Accepted to:
ASME IMECE 2000 Conference
Haptic Interfaces for Virtual Environments
and Teleoperator Systems Symposium

Calibration and Mapping of a Human Hand for Dexterous Telemanipulation

Weston B. Griffin, Ryan P. Findley, Michael L. Turner and Mark R. Cutkosky

Dexterous Manipulation Laboratory
Stanford University
Bldg. 560 Panama Mall
Stanford, CA 94305-2232
wgriffin@cdr.stanford.edu, http://www-cdr.stanford.edu/touch

ABSTRACT

This paper presents a calibration scheme and kinematic mapping to support dexterous telemanipulation. The calibration scheme is intended for use with an instrumented glove and permits an accurate determination of the intended motions of a virtual object grasped between a human operator's thumb and index finger. The motions of the virtual object are then mapped to analogous motions of a scaled virtual object held in a two-fingered robot hand. A non-linear mapping scheme allows better utilization of the human and robot hand workspaces.

1.0 INTRODUCTION

The work described in this paper is part of an effort to allow a general-purpose instrumented glove to be used as an input device for manipulating objects held in a dexterous telemanipulator. In this approach, the human operator grasps and manipulates virtual objects (e.g., between the thumb and index finger) using the glove. The resulting object motions are transmitted to a robot hand that performs the manipulations on real objects. A force feedback device, described in [11], can relay grasp forces back to the operator.

This approach has the advantage that it does not require a specially designed manipulandum or hand master and can be adapted for used with most robot hands. However, it raises a couple of technical challenges not encountered when using a manipulandum. First, a kinematic calibration of each user's hand must be obtained. The calibration should be specialized to obtain an accurate measure of the intended motions of objects manipulated between the fingers. Second, the intended motions of the virtual object in the operator's fingers must be mapped to analogous motions of real objects held in the robot hand. The robot may have different kinematics and a different workspace than the human hand; therefore a mapping from the human workspace to the robot workspace must account for these differences. In the present case, the robot hand is planar, and a particular issue is to account for the non-planar motions of a human hand.

In the following sections we first review related work and then present a kinematic model of the human hand that is specialized for thumb/index finger manipulations. The model is designed for use with the instrumented glove. Next, we present a calibration routine for the instrumented glove that quickly and accurately adjusts the model to fit a particular user. We then present a mapping from the motions of an unconstrained human hand to a non-anthropomorphic planar robot hand. The mapping is designed so that when a human makes a natural motion such as rolling, grasping or releasing an object, the robot makes an analogous motion. We present the approach using a particular two-fingered planar robotic hand and discuss extrapolations to the general case.

2.0 PREVIOUS WORK

The human hand is a remarkably complex mechanism, and researchers have made various approximations when modeling it for different purposes. In our case, we need a kinematic hand model that allows sensor readings to be mapped to coordinate frames and joint angles; modelling of the tendons or external appearance is secondary. Kuch and Huang [8] model the hand with 26 DOF, though they do not present accuracy values. Rohling and Hollerbach [9] show that the index finger can be modeled as a kinematic linkage with simple rotary joints with little loss of accuracy. Kramer [7] presents a kinematic linkage model of the thumb, though no accuracy results are presented.

Calibrating a model to a particular human hand is necessary for accurate telemanipulation. Fischer et al. [4] use a vision based system to track fiducial points on an individual hand, while measuring the joint angles with a data glove. By using a neural network, they achieve tip position errors less than 1.8 mm.

The calibration technique described in this paper applies closed-loop kinematic calibration methods to the human hand, requiring touching fingertip to thumb. This method was used to calibrate the fingers of the Utah-MIT Dextrous Robot Hand in [2], although rolling was not allowed in that case.

When mapping from human hand motions to robot motions, two principal approaches dominate the literature. In some work, the hand is placed within an exoskeleton which is kinematically similar to the robot.[14] By limiting the allowable motions of the human hand, the transformation to the robot motion is simplified. In other work, the robot hand is anthropomorphic and approximately matches the motion of the human hand.[4][10] Certain difficulties in mapping from human to robot are avoided, but requires a more complex and expensive robot hand.

3.0 HUMAN HAND CALIBRATION

3.1 Hand Model

We have developed a sufficiently accurate kinematic model of the human hand and an expedient means of customizing the model to a particular user's hand. Using the models developed by Rohling and Hollerbach [9] and Kramer [7], observations by Cooney et al.[3] and Hollister et al.[5], and our empirical examinations, we have developed a kinematic model which we feel is well suited for measuring and displaying fine fingertip manipulations. In this model, the human hand is converted to a mechanical linkage, with finger bones intersecting in Hookean pin joints. The model does not take into account effects such as soft tissue deformation or bone-on-bone sliding.

Referring to Fig. 1, the base coordinate system for convenience is located in the hand at the point where the thumb and the index metacarpal meet. The base frame x-axis points along the index metacarpal bone, the y-axis is directed outward from a flat open palm, with the z-axis defined by the right hand rule.

