

Humanoid Robot Teleoperation for Tasks with Power Tools

Rowland O'Flaherty[†], Peter Vieira[†], M.X. Grey[†],
Paul Oh[‡], Aaron Bobick[†], Magnus Egerstedt[†], and Mike Stilman[†]



Fig. 1. Picture of Hubo and operator just after using telemanipulation to cut a rectangular hole out of a cardboard wall.

I. INTRODUCTION

In this paper we present, analyze and evaluate the teleoperation of a humanoid robot platform using a feedback controller and the inverse kinematics of the limbs. The particular humanoid platform that is used in this work is the HUBO2+ platform (Hubo). Teleoperation of the arms is performed using a Polhemus FASTRAK motion tracking system for tasks in order to provide preliminary proof of concept of methods that will later be used to execute tasks autonomously in search and rescue missions. Hubo has six degrees of freedom (DOF) in each arm and leg, which is the minimum number of DOF required to control the three position and orientation variables. We have developed algorithms to control Hubo in the end-effector workspace via teleoperation using a closed-form kinematic solution and a basic feedback controller in order to execute simple tasks, such as using a rotary tool to cut out holes in a cardboard wall, see Fig. 1.

Ali et. al. presented a closed-form solution for the inverse kinematics (IK) of the limbs of the HUBO2+ robot platform [1]. They used a reverse decoupling mechanism method by viewing the kinematic chain of a limb in reverse order and decoupling the position and orientation. The authors then used the inverse transform method to compute eight possible solutions, with the correct solution selected based on joint limits and constraints. In working through their solution, discrepancies were found in the calculations. We corrected the errors and solved for the IK of all four limbs for our HUBO2+ humanoid robot.

There has been significant recent work on teleoperation of humanoid robots, such as [2] and [3], but few of them use tools to perform practical tasks. There is research being conducted on robots using tools, for instance [4] and [5], but these focus more on the design of robots for using tools, rather than the actual utilization of tools to perform specific tasks.

[†]Authors are affiliated with Georgia Institute of Technology, Atlanta, GA, 30332, USA. {rowlandflaherty, plvieira, mxa, ragnar}@gsa.tech.edu and {afb, ms}@iclm.gatech.edu.

[‡]Author is affiliated with Drexel University, Philadelphia, PA, 19104, USA. paul@coe.drexel.edu.

Our teleoperation of the HUBO2+ robot goes beyond design into the realm of performing useful tasks.

II. HUBO GEOMETRY AND KINEMATICS

In order to control Hubo using workspace control via teleoperation both forward and inverse kinematic solutions are required. The solution to this problem IK of the arms, and then joint coordinate frames are shown in Fig. 2 and of each link is shown in Table I. The DH parameters (the standard convention) for the arms and legs are shown in Table II and Table III, respectively.

TABLE I. Link Lengths

Link	Length (mm)	Link	Length (mm)
- L \	215	-> \	187
- \	179	-4 L \	88
- \	182	-4 \	182
- \	121	-4 \	300
- & \	100	-4 \	300
		-4 \	

TABLE II. DH parameters

Co. dr. Frame	a	α	D	θ
1	$\pi/2$	$\pi/2$	0	0
2	$\pi/2$	$\pi/2$	0	0
3	$\pi/2$	$-\pi/2$	0	$-\pi/2$
4	$\pi/2$	$\pi/2$	0	0
5	$\pi/2$	$-\pi/2$	0	$-\pi/2$
6	$\pi/2$	0	$-\pi/2$	0

Fig. 2. Hubo coordinate frames

The forward kinematics problem is that of solving for the end-effector orientation and position given the joint angles. This is easily solved using the geometry of the robot which is specified in the DH parameters. The general homogeneous transformation from one link to the next given the DH parameters is represented in matrix form as:

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where ${}^{i-1}T_i$ is the transformation from coordinate frame J_{i-1} to frame J_i . The base frame for the arm is at the neck, and its transformation to the first shoulder joint is

$${}^N T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & l_{A1} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

An additional transformation ${}^6 T_E$ is used for the transformation from the hand to the end-effector¹. In order to calculate the forward kinematics (FK) the six transformation matrices from each joint are pre-multiplied to obtain the position and orientation of the end-effector relative to the shoulder. We define the transformation from the shoulder to the hand as

$${}^0 T_6 = {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6. \quad (3)$$

