time conditions.

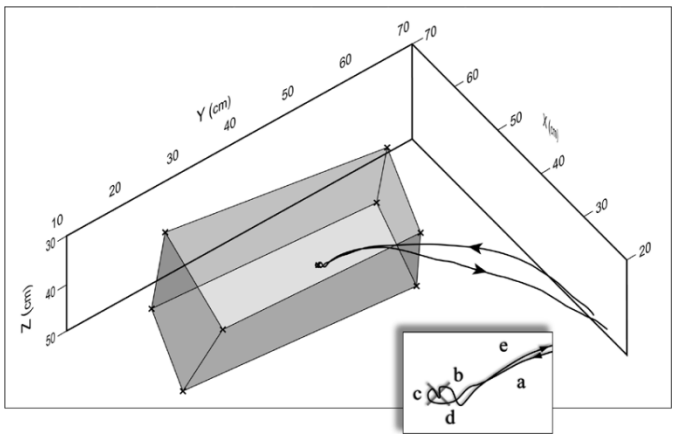



Fig. 2. Plot of the nine exemplar object locations from the robot's viewpoint. The dark-shaded regions are the bounding surfaces of workspace volume delineated by eight of the points. The contour is the end-effector trajectory from one trial of the experiment, where the object was at the ninth position, in the center of the box. The arrowheads indicate the direction of motion. The inset shows the grasp, hold, and release episodes in greater detail. They are: (a) reach, (b) grasp, (c) hold, (d) release, (e) withdraw.

### A. Teleoperation

The task performed by the teleoperator was, with the right arm, to reach forward to a wrench affixed to a frame, grasp the wrench, hold it briefly, release it, and withdraw the arm. The frame made it possible to reposition the wrench as needed, while keeping it steady during task performance. For the purposes of these experiments, the wrench was positioned in a reachable, nearly vertical position. Nine example locations were chosen. Eight of these were positioned approximately at the corners of a virtual box that defined the limits of the reachable workspace. The ninth was a point near the middle of the box. Five trials were repeated at each of the nine locations.

Fig. 2 shows a 3-D plot of the locations, with lines drawn to indicate the box, which is a warped parallelepiped. The curve in the middle is a plot of the end-effector's POR throughout one of the five trials where the object was at the central position. The box is depicted from the viewpoint of one of Robonaut's cameras. The coordinate frame used for all Cartesian locations in this paper is centered on Robonaut's chest. The $x$ axis points out, the $y$ axis points right, and the $z$ axis points down. Note in the figure that the $y$ dimension of the box is much longer than the $x$ and $z$ directions.

### B. Segmentation

A vector time series

$$\mathbf{v}_i(t) = \left[\, \mathbf{s}_{i,1} \quad \cdots \quad \mathbf{s}_{i,N} \,\right]^T (t) \tag{1}$$

was recorded during each teleoperated trial of the task. The time series contained $N$ separate signals, $\mathbf{s}_{i,j}(t)$ from the various sensors and actuators. The signals and their dimensions are listed in Table I.

The time-series data from the experiment was manually segmented into 45 trials, according to markers embedded in the voice channel of the robot's data stream. Then each trial was

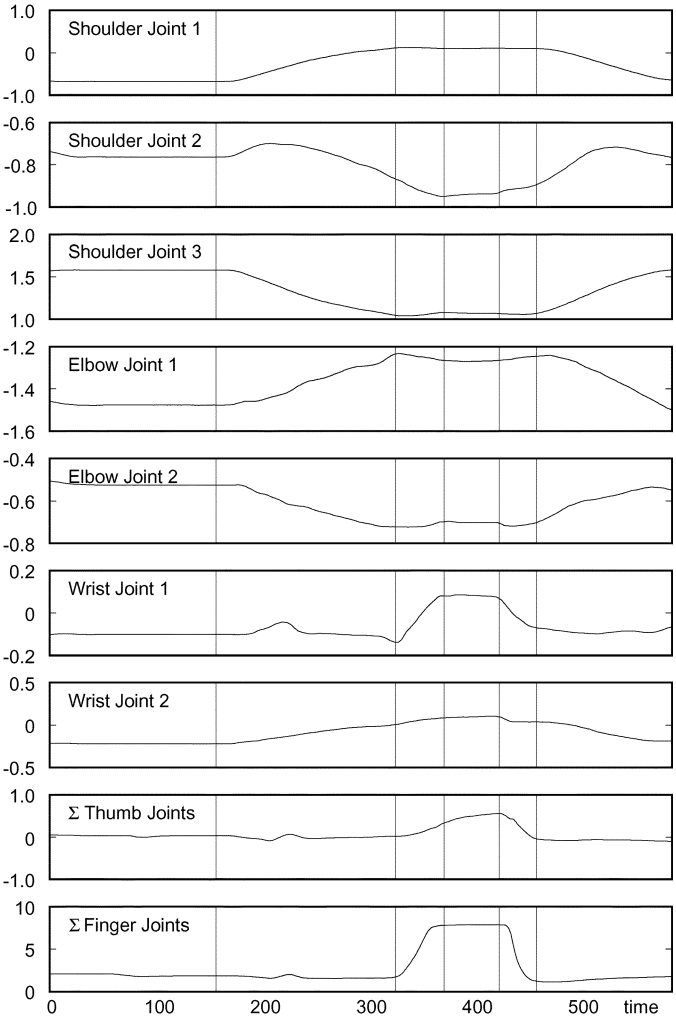



Fig. 3. Right arm and hand joint positions during a single trial at a single location plotted with respect to time. The three thumb DOFs are summed, as are the nine finger DOFs. The episode boundaries are demarcated as vertical lines. They are (L to R) the period prior to motion, reach, grasp, hold, release, withdraw.

partitioned into five SMC episodes[3] (reach, grasp, hold, release, withdraw) demarcated by SMC events that were found through an analysis of the mean-squared velocity (MSV) of the joint angles $\alpha_i$

$$z = \sum_i \dot{\alpha}_i^2 \tag{2}$$

the sums of the squares of all the joint velocities in the arm–hand system [24]. Changes in velocity are apparent in the joint position profiles of a single trial, as shown in Fig. 3.

An SMC event was defined as the beginning or end of a sufficiently large peak in the MSV, since that corresponded to a significant acceleration or deceleration in the arm–hand system. The beginning of a peak was marked at time $t_0$ if: 1) $z(t_0)$ exceeded a threshold $c$; and 2) $z(t)$ exceeded $15c$ at a later time $t_1 > t_0$ before 3) falling below the lower threshold $c$ again at a still later time $t_2 > t_1$. That is, an SMC event was marked at time $t_0$ if

$$z(t_0 - 1) < c \text{ AND } z(t_0) \geq c \text{ AND } z(t_1) > 15c \tag{3}$$

for some $t_1 > t_0$, providing that $z(t) > c$ for all $t \in (t_0, t_1)$. The end of the peak was detected at time $t_2$ if

$$z(t_2 - 1) > c \text{ AND } z(t_2) \leq c \text{ AND } z(t_1) > 15c \tag{4}$$

for some $t_1 < t_2$, providing that $z(t) > c$ for all $t \in (t_1, t_2)$.