The index finger is defined similarly to that presented in Rohling and Hollerbach.[9] The index metacarpophalangeal joint has two orthogonal collocated degrees of freedom, abduction (I_{ABD}) and flexion (I_{MPJ}). The transforms from the I_{MPJ} to the I_{PIJ}, I_{PIJ} to I_{DIJ} and I_{DIJ} to fingertip are all defined such that axes of rotation are parallel.

Modeling the thumb is more challenging. Cooney et al. [3] show that even though the metacarpal bone has three modes of motion (flexion/extension, abduction/adduction and pronation/supination), it only has two degrees of freedom. In other words, for a given flexion (rotation about the palm) and abduction (angle between the thumb and palm), the thumb has a unique pronation (twist). Hollister et al. [5] demonstrate that this is because the thumb rotation (T_{TR}) axis and the thumb abduction (T_{ABD}) axis are non-orthogonal and non-intersecting.

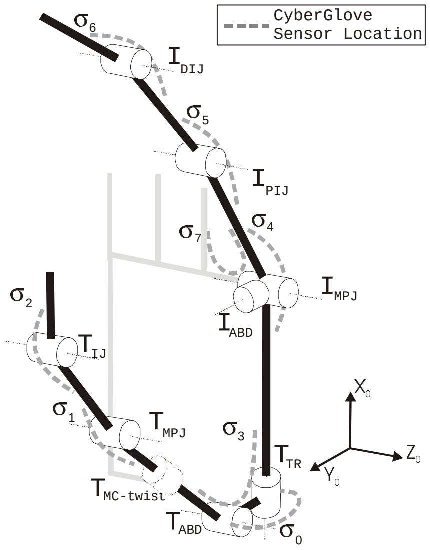

Figure 1. Kinematic Hand Model

In our model (Fig. 1), we maintain orthogonality for computation purposes, and to better match the sensor positioning of the CyberGlove®. The T_{TR} joint is located at the base of the thumb with the axis of rotation along the index metacarpal. The T_{ABD} axis is offset from and orthogonal to the T_{TR} axis. In order to account for the thumb pronation, an unsensed axis is placed along the thumb metacarpal ("$T_{MC\ twist}$"). The angle of this "joint" is a linear function of the abduction and flexion.

Some hand models [8] place a spherical joint at the metacarpophalangeal joint (T_{MPJ}). One of the axes of motion, however, exhibits little motion without being externally forced. The "principal" T_{MPJ} axis is parallel to the T_{ABD} axis for a 0° $T_{MC\ twist}$. The interphalangeal joint (T_{IJ}) is parallel to the T_{MPJ}.

3.2 Calibration Routine

We have generated a reliable and expedient means of calibrating a user's hand for use with the CyberGlove®. Proper calibration to a particular user's hand is essential for quality telemanipulation. By developing a routine which can be performed quickly and accurately each time the device is used, a new user can begin to work in a matter of minutes.

The current calibration routine is what we call a "zero-hardware solution," in which the user places his or her thumb and index finger tips together and maintains rolling contact while moving the fingers. The computer records 80 distinct data points with the CyberGlove® sensor values over 40 seconds.

Because the fingers remain in contact, we can approximate the hand as a closed kinematic chain with the fingertips in fixed contact. This is only an approximation because of several factors, including rolling motion and soft tissue deformation. This closed kinematic chain has one unsensed joint at the finger-thumb contact point, with three rotational degrees of freedom.

The error at each data point is the calculated separation between the two fingers in our nominal kinematic model. We use a least squares regression iteration to modify the nominal model to an error-minimizing model for the particular user. The major advantage of the zero hardware solution is the lack of any external calibration sensor, such as a vision system, which would increase complexity and cost.

3.3 Least Squares Fit

Rohling and Hollerbach [9] calibrated the human index finger using a least squares regression. They directly measured the position of the index tip using a vision system, which can be treated as a closed chain with the sensor measurement as the final link. We performed a similar procedure, using instead a closed finger-thumb kinematic chain.

3.3.1 Angular Offset Calibration

After storing N poses, we generate error vectors $\Delta\mathbf{d}$ from each calculated index position to the corresponding calculated thumb position, using the forward kinematics of the hand model. The hand model kinematics also provide Jacobians, $\mathbf{J}$, relating infinitesimal joint angle motion to operational space motion; these Jacobians are functions of the joint angles, ϕ. These relations are vertically concatenated from 1 to N to get

$$\begin{bmatrix} \Delta\mathbf{d}_1 \\ \dots \\ \Delta\mathbf{d}_n \\ \dots \\ \Delta\mathbf{d}_N \end{bmatrix} = \begin{bmatrix} \mathbf{J}(\phi_1 + \theta) \\ \dots \\ \mathbf{J}(\phi_n + \theta) \\ \dots \\ \mathbf{J}(\phi_N + \theta) \end{bmatrix} [\Delta\theta] \qquad (1)$$

or, more compactly, $\Delta\mathbf{d} = \mathbf{C}\Delta\theta$, where ϕ_n is the set of joint angles for pose n, θ is a vector of offsets applied to each set of ϕ_n, and $\mathbf{C}$ is the matrix of concatenated Jacobians. We can evaluate a least-squares solution, $\Delta\theta$, to this linearized system using the left pseudo-inverse of $\mathbf{C}$, based on the SVD. At each iteration step, θ is modified by $\Delta\theta$. The values of $\Delta\mathbf{d}$ and $\mathbf{C}$ are recalculated, using the new θ. The iteration continues until $\Delta\theta \to 0$. An extensive discussion on the numerical stability of this convergence is presented by Rohling and Hollerbach [9].