Thus when solving for the forward kinematics ${}^0 T_6$ must be pre-multiplied by ${}^N T_0$ and post-multiplied by ${}^6 T_E$. Therefore, the FK is calculated as

$${}^N T_E = {}^N T_0 {}^0 T_6 {}^6 T_E. \quad (4)$$

The inverse kinematics problem is how to solve for the joint angles given the end-effector orientation and location, specified as ${}^N T_E$. This is a much harder problem because there are multiple solutions. When solving the inverse kinematics of a manipulator, Pieper [6] indicates that a closed-form solution exists if three consecutive joint axes of the manipulator are parallel to one another, or intersect at a single point. The three shoulder joint axes on the Hubo intersect at a single point for the arm and the three hip joints intersect at a single point for the leg, therefore a closed-form kinematic solution exists for each.

We will solve the IK problem from the shoulder to the hand by using the transformation ${}^0 T_6$. This is obtained by pre-multiplying ${}^N T_E$ by ${}^0 T_N$ and post-multiplying by ${}^6 T_E$. Let us write the ${}^0 T_6$ obtained from ${}^N T_E$ as

$${}^0 T_6 = \begin{bmatrix} s_{\theta_4} & c_{\theta_4} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where s_{θ_4} and c_{θ_4} are the unit vectors along the principal axes of the hand frame and O_N is the position vector describing the location of the hand relative to the shoulder. These three unit vectors describe the orientation of the hand coordinate frame relative to the shoulder coordinate frame. The vectors n , s , a , and r represent the normal vector, sliding vector, approach vector, and position vector of the hand, respectively [7].

Using this knowledge, the arm can be viewed in reverse so that the last three joints make up the shoulder, thus the position and orientation of the shoulder frame can be described relative to the hand frame. This new position vector, r is only a function of θ_4 , θ_5 and θ_6 , and thus decouples the arm into position and orientation components. The IK-problem is solved in this reverse method by taking the inverse of both sides of (5).

$${}^0 T_6^{-1} = ({}^0 T_6)^{-1} = \begin{bmatrix} s' & a' & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

As in [1], we solved for the joint angles using the reverse method. The joint solutions are given below. The details on how these solutions were derived are in the following tech report [8].

First, the three lower joint angles of A_4 , A_5 and E_N are solved for and then the upper joint angles of A_1 , A_2 and A_3 are solved for. The solutions are as follows:^{2 3}

²The sine and cosine of an angle θ is abbreviated s_{θ} and c_{θ} respectively
³The interval between $-\pi$ and π .

¹This transformation is defined for a drill end-effector in the Section III.

$$\begin{aligned}
\theta_4 &= \text{atan2}(\pm \text{real}(\sqrt{1 - C_4^2}), C_4), \\
C_4 &= \frac{(p'_x + l_{A4})^2 + p'_y{}^2 + p'_z{}^2 - l_{A2}^2 - l_{A3}^3}{2l_{A2}l_{A3}}, \\
\theta_5 &= \text{atan2}(S_5, \pm \text{real}(\sqrt{1 - S_5^2})), \quad S_5 = p'_z / (S_4 l_{A2}); \\
\theta_6 &= \text{wrapToPi}(\text{atan2}(S_4 C_5 l_{A2}, -C_4 l_{A2} - l_{A3}) - \psi), \\
\psi &= \text{atan2}(p'_y, p'_x + l_{A4}); \\
\theta_2 &= \text{atan2}(S_2, \pm \text{real}(\sqrt{1 - S_2^2})), \\
S_2 &= a'_x(C_4 C_6 - C_5 S_4 S_6) - a'_y(C_4 S_6 + C_5 C_6 S_4) \\
&\quad - a'_z S_4 S_5;
\end{aligned}$$

$$\begin{aligned}
\theta_3 &= \text{atan2}(S_3, C_3), \\
S_3 &= g_{313} = -a'_x(C_6 S_4 + C_4 C_5 S_6) + a'_y(S_4 S_6 - C_4 C_5 C_6) \\
&\quad - a'_z C_4 S_5, \\
C_3 &= g_{333} = -a'_x S_5 S_6 - a'_y C_6 S_5 + a'_z C_5; \\
\theta_1 &= \text{atan2}(S_1, C_1), \\
S_1 &= -n'_x(C_4 C_6 - C_5 S_4 S_6) + n'_y(C_4 S_6 + C_5 C_6 S_4) + \\
&\quad n'_z S_4 S_5, \\
C_1 &= s'_x(C_4 C_6 - C_5 S_4 S_6) - s'_y(C_4 S_6 + C_5 C_6 S_4) - \\
&\quad s'_z S_4 S_5.
\end{aligned}$$