A threshold of $c = 0.02 \text{ deg}/\text{s}^2$ was derived empirically as the fifth percentile of a sampled distribution of measured accelerations. That is, let $\hat{z}$ be the largest value of $z$ measured throughout the trials. The value of $c$ was increased from $0.001\hat{z}$ to $0.1\hat{z}$ in increments of $0.001\hat{z}$. For each $c$, define the set of times $T_c = \{t | \hat{z}(t) > c\}$. Then compute the mean and standard deviation of the set $Z_c = \{\hat{z}(t) | t \in T_c\}$. As $c$ increases, the number of points in $Z_c$ decreases, while the mean value of $Z_c$ increases. Moreover, the data in $Z_c$ becomes dominated by values from the peaks which vary from one another more than do those points that are closer to zero. As a result, the standard deviation of $Z_c$ increased (roughly) logarithmically with $c$, but levelled off at an asymptote of approximately $0.6\hat{z}$. The 95th percentile, $0.57\hat{z}$, was reached at $c \approx 0.02$. The factor of 15 was used for the upper threshold because it yielded the number of episodes that were expected (see Fig. 4).

The MSV was found to be an excellent indicator of the grasp, hold, and release events if the hand joint velocities were included in it. It was not reliable in detecting those events if only the arm joint velocities were included. The vector time series between two SMC events were taken as SMC episodes that corresponded to distinct behaviors.

### C. Time Warping: Normalization and Averaging

Once the segmentation of the data was complete, the SMC episodes that comprise the task were time-warped through resampling to have a duration equal to the average duration of the 45 trial episodes. Then for each of the nine locations, the average vector time series was computed from the five corresponding trials. For example, the *reach* behavior averaged 150 time steps across the 45 trials. Each of the time series that comprised the *reach* episodes was time-warped and resampled to have length 150. The five *reach* episodes from the five trials at each location were averaged to create nine exemplar *reach* episodes, each with 150 samples in duration. Fig. 5 shows the trajectories from the five trials at one location and the average of the five.

In these experiments, we used a pointwise linear averaging of the time-normalized sensory-motor episodes to produce an exemplar for the task. The effect of averaging the five trials at each location was to enhance those characteristics of the sensory and motor signals that were similar in the five, and to diminish those that were not. One could use the median value at each point, if a minority of the exemplars showed deviations due to noise or other mismeasurement. Moreover, signals that exhibit nonlinear behavior with respect to time (e.g., a binary or on/off signal) would require a median or other order-statistic filter to preserve the signal characteristics. Certainly, averaging would produce a skewed result if one of the exemplar episodes were

[3]In this work, the trials were demarcated manually, and each trial was segmented automatically into episodes. The trials could have been likewise extracted automatically, but were not, since episode extraction was the focus of the work.

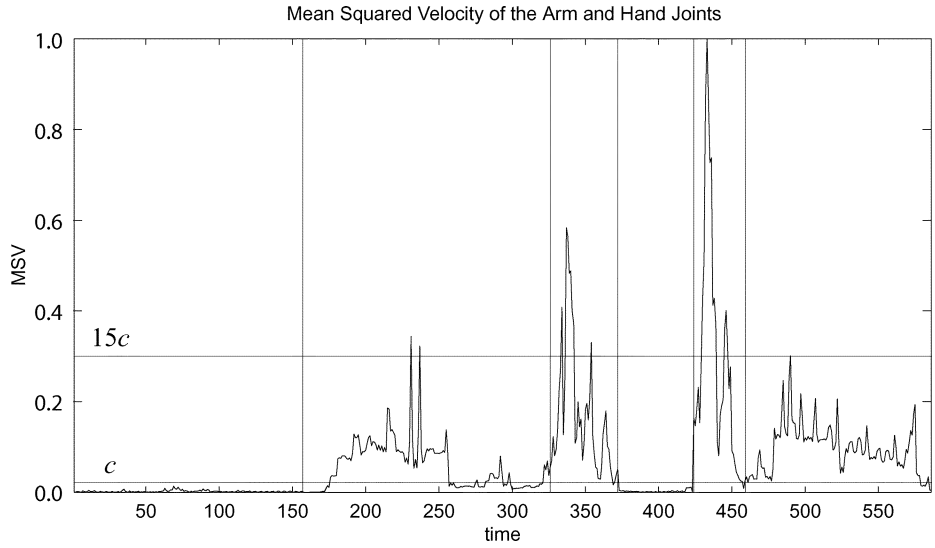



Fig. 4. Plot of the instantaneous mean-squared joint velocities $z$ for one teleoperated trial at one object location. The lower and upper thresholds, $c$ and $15c$, are indicated by horizontal lines. The episode boundaries (pre-motion, reach, grasp, hold, release, withdraw) are demarcated by vertical lines.

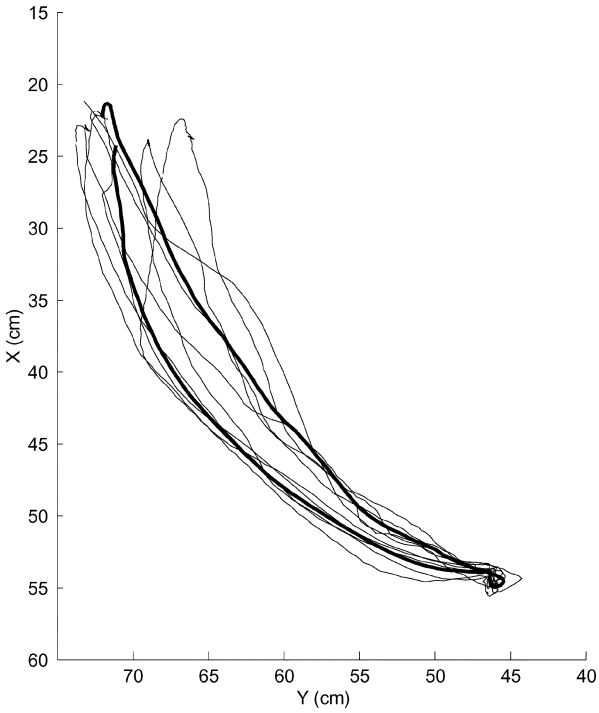



Fig. 5. Thin lines: the hand POR trajectories in the $xy$ plane over the five trials at one location. Thick line: the exemplar POR trajectory constructed as the pointwise average of the five trials.

significantly different from the others to be combined with it. However, it was a premise of this work that such episodes would not differ significantly from each other in their salient features. If that premise were incorrect, the characterization of a behavior through the type of analysis described here would be of dubious value. But the premise was found to hold throughout these particular experiments.

Through the four-step procedure (cf. Section III), nine sensory-motor state-space trajectories were created. These were taken to be the exemplars of the clusters formed by the five trials of the reach-and-grasp task at each of the nine locations.

Given the dynamics of Robonaut under teleoperation (its maximum velocity is limited), the durations of the episodes are relatively long and the sampling rate well exceeds the Nyquist limit. Thus, the salient sensory-motor characteristics are well represented in all the trials at each of the locations, and time-warping for episode normalization preserves those characteristics. This would not necessarily be the case if the sampling rate were near the Nyquist limit, and some of the episodes were of short duration.

### D. Superposition Using Verbs and Adverbs

After the resampling and averaging of the sensory-motor data from the example tasks, the data were analyzed to characterize the motions that would enable Robonaut to reach toward and then grasp, with its right hand, a vertically oriented wrench anywhere within its workspace. This was done with an interpolation method called Verbs and Adverbs, developed in the computer graphics community by Rose *et al.* [23]. The following description is an adaptation for robotics of the algorithm from that paper. Table II lists symbols used in the description.