Because this calibration generates a scale-independent model, for this phase of the calibration development we make the assumption that the ratios of bone lengths to one another are fairly constant among users. The relative length of each bone is taken from published biomechanics data.[1]

3.3.2 Including Unknown Sensor Gains

Implementation of the angular offset calibration resulted in poor accuracy of the modeled hand. Because the CyberGlove® attaches to the soft tissue of the hand, and due to the nature of the sensors, the conversion gain from sensor value to angular quantity is variable from user to user, unlike in an exoskeleton-type hand master. In addition, some glove sensors are physically cross-coupled, i.e., sensor values may change due to the movement of more than one joint.

The solution is to expand the calibration to optimize the values of the sensor gains, using the relation

$$\phi_i = g_i\sigma_i + \theta_i \qquad (2)$$

where σ_i is the raw sensor value, and g_i is the gain. A new Jacobian $\hat{\mathbf{J}}_n$ is formed by adding columns for the new gain parameters:

$$\Delta\mathbf{d}_n = \hat{\mathbf{J}}_n \begin{bmatrix} \Delta\theta \\ \Delta\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_n & \mathrm{Diag}(\sigma_n)\cdot\mathbf{J}_n \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\mathbf{g} \end{bmatrix} \qquad (3)$$

Note that the individual $\hat{\mathbf{J}}_n$ has linearly dependent columns, however the concatenated Jacobian matrix $\hat{\mathbf{C}}$ will not, because each set of sensor readings σ_n is distinct.

In addition, cross-coupling effects can be modeled by including a cross gain parameter. For instance, the T_{TR} and T_{ABD} sensors are highly cross-coupled between the T_{TR}, T_{ABD}, and $T_{\mathrm{MC\ Twist}}$ angles of the model. The relation between these two sensors and their corresponding model joints can be expressed in matrix form:

$$\begin{bmatrix} \phi_{TR} \\ \phi_{ABD} \\ \phi_{\mathrm{MC\ Twist}} \end{bmatrix} = \begin{bmatrix} g_{TR} & g_{TR}^{ABD} \\ g_{ABD}^{TR} & g_{ABD} \\ g_{MCTw}^{TR} & g_{MCTw}^{ABD} \end{bmatrix} \begin{bmatrix} \sigma_{0(TR)} \\ \sigma_{3(ABD)} \end{bmatrix} + \begin{bmatrix} \theta_{TR} \\ \theta_{ABD} \\ \theta_{\mathrm{MC\ Twist}} \end{bmatrix} \qquad (4)$$

To include the cross-coupled nature of these two sensors, we need six parameters instead of just g_{TR} and g_{ABD}. Note that the $T_{\mathrm{MC\ Twist}}$ parameter has no corresponding σ value because the CyberGlove® does not measure this type of motion. $\hat{\mathbf{J}}_n$ is extended for the cross gain parameters similarly to (3).

$\hat{\mathbf{J}}_n$ can be further expanded to include relations between bone lengths and fingertip positions. Bone lengths, gains, and offsets can then be incorporated into a generalized parameter vector $\mathbf{p}$. The model calibration can now be optimized in virtually every relevant parameter. In summary we are now calibrating for:

- nine constant offset parameters of the model: θ_i
- eight bone lengths, L_i
- eight sensor gains, g_i
- and four cross-coupling terms, g_j^i defined in Eq. (4)

When this revised calibration was performed, the hand parameters converged quickly to trivial solutions. In one case, all the gains converge to zero, eliminating all finger motion. The other parameters converge to a configuration where the finger offset is exactly zero. Because the fingers do not move, the error is zero for all sensor readings. In a second case, all bone lengths converge to zero, similarly eliminating all motion.

One tool for preventing closed-loop trivial solutions is to fix a gain and length [2]. An alternate method to prevent this difficulty is the use of *a priori* estimates, as advocated in [13]. The calibration routine can be further modified to limit parameter deviation as discussed below.

3.3.3 Limitation of Parameter Deviation

Using a graphical display of the hand model and the default calibration information which ships with the CyberGlove®, it is possible to manually adjust the set of parameters θ and $\mathbf{g}$ to produce a visually acceptable representation of the human hand across a wide variety of users. This type of manual calibration, however, produces spatial accuracy which is not sufficient for fine telemanipulation tasks. These values are used as a first approximation of the physical values, and are used to generate a nominal "biologically feasible" parameter set from which we can limit parameter variance.