There are two solutions for θ_2 , θ_4 , and θ_5 , which generate eight total solutions to the arm IK. When the goal position is outside the feasible workspace of the limb, the joint solutions will have imaginary parts. To deal with this, we take only the real part, which in turn gives the solution that is closest to the desired position. Furthermore there are five cases that result in singularities, and details of the solution methods can be viewed in [1]. The following are our final equations for each case.

5+v ! ' ^ & ' ! \$ # When $\theta_4 = 0$, joints θ_3 and θ_5 are collinear, thus an infinite number of solutions exist to orient the end-effector in the desired orientation. We set θ_3 to its previous value and then the difference between the θ_5 and θ_3 , θ_T , is set to the desired orientation by solving the equation,

$$\begin{aligned}
\theta_T &= \text{atan2}(-C_6 a'_y - S_6 a'_x, a'_z) \\
\text{if } C_2 \neq 0 \text{ then } \text{wrapToPi}(\theta_T) &= \theta_T + \pi \\
\theta_5 &= \text{wrapToPi}(\theta_T - \theta_3).
\end{aligned}$$

2_A^2 ! ' ^ ! \$ ' ! \$ # When $\theta_2 = \pi/2$ (for the left arm) or $-\pi/2$ (for the right arm) joints θ_1 and θ_3 are collinear. The same approach as above is taken, resulting in the difference equation,

$$\begin{aligned}
\theta_T &= \text{atan2}(S_6 s'_y - C_6 s'_x, S_6 n'_y - C_6 n'_x) \\
\text{if } S_4 \neq 0 \text{ then } \theta_T &= \theta_T + \pi \\
\text{Left Arm: } \theta_1 &= \text{wrapToPi}(\theta_T + \theta_3) \\
\text{Right Arm: } \theta_1 &= \text{wrapToPi}(\theta_T - \theta_3).
\end{aligned}$$

3_A^2 ! ' ^ " ' ! \$ # When $\theta_4 = \pi$ and $\theta_2 = \pi/2$ (for the left arm) or $-\pi/2$ (for the right arm) joints θ_1 ,

θ_3 and θ_5 are collinear. The same approach as above is taken, resulting in the difference equations,

$$\begin{aligned}
\text{Left Arm: } \theta_T &= \text{atan2}(n'_z, -s'_z) \\
\theta_5 &= \text{wrapToPi}(\theta_1 - \theta_3 - \theta_T), \\
\text{Right Arm: } \theta_T &= \text{atan2}(-n'_z, s'_z) \\
\theta_5 &= \text{wrapToPi}(\theta_T - \theta_1 - \theta_3). \tag{7}
\end{aligned}$$

After solving for the joint angles, eight solutions exist. To select the correct solution, joint constraints are considered. Finally, when solving for the inverse kinematics the inverse of the neck-to-shoulder transformation matrix, 0T_N , must first be pre-multiplied and the inverse of the hand transformation matrix, ${}^E T_6$, must be post-multiplied in order to compute the IK using the DH parameters.