A *verb* in this implementation of the algorithm is the motion component of a task exemplar, its motion trajectory in the sensory-motor state space. Let $\mathcal{T} \subset \Re$ be a set of consecutive sample times. Let $\mathcal{S}$ represent the motion state space $\dim(\mathcal{S}) = N_s$. Define $\mathbf{m}_i(t) : \mathcal{T} \rightarrow \mathcal{S}$ to be the value at time $t$ of the motion-state trajectory of the $i$th exemplar. Let $\mathbf{m}_i \in \mathcal{S} \times \mathcal{T}$ represent $\mathbf{m}_i(t)$ over all time, the trajectory in its entirety. Let $\mathbf{m}$ represent an arbitrary motion-state trajectory.

An *adverb* $\mathbf{p}$ is an $N_a$-dimensional vector in adverb space $\mathcal{A}$ that characterizes in some way a particular motion trajectory $\mathbf{m}$. The adverb is a specific parameterization of the motion trajectory. Thus, by implication, there exists a mapping

$$\Phi : \mathcal{A} \rightarrow \mathcal{S} \times \mathcal{T} \tag{5}$$

such that

$$\mathbf{m}_i = \Phi[\mathbf{p}_i] \tag{6}$$

TABLE II
SYMBOLS FOR VAV ALGORITHM

| Symbol | Dimension | Meaning |
|---|---|---|
| $\mathcal{A}$ | $N_a$ | adverb space, $N_a$ = no. comp'ts. per adv . |
| $\mathcal{E}$ | $N_e$ | exemplar state space, $N_e$ = no. of exemps. |
| $\mathcal{S}$ | $N_s$ | motion state space, $N_s$ = no. of states |
| $\mathcal{S}\times\mathcal{T}$ | $N_s+1$ | motion state trajectory space |
| $\Phi[\cdot]$ | $(N_s+1)\times N_a$ | exact mapping from $\mathcal{A}$ to $\mathcal{S}\times\mathcal{T}$ |
| $\Phi(t)[\cdot]$ | $N_s\times N_a$ | exact mapping from $\mathcal{A}$ to $\mathcal{S}$ at time $t$ |
| $A(t)$ | $N_s\times(N_a+1)$ | LMS approx. of $\Phi(t)$, rows: $\mathbf{a}_j^T(t)$ |
| $M(t)$ | $N_s\times N_e$ | exemp. state mat: cols: $\mathbf{m}_i(t)$, rows: $\mathbf{n}_j^T(t)$ |
| $\bar{M}(t)$ | $N_s\times N_e$ | resid. mat: cols: $\bar{\mathbf{m}}_i(t)$, rows: $\bar{\mathbf{n}}_j^T(t)$ |
| $P$ | $N_a\times N_e$ | adverb exemplar matrix, cols: $\mathbf{p}_i$ |
| $P^h$ | $(N_a+1)\times N_e$ | hom. adverb exemplar matrix, cols: $\mathbf{p}_i^h$ |
| $Q(t)$ | $N_e\times N_s$ | interp. mat., col. $j$: RBF amps. for state $j$ |
| $R(t)$ | $N_e\times N_e$ | matrix of RBF vals. at adv. locs at time $t$ |
| $\mathbf{a}_j(t)$ | $N_a+1$ | affine coeff. vector that maps $\mathbf{p}_i^h$ to $m_{ij}(t)$ |
| $f(t)[\cdot]$ | $N_s$ | interp. function that maps $\mathbf{p}_i$ to $\mathbf{m}_i(t)$ |
| $\mathbf{m}$ | $N_s+1$ | trajectory of motion state vector; $\subset\mathcal{S}\times\mathcal{T}$ |
| $\mathbf{m}_i$ | $N_s+1$ | traj. of motion state exemplar $i$; $\subset\mathcal{S}\times\mathcal{T}$ |
| $\mathbf{m}(t)$ | $N_s$ | motion state vector at time $t$; $\subset\mathcal{S}$ |
| $\mathbf{m}(\mathbf{p};t)$ | $N_s$ | state vect. of motion with adv. $\mathbf{p}$ at time $t$ |
| $\tilde{\mathbf{m}}$ | $N_s+1$ | LMS approximation of motion state traj. |
| $\bar{\mathbf{m}}$ | $N_s+1$ | motion state residual: $\mathbf{m}-\tilde{\mathbf{m}}$ |
| $\mathbf{n}_j(t)$ | $N_e$ | vect. of state $j$ from all exemps. at time $t$ |
| $\tilde{\mathbf{n}}_j(t)$ | $N_e$ | LMS approximation of $\mathbf{n}_j(t)$ |
| $\bar{\mathbf{n}}_j(t)$ | $N_e$ | state $j$ residuals $\mathbf{n}_j(t)-\tilde{\mathbf{n}}_j(t)$ |
| $\mathbf{p}$ | $N_a$ | adverb space vector: an adv., or adv. loc. |
| $\mathbf{p}_i$ | $N_a$ | adv. corresponding to exemplar motion $i$ |
| $\mathbf{p}_i^h$ | $N_a+1$ | homogeneous $\mathbf{p}_i$; $\mathbf{p}_i^h=[1\ \ \mathbf{p}_i^T]^T$ |
| $\mathbf{q}$ | $N_e$ | vector of RBF amplitudes, 1 per exemplar |
| $\mathbf{r}(\mathbf{p})$ | $N_e$ | vect. of exemplar RBF intensities, $r_i(\mathbf{p})$ |
| $\mathcal{T}$ | 1 | set of consecutive time steps $\subseteq\Re$ |
| $\beta_i$ | 1 | decay constant for RBF at adv. $\mathbf{p}_i$ |
| $m_{ij}(t)$ | 1 | $j$th component of (state in) $\mathbf{m}_i(t)$ |
| $\rho_i(\mathbf{p})$ | 1 | distance from exemplar adv. $\mathbf{p}_i$ to $\mathbf{p}\in\mathcal{A}$ |
| $r_i(\mathbf{p})$ | 1 | mag. at $\mathbf{p}$ of RBF located at adverb $\mathbf{p}_i$ |
| $r_{ij}$ | 1 | $r_i(\mathbf{p}_j)$, intensity at $j$th adv. of $i$th RBF |

for each of the $N_e$ exemplar motion-trajectory-plus-adverb pairs $(\mathbf{m}_i,\mathbf{p}_i)$. Generally, the mapping is unknown for trajectories other than the exemplars. The VaV algorithm, in effect, computes $\Phi$ to find a trajectory $\mathbf{m}$ for a given parameterization $\mathbf{p}$.

In [23], several example motions were created for articulated characters. The mapping of these motions into a multidimensional adverb space defined extremal points along axes of the space. A particular adverb extremum characterized the appearance of the associated motion. To create motions that exhibited combinations of the characteristics, a location in the adverb space was selected and mapped back into the motion space. In the work described here, the adverbs are the 3-D Cartesian world coordinates of the object to be grasped (the wrench). Exemplar reach-and-grasps were acquired near workspace extrema for the robot's right arm. To perform the operation at other locations in the workspace, the VaV algorithm was used to interpolate the exemplar motions.