To complement our generalized parameter vector $\mathbf{p}$, we introduce two new vectors, $\mathbf{p_0}$ and ρ, which are generalized vectors of nominal values, and acceptable variance of each parameter, respectively. In this way, the nominal value and acceptable deviation of each parameter can be controlled independently. The equation

$$[\mathbf{p_0} - \mathbf{p}] = \mathbf{I}\Delta\mathbf{p} \qquad (5)$$

has the effect of driving $\mathbf{p}$ toward $\mathbf{p_0}$, if iterated and $\Delta\mathbf{p}$ is applied to $\mathbf{p}$. The rate of convergence can be modified by left multiplying both sides by any nonsingular matrix $\mathbf{V}$. We choose $\mathbf{V}$ to be a diagonal square matrix with elements

$$V_{ii} = -\sqrt{N}\frac{\partial}{\partial p_i}\left[\left(\frac{p_{0i} - p_i}{\rho_i}\right)^m\right] \qquad (6)$$

where m is an integer. This $\mathbf{V}$ matrix restricts each parameter to a virtual potential well about the nominal value when incorporated into the least squares fit, equation (7).

$$\begin{bmatrix} \Delta\mathbf{d}_1 \\ \cdots \\ \Delta\mathbf{d}_N \\ \mathbf{V}[\mathbf{p_0} - \mathbf{p}] \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{J}}(\phi_1, \mathbf{p}) \\ \cdots \\ \hat{\mathbf{J}}(\phi_N, \mathbf{p}) \\ \mathbf{V} \end{bmatrix} [\Delta\mathbf{p}] \qquad (7)$$

Equation (7) limits the variance of our parameters to pre-defined ranges, and weights the variance equally with the position error, $\Delta\mathbf{d}$. The potential well function can also be adjusted by the choice of m; at $m = 10$, when the difference between a parameter, p, and its default value, p_o, is $0.9 \cdot \rho$, the corrective Δp is equivalent to a positional error of 0.34mm, but if the difference is $1.5 \cdot \rho$, the corrective Δp is equivalent to a positional error of 57.0mm. It is important for numerical stability that each p_i not stray too much outside the variance. This is avoided by reducing the step size on the optimization iteration adaptively.

Some refinements in the hand model and calibration routine continue to be explored, but the current relative position accuracy is sufficient for our telemanipulation tasks. It is worthwhile to note that the entire calibration process is automated, requires no hardware other than the CyberGlove® and a standard personal computer, and requires less than 5 minutes of data collection and computation time. Thus a new user can be completely calibrated and ready to begin using the CyberGlove with good accuracy within just a few minutes.

3.4 Performance Metrics and Experimental Results

An initial metric of calibration quality can be obtained by examining the final calibration set of separation vectors $\Delta\mathbf{d}$ from index tip to thumb tip. The RMS calibration error, ε, is defined as the RMS of the set of N distances between the fingertips, one for each pose. The separation between fingertips, however, cannot be measured exactly because of two important effects. The first limitation is the resolution of the CyberGlove, e.g., the I_{MPJ} joint resolution is approximately ±0.85°, which corresponds to a spatial position precision no better than 1mm in some hand configurations. Secondly, when actually in contact, the fingertips deform depending on the pinch force exerted. From empirical measurements, we estimate that this incurs an RMS separation uncertainty of approximately 2mm, with peak uncertainty of approximately 3mm. Therefore, because of these two effects we expect an ideal calibration technique to produce an ε on the order of 2-4mm.

Users in a related experiment [12] were calibrated using the full calibration technique. For ten users, the average ε was 5.26mm, with a standard deviation across users of 1.40mm and worst-case value of 7.8mm. In contrast, when calibrating with angular offsets only, ε values range from 11mm up to 25mm in the worst case. The difference between the two calibration methods was readily apparent by visual inspection of the graphical hand model display. Use of the offsets-only technique resulted in visually incongruous hand configurations.

To further compare the two methods, we introduce a second performance criterion, with our goal of tele-manipulation in mind. The goal of the calibration technique is to provide accurate information about the relative positioning of the user's fingertips, in particular the separation distance between them. To this end, a test was performed to compare the calculated separation to the actual separation of the user's fingers

While the user manipulated a thin rod of known length between the index and thumb fingertips, a set of poses were recorded. Rods of length 12.5, 25.5, 40.5, 50, and 64mm were used. The 12.5mm rod was difficult to manipulate and therefore prone to error. Figure 2 demonstrates the relationship between

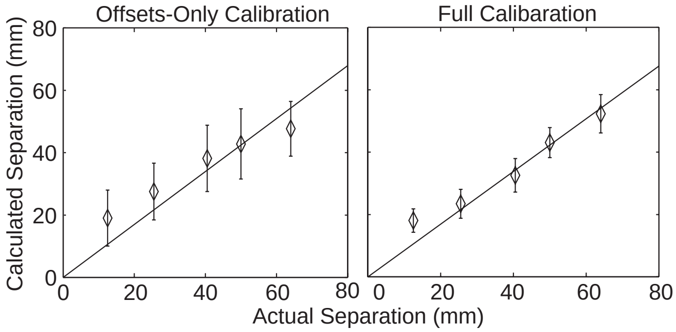

Figure 2. Fingertip Separation Linearity

actual and calculated separation for both calibration techniques. The error bars correspond to one standard deviation in a set of 300 poses across the workspace of the hand. We constrain the linear regression fit to pass through the origin.