As with the arm the forward kinematics of the leg, ${}^N T_F$, are straightened forward once the DH parameters are derived. The DH parameters for the right leg are shown in Table III and again (1) is used find the transformation between adjacent joint coordinate frames. The base frame for the leg is at the waist, and its transformation to the first hip joint is,

$${}^N T_w = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & l_{L1} \\ 0 & 0 & 1 & 0 & 1 & -l_{L2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

We define the transformation from the hip to the foot as

$${}^0 T_6 = {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6. \tag{9}$$

Thus when solving for the forward kinematics ${}^N T_F$ must be pre-multiplied by ${}^N T_w {}^W T_0$ and post-multiplied by ${}^6 T_F$. Therefore, the FK is calculated as

$${}^N T_F = {}^N T_w {}^W T_0 {}^0 T_6 {}^6 T_F. \quad \text{'kK-v}$$

TABLE III. DH 9 ; 6 (? (=8) ? + (\5 (= \

R i g h t H p p a r a m e t e r s				
Co ord.Fr a m e	θ_i	α_i	a_i	d_i
	\		0	0
	\		0	0
	\	0	M 4	\ 0
	\	0	M 4	0
	\	$\pi/2$		0
	\	0	M 4	0

Similar to the IK for the arm the IK for the leg is solved from the hip to the foot by using the transformation ${}^0 T_6$. This is obtained by pre-multiplying ${}^N T_F$ by ${}^0 T_w {}^W T_N$ and post-multiplying by ${}^F T_6$. Let us write ${}^0 T_6$ for the legs in the same way as for the arms in (5).

As with the arm, the three hip joint axes in the leg on Hubo intersect at a single point, therefore a closed-form kinematic solution exists. Thus the procedure for solving the IK for legs is similar to the arms. As with the IK of the arm the details of deriving the solutions for the joint angles are given in the following tech report [8].

First, the three lower joint angles of L_4 , L_5 , and L_6 are solved for, and then the three upper joint angles of L_1 , L_2 , and L_3 are solved for. The solutions are as follows:

$$\theta_4 = \text{atan2}(\pm \text{real}(\sqrt{1 - C_4^2}), C_4),$$

$$C_4 = \frac{(p'_x + l_{L5})^2 + p'_y{}^2 + p'_z{}^2 - l_{L3}^2 - l_{L4}^3}{2l_{L3}l_{L4}};$$

$$\theta_5 = \text{wrapToPi}(\text{atan2}(-p'_z, \pm \text{real}(\sqrt{(p'_x + l_{L5})^2 + p'_y{}^2}) - \psi),$$

$$\psi = \text{atan2}(S_4 l_{L3}, C_4 l_{L3} + l_{L4});$$

$$\theta_6 = \text{atan2}(p'_y, -p'_x - l_{L5});$$

$$S_2 = S_6 a'_x + C_6 a'_y,$$

$$\theta_2 = \text{atan2}(S_6 a'_x + C_6 a'_y, \pm \text{real}(\sqrt{1 - (S_6 a'_x + C_6 a'_y)^2}));$$

$$\theta_1 = \text{atan2}(S_6 s'_x + C_6 s'_y, S_6 n'_x + C_6 n'_y);$$

$$C_2 S_1 = S_6 s'_x + C_6 s'_y,$$

$$C_2 C_1 = S_6 n'_x + C_6 n'_y;$$

$$\theta_3 = \text{wrapToPi}(\theta_{345} - \theta_4 - \theta_5),$$

$$\theta_{345} = \text{atan2}(a'_z, C_6 a'_x - S_6 a'_y).$$

If $C_{45} l_{L3} + C_5 l_{L4} \neq 0$ then $\theta_6 = \text{wrapToPi}(\theta_6 + \pi)$. If $C_2 \neq 0$ then $\theta_1 = \text{wrapToPi}(\theta_1 + \pi)$.

We now have the inverse kinematic solution to the right leg. The above solution can be applied to the left leg by changing $+l_{L1}$ to $-l_{L1}$ in the base-to-hip transformation matrix in (8).

Like the arm there are two solutions for θ_2 , θ_4 and θ_5 , which generate eight total solutions to the leg IK. As with the arm, if the goal position is outside the feasible workspace of the limb the joint solutions will have imaginary parts and only the real part is used.

% \ ! ' " \$ ' .

For the inverse kinematics of each of the arms and the legs there are eight joint solutions. The sum of squared joint values is the primary metric that is used in picking one of the eight solutions. Choosing the solution that minimizes this metric is the solution that is "closest" to the zero position of the joints. This works well if at least one of the solutions has all of its joints values within the joint limits (Table IV).