The algorithm projects the motion exemplars at each time $t$ onto an $N_a+1$-dimensional, linear subspace of the motion state space $\mathcal{S}$. That subspace is the range of a matrix $A(t)$ that is the least-mean-square (LMS) approximation of $\Phi[\cdot](t)$. Since $\Phi[\cdot](t)$ is nonlinear, and because the dimension of the adverb space is usually much smaller than that of the motion state space, the projection through $A(t)$ is inaccurate. In fact, the exemplar adverbs are not mapped by $A(t)$ to their corresponding motion trajectories. To compensate, a radial basis function (RBF) interpolation operator is defined that restores the exemplar's components lost in the projection. Given a new adverb (in this case, a new grasp location) $\mathbf{p}$ the corresponding motion $\mathbf{m}(t)$ is found by computing $\mathbf{m}(t)=A(t)\mathbf{p}$, then adding to that the RBF interpolation of the exemplars that is indicated by $\mathbf{p}$. This approach permits a limited extrapolation of the data, since the subspace projection can construct new trajectories that extend parametrically beyond the exemplars.

*1) Linear Approximation:* The LMS subspace is found by deriving an approximation of $\Phi(t)$ directly for each time step (sample) of the exemplars. Since the $i$th motion exemplar $\mathbf{m}_i$ is functionally related to the $i$th adverb $\mathbf{p}_i$ (for $i=1,\ldots,N_e$), each state $m_{ij}(t)$ (for $j=1,\ldots,N_s$) at each instant $t$ is likewise related to $\mathbf{p}_i$. Assume the relationship is first order (affine). Then at time $t$, state $j$ of exemplar $i$ is related to the $i$th adverb through a vector of coefficients $\mathbf{a}_j(t)\in\Re\times\mathcal{A}$, as follows:

$$m_{ij}(t)=\left[\mathbf{p}_i^h\right]^T\mathbf{a}_j(t) \tag{7}$$

where $\mathbf{p}_i^h=[1\,\mathbf{p}_i^T]^T$ is a homogeneous representation of the adverb space pre-image of $\mathbf{m}_i$. To compute all the states of all exemplars at time $t$, use

$$M(t)=A(t)P^h. \tag{8}$$

$M(t)$ is the $N_s\times N_e$ matrix of exemplar states at time $t$. The $i$th column of $M(t)$ is $\mathbf{m}_i(t)$, the vector of $N_s$ state values of exemplar $i$ at time $t$. The $j$th row of $M(t)$ is $\mathbf{n}_j^T(t)$, the transpose of the vector that contains the $j$th state of all $N_e$ exemplars at time $t$. $P^h$ is the $(N_a+1)\times N_e$ constant matrix whose $i$th column is $\mathbf{p}_i^h$, the homogeneous representation of the $i$th adverb vector. $A(t)$ is the $N_s\times(N_a+1)$ matrix, whose $j$th row is $\mathbf{a}_j^T(t)$, the transpose of the vector of coefficients, which are unknown. There is one $\mathbf{a}_j(t)$ for each state variable at each time step in a motion trajectory.

If $\Phi(t)[\cdot]$ were linear, then

$$M(t)=\Phi(t)\,[P]=A(t)P^h. \tag{9}$$

Probably $\Phi$ is not linear, so (9) does not hold. Instead the $A(t)$ is found that minimizes the mean-squared error

$$||M(t)-\tilde{M}(t)||^2,\quad\text{where}\ \ \tilde{M}(t)=A(t)P^h. \tag{10}$$

The LMS solution is

$$A(t)=M(t)(P^h)^T\left[P^h(P^h)^T\right]^{-1}. \tag{11}$$

Then

$$\tilde{M}(t)=A(t)P^h=M(t)(P^h)^T\left[P^h(P^h)^T\right]^{-1}P^h. \tag{12}$$

Matrix $A(t)$ maps $\mathbf{p}_i^h$, which contains the adverb associated with exemplar $i$, into $\tilde{\mathbf{m}}_i(t)$, which is the orthogonal projection

of $\mathbf{m}_i(t)$ onto the range of the LMS approximation of $\Phi$. For any adverb, $\mathbf{p} \in \mathcal{A}$, the approximate motion-state vector at time $t$, is therefore

$$\tilde{\mathbf{m}}(t) = A(t)\,\mathbf{p}^h. \tag{13}$$

*2) Interpolation:* Trajectory $\tilde{\mathbf{m}}$, as computed with (13) over all $t$, is a linear subspace approximation of the true trajectory $\mathbf{m}$. Usually, $N_a + 1$, the dimension of the subspace, is considerably smaller than $N_s + 1$, which means that the approximation is, likely, not very accurate. In fact, it is usually the case that

$$\mathbf{m}_i(t) \neq \tilde{\mathbf{m}}_i(t) = A(t)\mathbf{p}_i^h \tag{14}$$

for $i = 1, \ldots, N_e$, and for all $t$; the mapping is incorrect even from the exemplar adverbs to the exemplar trajectories. Let $\bar{\mathbf{m}}_i(t)$ represent the $i$th residual

$$\bar{\mathbf{m}}_i(t) = \mathbf{m}_i(t) - \tilde{\mathbf{m}}_i(t) \tag{15}$$

for $i = 1, \ldots, N_e$.

RBFs can be used to define at each timestep a function $f(t)[\cdot]$ that augments the LMS transform $A(t)$, so that the resultant transform holds for all the exemplars. That is

$$\mathbf{m}_i(t) = A(t)\mathbf{p}_i^h + f(t)[\mathbf{p}_i] \tag{16}$$

for $i = 1, \ldots, N_e$. RBFs so defined act as interpolation functions, so that an arbitrary adverb $\mathbf{p}$ (not necessarily one of the exemplars) maps to a combination $\mathbf{m}$ of the exemplar motions through the expression

$$\mathbf{m}(t) = A(t)\mathbf{p}^h + f(t)[\mathbf{p}]. \tag{17}$$

$f(t)[\cdot]$ is determined as follows. (13), (15), and (16) imply

$$f(t)[\mathbf{p}_i] = \bar{\mathbf{m}}_i(t). \tag{18}$$

$f(t)[\cdot]$ maps adverb $\mathbf{p}_i$ to the residuals of the states of the $i$th exemplar trajectory. If we consider all the exemplars at once, this becomes

$$f(t)[P] = \bar{M}(t) \tag{19}$$

To determine $f(t)[\cdot]$, we consider the rows of $\bar{M}(t)$

$$\bar{M}(t) = \begin{bmatrix} \bar{\mathbf{n}}_1^T(t) \\ \vdots \\ \bar{\mathbf{n}}_{N_s}^T(t) \end{bmatrix}. \tag{20}$$

Vector $\bar{\mathbf{n}}_j(t)$, the transpose of the $j$th row, contains the residuals for state $j$ of all $N_e$ exemplars at time $t$. Note that $\mathbf{n}_j(t) \in \mathcal{E}$.

Let $r_i$ be an exponential RBF defined at the $i$th adverb location, $\mathbf{p}_i$. Its intensity at any point $\mathbf{p} \in \mathcal{A}$ is

$$r_i(\mathbf{p}) = e^{-\beta_i \rho_i^2(\mathbf{p})} \tag{21}$$

where

$$\rho_i(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_i\| \tag{22}$$

the distance from $\mathbf{p}$ to $\mathbf{p}_i$. Parameter $\beta_i$ determines the falloff in intensity of the $i$th RBF as the distance from it increases. For the reach-and-grasp experiments, these were computed as

$$\beta_i(\mathbf{p}) = \frac{2\ln 10}{\min\limits_{j \neq i} \left\{ \|\mathbf{p}_j - \mathbf{p}_i\|^2 \right\}} \tag{23}$$

so that at $\mathbf{p}_j$, the exemplar adverb closest to $\mathbf{p}_i$, the intensity was $r_i(\mathbf{p}_j) = 0.01$.