The full calibration technique shows substantially better linearity than the offsets-only technique. Linearity is more important than exact size, because the scale of the user's hand is unknown. Good linearity directly corresponds to a better measurement of the size of a virtual object grasped by the hand, which is important in the next phase of telemanipulation; mapping the human hand to a non-anthropomorphic robot.

4.0 HUMAN TO ROBOT MOTION MAPPING

A motion-mapping scheme is used to map the motions of a user's hand, measured with the calibrated glove, to commanded positions for a dexterous robot. Difficulties arise when attempting to map the three-dimensional motion of the human hand to the two-dimensional motion of a planar robot, yet still make the mapping intuitive to the teleoperator. The differences in workspace size and shape must be compensated for.

We have developed a dextrous planar robotic hand, referred to as Dexter, that serves as a test bed for investigating motion mapping methods.[12] Dexter is a two-fingered hand, with two degrees-of-freedom per finger. The robot operates under an operational space control framework [6] in which the commanded positions are generated by the mapping method.

4.1 Point-to-Point Mapping

An initial solution to the mapping problem is a point-to-point scheme in which finger tip positions are mapped to robotic finger tip positions by a single transformation matrix. The three-dimensional fingertip positions gathered from the calibrated glove are projected on to the x-y base plane as in Fig. 1 (the plane perpendicular to the palm and containing the index finger motion with no abduction). The projected tip positions are transformed and scaled to the robot workspace through a standard frame transformation, shown below:

$$\mathbf{x}_f^R = \begin{bmatrix} G_x & 0 \\ 0 & G_y \end{bmatrix} \cdot \left(\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \mathbf{x}_f^H + \mathbf{x}_0 \right) \qquad (8)$$

In this equation, $\mathbf{x}_f^H$ represents the x-y planar projection of a fingertip position in the human hand frame, which is transformed to a position in the robotic hand fame, $\mathbf{x}_f^R$, under a rotation of θ and translation of $\mathbf{x}_0$, and scaled independently in x and y by gain matrix, $\mathbf{G}$. The transformation parameters are modified to best utilize the robotic workspace.

Under this method, the motion of the robotic finger corresponding to the index finger was relatively easy to control. This is primarily due to the similarities between the planar motion of the index finger and the robotic finger. However, control of the robotic finger corresponding to the thumb proved awkward. Moreover, as discussed further below, achievable positions for the thumb were mapped to a relatively small region of the corresponding robot finger workspace.

The point-to-point method highlighted two underlying difficulties associated with mapping from a human hand to a robot hand. First, the human thumb does not directly oppose the index finger. Thus restricting manipulation to a plane removes important information about the intended manipulation. Second, the robot has its greatest range of object manipulation at approximately the geometric center of its workspace (Fig. 3, right). Ideally, this preferred robot position should correspond to the natural pinch position of the human hand. Humans, however, tend to manipulate small objects toward the outer edge of the hand workspace (Fig. 3, left). Simply scaling the motions of the thumb and index fingertips to fit in the robots workspace results in manipulations being performed near the lower limit of the robot hand workspace, thereby limiting the manipulation range of the robot.

4.2 Virtual Object Based Mapping

To address the deficiencies of the point-to-point mapping, a method was developed based on the manipulation of a virtual object. The goal of this mapping is to allow the user to make natural motions, such as grasping, releasing, or rolling an object, and have the robot perform analogous motions. The object based mapping scheme assumes that a virtual sphere is held between the user's thumb and index finger. Important parameters of the virtual object (the size, position and orientation) are scaled independently and non-linearly to create a transformed virtual object in the robotic hand workspace. This modified virtual object is then used to compute the robotic fingertip locations (Fig. 3).

4.2.1 Mapping Implementation

The size of the virtual object is based on the three-dimensional distance between the fingertips. For example, if a person is manipulating a golf ball, the modeled finger separation should remain the same even as the thumb moves out of plane of the index finger.

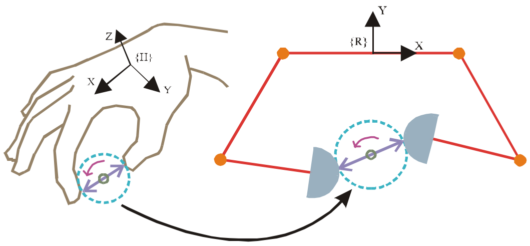

Figure 3. Object based mapping schematic

The position of the virtual object is first calculated in the hand frame by finding the midpoint between the thumb and index fingertips. The midpoint is then projected on to the x-y plane (the same plane described in Section 4.1, also see Fig. 1). The orientation of the virtual object is based on the angle of the projection of the line between the fingertips. The calculated object midpoint in the human hand frame is transformed to the robotic hand frame using a standard transform as in Eq. (8), but with a unity gain matrix. The translation maps the user's comfortable pinch location to the robot's preferred manipulation point. The rotation angle assures that the human motion of moving a virtual object into and away from the palm corresponds to an up/down motion on the robotic fingers.