Define $R$ as the $N_e \times N_e$ matrix of RBF intensities at the locations of the $N_e$ adverb vectors

$$R = [r_{ik}], \quad \text{where} \quad r_{ik} = r_i(\mathbf{p}_k) \tag{24}$$

for $i, k \in \{1, 2, \ldots, N_e\}$. The $i$th row of $R$ contains the values of the $i$th RBF measured at each adverb location. The $k$th column contains the values of all the RBFs measured at the location of adverb $k$.

Vector $\bar{\mathbf{n}}_j(t)$ can be represented in terms of $R$ by

$$\bar{\mathbf{n}}_j(t) = R^T \mathbf{q}_j(t) \tag{25}$$

where $\mathbf{q}_j(t) = [q_{j,1}(t) \;\; \ldots \;\; q_{j,N_e}(t)]^T$ weights the values of the RBFs at each adverb location, so that the residual of state $j$ is matched. For all $N_s$ states, this can be written as

$$\bar{M}^T(t) = R^T Q(t) \tag{26}$$

where $Q(t)$ is an $N_e \times N_s$ matrix. Since $\bar{M}(t)$ and $R$ are known, $Q(t)$ can be found by inverting $R$

$$Q^T(t) = \bar{M}(t) R^{-1}. \tag{27}$$

(Note that $R$ is not a function of time, since it depends on the adverbs which are constant.) If $R$ is not invertible, an appropriate pseudoinverse can be employed. With this, the exemplar adverbs will map to their corresponding trajectories through

$$M(t) = A(t)P + Q^T(t)R \tag{28}$$

or for exemplar $i$

$$\mathbf{m}_i(t) = A(t)\mathbf{p}_i^h + Q^T(t)\mathbf{r}(\mathbf{p}_i) \tag{29}$$

where

$$\mathbf{r}(\mathbf{p}_i) = [\, r_1(\mathbf{p}_i) \quad \cdots \quad r_{N_e}(\mathbf{p}_i) \,]^T \tag{30}$$

is the contribution at each adverb of the $i$th RBF. $Q$ is found not for the purpose of recreating the exemplars, but for interpolating between them. If an arbitrary adverb $\mathbf{p}$ is used in (29), $Q^T(t)\mathbf{r}(\mathbf{p})$ interpolates $\bar{M}(t)$ to produce a "difference" estimate, $\bar{\mathbf{m}}(t)$ for what would (presumably) be the associated true motion-state vector, $\mathbf{m}(t)$.

Therefore, $f(t)[\cdot] = Q^T(t)\mathbf{r}(\cdot)$, and given a grasp location $\mathbf{p}$, an estimated motion trajectory is computed by

$$\mathbf{m}(t) = A(t)\mathbf{p}^h + Q^T(t)\mathbf{r}(\mathbf{p}) \tag{31}$$

for each time step $t$. Fig. 6 shows 11 trajectories generated by the VaV algorithm. Two of them place the POR at exemplar positions, at opposite corners of the box delineated by the teleoperated trials. Seven of the trajectories lie within the exemplar box

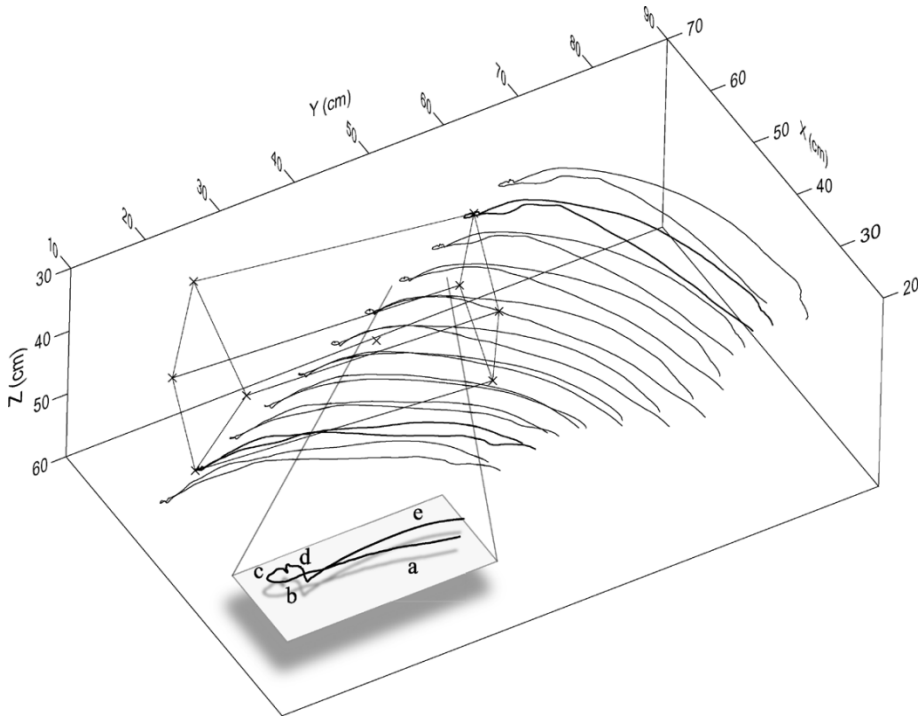



Fig. 6. Eleven trajectories generated by the VaV algorithm. The inset is a detail that shows (a) the approach, (b) grasp, (c) hold, (d) release, and (e) retract episodes of one of the 11 synthesized trajectories.

and, therefore, result from the interpolation of all nine exemplars. Two of the trajectories lie outside the box. These demonstrate that extrapolation beyond the exemplar area can result in good trajectories.

### *E. Other Methods*

In addition to the method described above, two other methods were used to perform the reach-and-grasp autonomously. The first method, called AutoGrasp and described in [8], used only one exemplar trajectory derived from six repetitions of a similar reach-and-grasp operation, but to only one central workspace location. Given another grasp location, this original trajectory was adjusted toward the grasp location at each time step. In simulation, this method achieved a high placement accuracy, but for locations far from the original, the hand approached the wrench from the wrong direction for a grasp to be successful. When implemented on Robonaut, the AutoGrasp method interacted poorly with the vision system. While it was possible to run one or two trials without a problem, the continual update of the wrench location would gradually introduce an error into the trajectory adjustment.

The second method, LinearGrasp, linearly interpolated the learned trajectories directly. First, the distance in each Cartesian dimension from each of the nine example locations to the current wrench location was calculated. A Gaussian curve centered at each example provided weights for each dimension based on these distances. The weights were normalized, and each example motion was multiplied by its corresponding weight. When these weighted motions were superpositioned, the result was a motion that would, ideally, perform the reach-and-grasp at the new location. Both in simulation and on Robonaut, however, the method was found to be imprecise. Sometimes it would grasp at the correct location; other times it would miss. A full description and analysis of both these programs and their results is available in [25].

## IV. Experimental Methods and Procedures

The VaV procedure was tested in simulation and on Robonaut. Simulation tests were run on a randomized list of 269 reachable targets in a 3-D grid that covered the entire workspace and extended somewhat beyond the edges defined by the original box. The test on Robonaut was performed by affixing a wrench to a jig, and placing it arbitrarily at reachable points in the workspace. Some attempt was made to cover the entire workspace, but since the goal was to prove that Robonaut could reach randomly generated targets, a systematic selection was not used. Robonaut's vision system was employed to locate the wrench in the workspace. The following infinite-impulse response (IIR) low-pass filter was applied to the wrench pose $\mathbf{w}(t)$ (location and rotation) to smooth perturbations due to noise

$$\mathbf{w}_{\mathrm{LPF}}(t) = 0.1\mathbf{w}(t) + 0.9\mathbf{w}_{\mathrm{LPF}}(t-1). \tag{32}$$

The major difficulty encountered in performing these experiments was Robonaut's eye–hand coordination. The actual location of the hand can vary, as the encoders that measure the joint angles are turned on and off. At the time of the tests, the solution to the problem was a manual calibration with three steps. First, the arm was reset (by eye) to its zero position, and the encoders were reset so that they would report zero at that location. Second, the reported POR on Robonaut's hand was changed from the standard location for teleoperation, which is on the back of the hand. That location on the robot corresponds to the

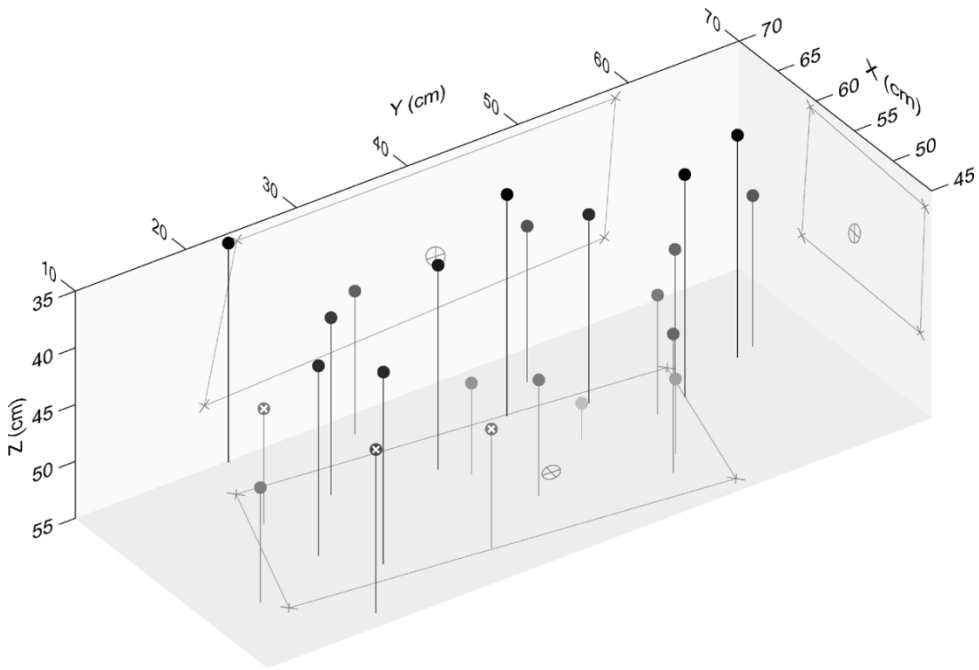



Fig. 7. Plot of the 23 test locations. The wrench was placed at each of the locations marked with a disk. Projections onto each coordinate plane of the locations of the wrench during the teleoperated trials are indicated by $\times$ and $\oplus$ marks. The three disks, each marked with a white $\times$, indicate locations at which the grasp failed. At the 20 other locations, the robot grasped the wrench successfully.

location of the position sensor on the teleoperator's data glove. The POR was changed to the standard location for autonomous operation, which is in the middle of the palm. Third, a wrench was placed in the workspace and was reached for manually by moving the individual joints to the correct location, then the reference location for the hand was changed again by a few centimeters. This was made as a final adjustment between the location reported by vision and that reported by the arm kinematics. After this adjustment, when the hand was grasping the wrench, the location of the hand as reported by Robonaut matched the location of the wrench as reported by the vision system.

During the experiment, the wrench was put in 23 different locations (see Fig. 7). The run-time part of the VaV program, which implements (31), was run for each of the locations. The only input that the program had were the results of the offline analysis and the location of the wrench reported by the visual system, which was updated in real time.

## V. Results

The simulator was of limited value in testing the procedure, since it had no direct method for judging the outcome of a grasp attempt. Nevertheless, the simulator was used, since it enabled a more complete analysis of the workspace than with Robonaut, due to time-sharing constraints. To ameliorate the deficiencies of simulation, numeric criteria were created from the trials run physically on Robonaut (both the original teleoperator examples and the experimental results). The trials were sorted based on physical evidence of a good or bad grasp, and then analyzed within the two categories. Three criteria for a good grasp in the simulated data were created. The first criterion was the most obvious. If the grasp occurred too far away from the wrench to have enveloped it, the grasp could not have been successful. Any grasp that was more than 2.6 cm from the wrench location was labeled as bad. The second and third criteria concern the approach angles. If the arm motion caused the hand to approach the wrench at the wrong angle, the hand could not grasp it because the fingers, or even the hand itself, would have had to physically pass through the wrench. To judge approach angles, a vector was created by finding the direction of motion produced in the final stages of the Reach behavior. When converted to spherical coordinates, this direction provided two approach angles: $\theta$, measured in the $x-y$ plane; and $\phi$, the angle with respect to the $z$ axis. These angles provided a way to judge if the trial in simulation correctly approached the wrench. The data recorded from Robonaut determined that, for a successful approach, the angles had to be between $-1.7^\circ$ and $-25.8^\circ$ in $\phi$, and between $134.7^\circ$ and $76.8^\circ$ in $\theta$.

Finally, some of the physical grasps that were incorrect were not the fault of the superposition method, but of the calibration of the vision system (which was beyond the authors' control). Also, occasional inaccuracies in depth perception within various regions of the workspace resulted in errors in reported wrench location. When that happened, the hand grasped in front of or behind the wrench. Nevertheless, it did grasp at the location indicated. Since these errors were not in the superposition method itself, the corresponding grasps were defined as "marginal," and were classified as good grasps for the purpose of creating the simulator criteria and calculating results. This issue will be addressed further in future work. In particular, it should be straightforward for the robot to learn that it has been unsuccessful so that it can retry the task. Fig. 8 shows the index-finger joint trajectory and the force response characteristic of a successful and an unsuccessful grasp. Although the joint trajectories are similar in both cases, there is a clearly discernable difference in the force signatures. If such a difference is consistent