4.2.2 Workspace Matching

The above procedure maps the size, position, and orientation of a virtual object in the hand frame to a virtual object in the robotic hand frame. The object parameters, in the robotic hand frame, are further modified to better match the human hand workspace to the robot workspace yet still maintain the correspondence between the natural human pinch position and the preferred robot manipulation position. As discussed below, we implemented an algorithm that modifies the object parameters using non-linear scaling functions. Figure 4 displays typical hand postures and the corresponding desired robotic hand configuration.

The object size parameter is varied non-linearly so the user can better control the robotic fingertip separation. At the comfortable manipulation region the gain on the object size is proportional to the size difference between the human hand and the robotic hand. However, when the distance between the user's fingertips increases beyond the size of a typical grasped object the gain increases such that the robot fingers can extend to the edge of the workspace. Figure 4A demonstrates the human hand posture for holding a small object and the corresponding robotic hand posture. As the user spreads his fingers the robotic fingers should spread apart as in Fig. 4B.

The object size is calculated by:

$$d^{R}_{OBJ} = K_{OBJ} \cdot d^{H}_{OBJ} \tag{9}$$

where d^{H}_{OBJ} is the calculated object size in the hand frame (three dimensional distance between the fingertips) and d^{R}_{OBJ} is the virtual object size in the robot hand frame used to calculate the robot fingertip positions. The gain, K_{OBJ}, is a piecewise linear function of the virtual object size in the hand frame, illustrated in Fig. 5a. The markers A and B reflect the gain used in the postures A and B in Fig. 4.

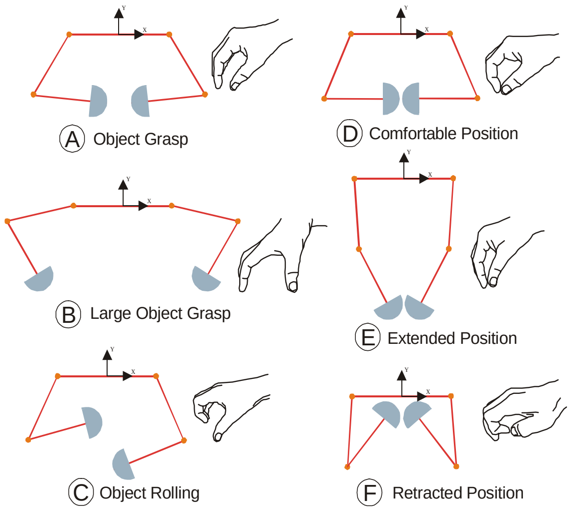

Figure 4. Typical robotic and human hand positions under object based mapping

The object midpoint parameter is also varied in a non-linear manner. Around the comfortable pinch position, we wish to map motions to the preferred manipulation region for the robotic hand, as in Fig. 4D. When the user extends his fingers away from or towards the palm, the robot should approach its own workspace limits (Figs. 4E, 4F). From the frame transformation, the rotation matrix aligns this motion with the vertical (y-axis) in the robotic hand frame. Thus the vertical position of the midpoint in the robotic hand frame is modified as follows:

$$y^{R}_{MP} = K_{Y} \cdot \hat{y}^{R}_{MP} \tag{10}$$

where $\hat{y}^{R}_{MP}$ is the unit gain midpoint y-value (height) in the robotic hand frame and y^{R}_{MP} is the modified midpoint y-value of the object used to compute the robot fingertip positions. The gain, K_{Y}, is a piecewise linear function of $\hat{y}^{R}_{MP}$. The position gain function is illustrated in Fig. 5b. The gains necessary for reaching postures D, E and F, shown in Fig. 4, are indicated with corresponding markers in the figure.

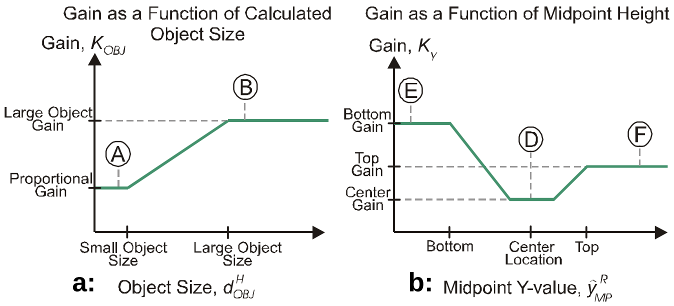

Figure 5. Functions for object size and midpoint gains

The modified size and position of the virtual object in the robotic hand frame are then used in conjunction with the object orientation (calculated in the projected hand frame) to compute the necessary commanded robotic fingertip positions.