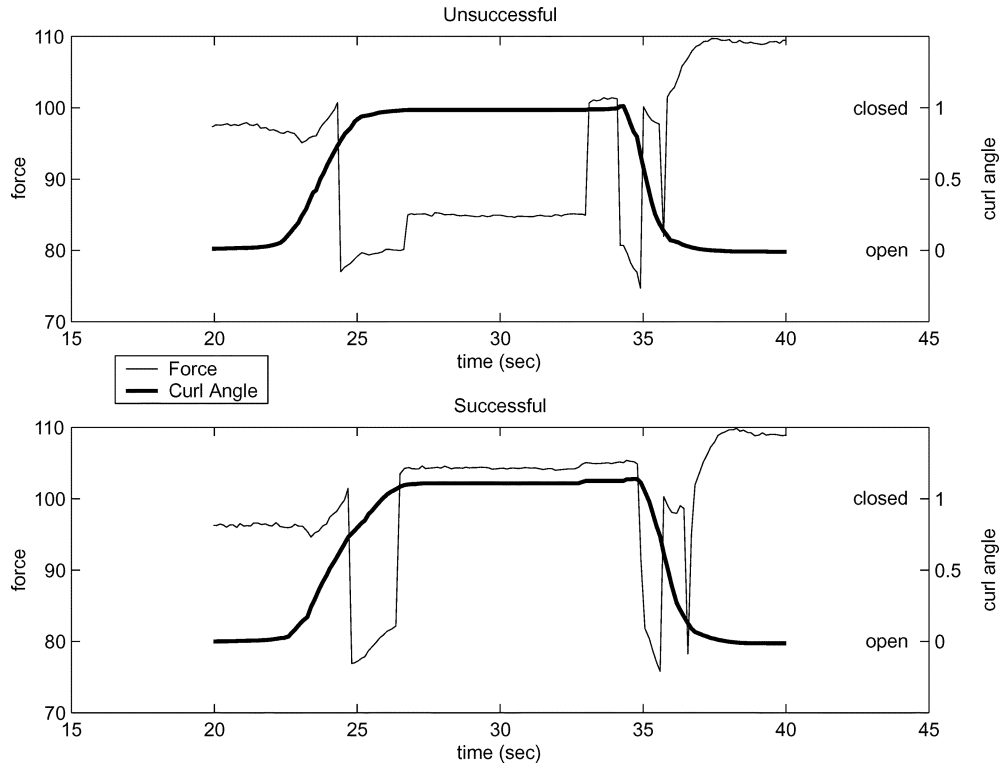



Fig. 8. Plot of the index-finger joint trajectories (their sum) superimposed on the finger force response (also summed) for a successful grasp (bottom) and an unsuccessful grasp (top).

TABLE III
RESULTS FROM SIMULATION

| Method | Good Angle | Good Dist. | Good Overall | % Good |
|---|---|---|---|---|
| AutoGrasp | 192 | 269 | 192 | 71.38 |
| LinearGrasp | 267 | 39 | 39 | 14.50 |
| VerbsAdverbs | 267 | 269 | 267 | 99.26 |

TABLE IV
RESULTS FROM EXPERIMENTS ON ROBONAUT

| Method | Good Grasps | Marginal Grasps | % Good or Marginal |
|---|---|---|---|
| AutoGrasp | 3 | 7 | 43.48 |
| LinearGrasp | 8 | 10 | 78.26 |
| VerbsAdverbs | 10 | 10 | 86.96 |

(we found it to be so in the 20 successful and 3 unsuccessful experiments), it can be detected and so used.

Tables III and IV report the results of the three methods in simulation and on Robonaut. The VaV method outperformed the other two programs. It had better than 99% accuracy in the simulator trials, which were designed to cover the entire workspace. While not performing perfectly in the physical trials, it outperformed the other methods used.

## VI. CONCLUSION AND FUTURE WORK

The work reported in this paper has supported the hypothesis that a task can be learned by an articulated, dexterous robot through teleoperation, and that the task can be performed later autonomously with reasonable robustness. It was demonstrated that 45 repetitions of a reach-and-grasp task, 5 at each of 9 locations, was sufficient for autonomous performance at random locations throughout the workspace with a success rate of 87%. After teleoperation, sensory-motor data was segmented into episodes, and averaged to find nine exemplar state-space trajectories. In the framework of the larger project that uses the results (cf. Section I), the exemplars are nine instances of a sequence of five basic behaviors that were guided by nine different sensory cues. The trajectories were interpolated successfully using the VaV algorithm. This, in turn, supports the larger project's hypothesis that tasks learned as sequences of behaviors in the form of exemplars of clusters of sensory-motor state-space trajectories can be superpositioned to enable the robot to perform the task under more widely varying conditions than those during which the task was learned. That is, the run-time superpositioning of previously learned behaviors enables robust task performance.

The VaV approach interpolates both reaching and grasping. However, the grasping in our experiments was restricted to a simple distal closure around a vertically oriented object (a wrench). The position of the hand relative to the arm changed at each of the exemplar object locations. The grasp varies slightly over the locations. Therefore, the grasp (as defined by the finger position trajectories) was interpolated, as well. To further understand the capabilities and limitations of VaV, we intend to test the interpolation of grasping in the near future. Our first test will be to learn the trajectories for various distal grasps (e.g., from above and from below) of a quasi-cylindrical object in vertical and horizontal orientations. Then we will test the distal grasping of a similar object in arbitrary orientation within the same plane as the examples.

Our work in this paper used the vision system to obtain the target location, it used proprioceptive data, i.e., sensory

data from the joint sensors in its trajectory calculations, and it used haptic, force, and torque data to detect events during autonomous operation. The vision system was not used for visual servoing. That is, it was not used in closed loop with the arm and hand controllers to determine and change the pose of the hand relative to the target. A hybrid approach that includes vision in closed loop may prove to be more capable than VaV alone. Platt has shown recently that Grupen's closed-loop controller approach works very well for certain complicated manipulation tasks [26]. Therefore, we are currently researching the combination of our trajectory extraction with Grupen's grasp controllers. In effect, VaV is used to control unconstrained motion, sequencing, and pre-grasp pose control, while Grupen's controllers handle dextrous manipulation. In one current experiment, the VaV algorithm is being used to extract a description of a multistep task performed by Robonaut through teleoperation. The task includes various reaching, grasping, and manipulation steps. The VaV trajectories are used to guide Robonaut's hand toward intermediate goal positions with a pose evolution that mimics that of the teleoperator and sets up the hand for a grasp. When the hand comes within range of the target, an appropriate closed-loop controller, predesigned for the manipulation task, is selected from among a set of such controllers. The controller guides the manipulation task, and then releases control back to the VaV trajectory generator for the next unconstrained motion.

A set of closed-loop controllers can be used as sensors during teleoperated and VaV-controlled tasks. Fagg has demonstrated recently that a set of such controllers operating independently and in parallel can reliably predict the intent (within a task context) of an unconstrained motion [27]. As observers, the controllers indicate which behavior (from among a set of possibilities) is the most likely to be required next in the task. Thus, the controllers themselves guide their selection during the unconstrained phases. This information, along with the VaV trajectories, becomes a recipe for autonomous task execution in the face of environmental contingencies.

The VaV algorithm may also enable the automatic design of such controllers. In Grupen's approach, closed-loop controllers follow trajectories in a state space that lead to attractors. Our approach extracts and recombines trajectories in a sensory-motor state space. Therefore, it appears possible that our approach could be used to discern regions of attraction in the state space, which could then be used to define closed-loop controllers automatically.

The next step in the project is to extend the types of behaviors used and demonstrate that behaviors learned at different times for different tasks can be composed at runtime to solve new problems. This method bears some similarity to classical gain scheduling; however, the dynamics of the underlying Robonaut controllers did not dominate the behaviors we have explored so far. We plan to extend the range of behaviors to include highly dynamic ones, and determine how well this procedure extends. Given a robust set of learned behaviors, we believe that their composition will allow Robonaut to become robust at problem solving.

## ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their comments, which greatly strengthened the paper. The authors also thank R. Savely and the Dexterous Robotics Laboratory of the Automation, Robotics, and Simulation Division at NASA Johnson Space Center for comments, advice, and help throughout the project. Thanks also to D. Gage of the DARPA Information Processing Technology office for supporting this work and to M. Swinson, now of the Mathematical and Information Sciences Directorate of the Army Research Office, for initializing this collaboration.