In summary, the virtual object based mapping method provides a solution to the problems associated with the kinematic differences between the human and robot hands. Using a virtual three-dimensional object yields additional information about the intent of the manipulation compared to a simple planar projection of tip positions. The use of non-linear gains on the object size and midpoint help match the workspace while mapping the human pinch point to the robot's ideal manipulation point.

By customizing the object gain parameter and the midpoint mapping parameter, as well as parameters for the transformation from the human hand frame to the robot hand frame, a particular mapping can be constructed for each user which allows him/her to control the robot in a relatively intuitive manner. The parameters associated with the gain functions, such as gain values, slopes between gains, and locations of gain regions, can also be modified to suit a particular user's hand. Because the gain of the position is based on the object and not on projected fingertip positions, if a user rolls a ball about his/her comfortable manipulation location, the robot will ideally roll a slightly larger ball about its optimal point in the center of the workspace, see Fig. 4C.

4.3 Mapping Method Results

The mapping methods presented above were tested using the planar robotic hand described previously. The data presented here represents a typical user's mapping results. The user's hand was first calibrated to the kinematic model using the procedure developed in Section 3. The user moved his fingers through the range of postures shown in Fig. 4 (motions were made independent of the mapping method used). For comparison, both methods attempted to map the postures shown in Figs. 4E and 4F to the vertical motion on the robot; this is also important for making the mapping intuitive. The parameters of each method were adjusted using a virtual display of the robotic hand, before allowing the user to directly control the robot.

Figure 6a represents achievable human positions as mapped under the point-to-point method in the robotic hand frame. The mapped positions of the index and thumb are plotted over the workspace of the robotic hand, where the index and thumb correspond to the left and right robot finger, respectively.

From the figure, one can see that the index finger maps to much of the left robot finger workspace. The limitations of using the planar projection of the thumb tip position are also displayed. The thumb motion is mapped to a relatively small area roughly along a line crossing the workspace.

The large number of achievable points significantly outside the workspace leads to a distortion in the mapping. For commanded positions outside the robot's workspace, the robot will go to the closest possible position. Thus, for many human finger motions the robot is driven to the edge of its workspace.

The mapped human pinch point is also illustrated in the figure. The pinch point corresponds to a point closer to the edge of the robot's workspace, which limits the robot's manipulation range at that location.

Achievable human positions mapped under the object based mapping are also shown in Fig. 6b. Again, the mapped positions of the index and thumb are plotted with the workspace of the robot. Notice that the reachable positions for the index finger lie almost completely within the left finger workspace, and the motion of the right finger (thumb) is greatly expanded. The figure also shows that the mapped human pinch point location and the preferred robotic manipulation point match directly, allowing greater manipulation range.

Some aspects of the object mapping method are unique to our robotic hand. Because our robot is a two fingered hand with two degrees of freedom per finger, symmetric motions increase the range of motion in rolling manipulations. However, when a human rolls an object between the index and thumb, the motion of the thumb tends to be smaller than that of the index. The motion disparity is also the case when separating the fingers from the natural pinch position. To map this asymmetry to our symmetrical robot we shift the midpoint of the object closer to the thumb to accentuate thumb motion. By shifting the midpoint, we avoid using a non-linear gain function on the midpoint x-position. As a result, the release posture (Fig. 4B) is

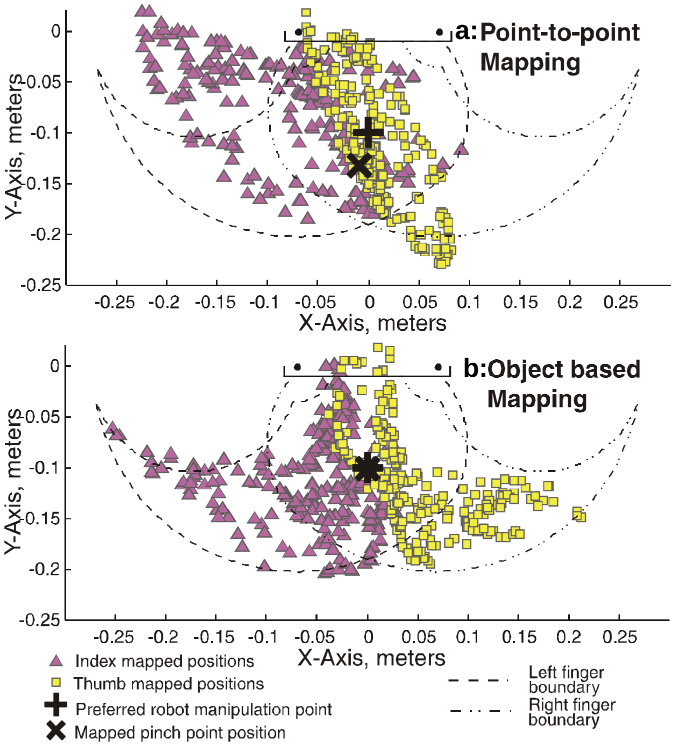

Figure 6. Mapping results for a typical user

symmetrical in the robot's workspace.