## REFERENCES

[1] R. A. Peters, II, K. Kawamura, D. M. Wilkes, D. M. Gaines, and R. O. Ambrose. (2000) Robot learning and problem solving through teleoperation with application to human – robot teaming: A white paper. [Online]. Available: http://www.vuse.vanderbilt.edu/~rap2/papers/RobotLearningWhitePaper20001129.pdf

[2] R. Pfeifer and C. Scheier, "Sensory-motor coordination: The metaphor and beyond," *Robot. Auton. Syst.*, vol. 20, no. 2–4, pp. 157–178, Jun. 1997.

[3] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Feb. 1986.

[4] M. M. Matarić, "Integration of representation into goal-driven behavior-based robots," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 304–312, Jun. 1992.

[5] P. Maes and R. A. Brooks, "Learning to coordinate behaviors," in *Proc. 8th Nat. Conf. Artif. Intell.*, vol. AAAI-90, Jul. 1990, pp. 796–802.

[6] M. Huber and R. A. Grupen, "A hybrid discrete event dynamic systems approach to robot control," Univ. Mass., Dept. Comput. Sci., Amherst, MA, Tech. Rep. 96-43, 1996.

[7] M. Huber, "A hybrid architecture for adaptive robot control," Ph.D. dissertation, Univ. Mass., Amherst, MA, 2000.

[8] R. A. Peters, II, C. L. Campbell, W. J. Bluethmann, and E. Huber, "Robonaut task learning through teleoperation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., Oct. 2003, pp. 2806–2811.

[9] P. R. Cohen, "Learning concepts by interaction," Univ. Mass., Dept. Comput. Sci., Amherst, MA, Tech. Rep. 00-52, 2000.

[10] P. R. Cohen and N. Adams, "An algorithm for segmenting categorical time series into meaningful episodes," in *Proc. 4th Symp. Intell. Data Anal.*, vol. 2189, 2001, pp. 198–207.

[11] J. A. Coelho, Jr., J. H. Piater, and R. A. Grupen. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. presented at *Proc. 1st IEEE/RAS Int. Conf. Humanoid Robots*. [Online]. Available: http://www-robotics.cs.umass.edu/Papers/Humanoids2000.pdf

[12] M. E. Cambron and R. A. Peters, II, "Determination of sensory motor coordination parameters for a robot via teleoperation," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 5, Oct. 2001, pp. 3252–3257.

[13] M. J. Matarić, "Getting humanoids to move and imitate," *IEEE Intell. Syst. Mag.*, vol. 15, no. 4, pp. 18–24, Jul. 2000.

[14] O. C. Jenkins and M. J. Matarić, "Deriving action and behavior primitives from human motion data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2002, pp. 2551–2556.

[15] A. Ude, C. G. Atkeson, and M. J. Riley, "Planning of joint trajectories for humanoid robots using b-spline wavelets," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2001, pp. 2223–2228.

[16] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, pp. 1390–1397.

[17] C. G. Atkeson, J. G. Hale, F. Pollick, M. J. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, and M. Kawato, "Using humanoid robots to study human behavior," *IEEE Intell. Syst. Mag.*, vol. 15, no. 4, pp. 46–56, Jul. 2000.

[18] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec. 2000.

[19] R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: NASA's space humanoid," *IEEE Intell. Syst. Mag.*, vol. 15, no. 4, pp. 57–63, Jul. 2000.

[20] (2004) Robonaut Avionics. Robonaut Devel. Team, NASA Johnson Space Center, Tech. Rep.. [Online]. Available: http://robonaut.jsc.nasa.gov/Avionics.htm
[21] J. C. Krieg, "Motion tracking: Polhemus technology," *Virtual Reality Syst.*, vol. 1, no. 1, pp. 32–36, Mar. 1993.
[22] E. Huber and K. Baker, "Using a hybrid of silhouette and range templates for real-time pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2004, pp. 1652–1657.
[23] C. Rose, B. Bodenheimer, and M. F. Cohen, "Verbs and adverbs: Multidimensional motion interpolation," *IEEE Comput. Graphics Appl.*, vol. 18, no. 5, pp. 32–44, Sep. 1998.
[24] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Auton. Robots*, vol. 12, no. 1, pp. 39–54, Jan. 2002.
[25] C. L. Campbell, "Learning through teleoperation on Robonaut," Master's thesis, Vanderbilt Univ., Nashville, TN, 2003.
[26] R. Platt, Jr., A. H. Fagg, and R. A. Grupen, "Manipulation gaits: Sequences of grasp control tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Apr. 2004, pp. 801–806.
[27] A. H. Fagg, M. T. Rosenstein, R. Platt, Jr., and R. A. Grupen. Extracting user intent in mixed initiative teleoperator control. presented at *Proc. 1st AIAA Intell. Syst. Tech. Conf.*. [Online]. Available: http://www-all.cs.umass.edu/~fagg/papers/2004/fagg_etal_aiaa04.pdf

**Christina Louise Campbell** received the B.S. and M.S. degrees in electrical engineering from Vanderbilt University, Nashville, TN, where she is currently working toward the Ph.D. degree in electrical engineering and computer science.

Her interests are in intelligent robotics and machine learning.

Ms. Campbell is a NASA Graduate Student Research Fellow.

**Richard Alan Peters, II** (S'86–M'87) received the A.B. degree in mathematics from Oberlin College, Oberlin, OH, and the M.S. and Ph.D. degrees in electrical engineering from the University of Arizona, Tucson, AZ.

He is currently an Associate Professor of Electrical Engineering with Vanderbilt University, Nashville, TN. His current research involves sensory-guided robotics, computer vision, and machine learning.

Dr. Peters was a Fellow of the American Electronics Association while at the University of Arizona.

**Robert E. Bodenheimer** (S'85–M'86) received the M.S. degree from the University of Tennessee, Knoxville, and the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena.

He is currently an Assistant Professor of Computer Science with Vanderbilt University, Nashville, TN. Previously, he was a Visiting Researcher at Microsoft Research, Redmond, WA, and a Postdoctoral Fellow at the Georgia Institute of Technology, Atlanta. His research focuses on computer graphics and computer animation, especially human figure animation.

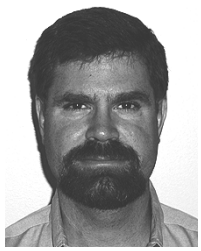
Dr. Bodenheimer received the NSF Faculty Early CAREER Development Award.

**William J. Bluethmann** received the Ph.D. degree in mechanical engineering from the University of Kansas, Lawrence.

Currently, he is the Lead Software Engineer on the Robonaut project in the Robotic Systems Technology Branch of the Automation, Robotics, and Simulation Division, NASA Johnson Space Center, Houston, TX. His research interests include manipulator control, force control, distributed autonomy, and task-level automation.

**Eric Huber** received the B.S. degree in mechanical engineering from Rutgers University, New Brunswick, NJ, and the M.S. degree in robotics engineering from the Stevens Institute of Technology, Hoboken, NJ.

Currently, he is a Research Engineer with Metrica Corporation, working at the Robotic Systems Technology Branch, NASA Johnson Space Center, Houston, TX, where he is the architect of Robonaut's stereo vision system. His research interests include the development of real-time vision systems for recognizing, locating, and tracking real-world objects in complex domains.



**Robert O. Ambrose** received the Ph.D. degree in mechanical engineering from the University of Texas at Austin.

Currently, he is the Branch Chief for NASA Johnson Space Center's Robotic Systems Technology Branch, Houston, TX, where he designed the Robonauts, NASA's space-capable humanoid robots. His research interests include robot design, space environmental modeling, actuator development, biomechanics, interactive design software, and nonlinear optimization.