While the virtual object mapping concept has been demonstrated using a particular robotic hand, the mapping method is a general technique and is extensible to most planar robot hands. By modifying the mapping parameters, particularly the non-linear gain functions, the motions of the human hand can be mapped to a different robot workspace. If a robot hand has additional degrees-of-freedom per finger, there is not a unique configuration to achieve a desired virtual object position. The redundancy may be used to control contact location, extend the reachable workspace or other optimizations.

One caveat that must be observed when performing the non-linear mapping is that, in some cases, relatively small motions of the human fingers could result in large motions of the robot fingers. Thus it may be desirable to plot corresponding velocity ellipsoids for various regions of the workspace to guard against this possibility.

5.0 FUTURE WORK

Certain issues regarding the calibration of the user's hand to the CyberGlove continue to be explored. These include the effect of non-orthogonal thumb trapeziometacarpal rotation axes, methods to insure numerical stability of the least squares regression, and extension to all five fingers of the human hand. In addition, current study is underway to more thoroughly explore the significance of the various possible sensor cross-coupling combinations.

Further investigation of the object mapping method will focus on using the intended manipulation information to command the robot to perform specific manipulation actions. Manipulation actions, such as rolling, would use robot fingertip contact information and rolling kinematics to accurately create commanded positions. Other actions, such as object exploration, could give the user direct control over an individual finger tip.

6.0 ACKNOWLEDGMENTS

This research is funded by STTR grant N96T003 from the Office of Naval Research with Virtual Technologies, Inc., and the Charles M. Pigott fellowship. The authors thank Virtual Technologies, particularly Daniel Gomez and Marc Tremblay, for their assistance in developing the test bed system.

Additionally, the authors would like to thank Dr. Herb Rauch, for helping develop parameter variation limitation.

7.0 REFERENCES

[1] An, K.N., Chao, E.Y., Cooney, W.P., and Linscheid, R.L., 1979, "Normative Model of Human Hand for Biomechanical Analysis," J. Biomechanics, vol. 12, pp. 775-788.

[2] Bennet, D.J., Hollerbach, J.M., 1990, "Closed-loop Kinematic Calibration of the Utah-MIT Hand," in Experimental Robotics I: The First International Symp., V. Hayward, O. Khatib, (eds.), Springer-Verlag, N.Y., pp. 539-552.

[3] Cooney, W.P., Lucca, M.J., Chao, E.Y.S., Linschied R.L., 1981, "The kinesiology of the thumb trapeziometacarpal joint," J. Bone Joint Surg. 63A:1371-1381.

[4] Fischer, M., van der Smagt, P., Hirzinger, G., 1998, "Learning Techniques in a Dataglove Based Telemanipulation System for the DLR Hand," 1998 IEEE ICRA, pp1603-1608.

[5] Hollister, A., Buford, W.L., Myers, L.M., Giurintano, D.J, Novick, A., 1992, "The Axes of Rotation of the Thumb Carpometacarpal Joint," J. of Orthopaedic Res., vol. 10, pp. 454-460.

[6] Khatib, O., 1987, "Unified Approach for motion and force control of robot manipulators: the operational space formulation," IEEE J. of Robotics and Automation, vol. 3, no. 1, pp. 43-53.

[7] Kramer, J.F., "Determination of Thumb Position Using Measurements of Abduction and Rotation," U.S. Patent #5,482,056.

[8] Kuch, J.J., Huang, T.S., 1995, "Human Computer Interaction via the Human Hand: A Hand Model," 1995 Asilomar Conf. on Signals, Systems. and Computers. pp. 1252-1256.

[9] Rohling, R.N., Hollerbach, J.M., 1993, "Calibrating the Human Hand for Haptic Interfaces," Presence, vol. 2 no. 4, pp. 281-296.

[10] Rohling, R.N, Hollerbach, J.M., Jacobsen, S.C., 1993, "Optimized Fingertip Mapping: A General Algorithm for Robotic Hand Teleoperation," Presence, vol. 2 no. 3, pp. 203-220.

[11] Turner, M.L., Gomez, D.H. Tremblay, M.R. and Cutkosky, M.R., 1998, "Preliminary Tests of an Arm-Grounded Haptic Feedback Device in Telemanipulation," 1998 ASME IMECE Symp. on Haptic Interfaces. pp.145-149.

[12] Turner, M.L., Findley, R.P., Griffin, W.B., Cutkosky, M.R., Gomez, D.H., 2000, "Development and Testing of a Telemanipulation System with Arm and Hand Motion," Accepted to 2000 ASME IMECE Symp. on Haptic Interfaces.

[13] Wampler, C.W., Hollerbach, J.M., Arai, T., 1995, "An Implicit Loop Method for Kinematic Calibration and its Application to Closed-chain mechanisms," IEEE Trans. Robotics and Automation, vol. 11, no. 5, pp. 710-724.

[14] Wright, A.K., Stanisic, M.M., 1990, "Kinematic Mapping between the EXOS Handmaster Exoskeleton and the Utah-MIT Dextrous Hand," 1990 IEEE Int'l Conf. on Systems Engineering, pp. 809-811.