If none of the solutions have all the joint values within the limits then there is no solution that satisfies the desired pose (orientation and location). To get the end-effector to a position as close as possible to the desired position the joint values in all the solutions are capped at the closest joint limit value. Each of the solutions are then given to the FK to calculate the end-effector location with the capped joint values. The solution that gets the end-effector position the closest to the desired position is used. If none of the joint solutions get the end-effector within 5 cm of the desired position then the previous joint values are used.

III. IMPLEMENTATION

It takes more than just solving for the kinematics to actually have Hubo do something meaningful. In this section we describe various important considerations and algorithms that are needed to implement teleoperation on Hubo.

TABLE IV. 2 8 / ? \ 5 / 6 / ? \ ? \ (\ ; 6 = \ 7 \$ \ 5 (* = \

Joint	Arms				Legs			
	Left		Right		Left		Right	
	min	max	min	max	min	max	min	max
1	-2.0	2.0	-2.0	2.0	0	1.8	-1.8	0
2	-0.3	2.0	-2.0	0.3	0	0.6	-0.6	0
3	-2.0	2.0	-2	2.0	-1.3	1.4	-1.3	1.4
4	-2.5	0	-2.5	0	0	2.5	0	2.5
5	-2.5	2.0	-2.5	2	-1.3	1.8	-1.3	1.8
6	-1.4	1.2	-1.4	1.2	-0.3	0.2	-0.2	0.3

The workspace of Hubo's arms is limited because the arms are short. To increase the vertical workspace of Hubo's arms, Hubo can use its legs to move its body up and down. Getting Hubo to move its end-effector to a desired location that requires its body to move up or down requires some form of inverse kinematics for all 12 joints of each arm and leg pair (left and right). To simplify this we assume that both hands will be working at the same height level and have developed an algorithm that uses the decouple inverse kinematics of the arms and the legs as described in Section II. To summarize this algorithm, Hubo keeps its hands at shoulder level and moves its body up and down with its legs. If a desired pose is below or above the points the body can be elevated to then the arms will move down or up from the fully lowered or raised body positions.

The algorithm is as follows: (1) Get desired hand pose and extract height information; (2) Use leg IK to move the shoulder to as close as possible to desired height; (3) Use arm IK to move hand to desired hand pose.

." .

Currently the motor control boards on Hubo only support position control. The gains for this position control are extremely high to deal with the external forces that the joints may encounter. Due to these high gains, giving arbitrary joint angles is not possible because the joint will move to the position in a violent manner. Therefore, a feedback controller algorithm is implemented using nominal maximum velocities and accelerations in order to achieve fluid, safe motion. This algorithm works by giving the motor control board for a given joint a trajectory to follow from its current position to the desired position that minimizes the jerk on the joints. This allows for the joint to reach the desired position by accelerating and decelerating in smooth fashion.

In order to achieve any of these task, balancing is a necessary requirement. Four sensor values are used on Hubo to achieve balancing. We obtain the angles, ϕ_x and ϕ_y about the x- and y-axes that the waist is at relative to the vertical z-axis from an inertial measurement unit (IMU) in his waist, and the moments, M_x and M_y , about each ankle from the force/torque sensors in the feet. Often, when a humanoid plants its feet on the ground it creates a closed loop, which in turn can result in dangerously high torques if the feet happen to slide or shift relative to each other while still on the ground. This can cause the motors to draw extremely high current and potentially burn the motors out. One instance when this is an important consideration is when first placing the robot on the ground. If

its feet are not both parallel to the ground, and the ankle motors are being used for balancing, then this phenomenon can arise. To avoid this problem we devised a method to even out the feet and then balance such that the ankle motors comply with the moments M_x and M_y but resist the IMU angles ϕ_x and ϕ_y . We achieve this by setting the compliant term for the ankle angular velocities equal to a gain multiplied by the moment, and the resistive term equal to a gain multiplied by the IMU angle. Thus, the ankle angular velocities are

$$\omega_r = \frac{1}{K_r} K_c \phi_x - K_c M_x, \quad (11)$$

$$\omega_p = \frac{1}{K_p} K_c \phi_y - K_c M_y \quad (12)$$

where ω_r and ω_p are the angular velocities of the ankle roll and pitch joint motors, respectively, and K_r and K_c are the resistive and compliant gains, respectively. These angular velocities are sent to the feedback controller as inputs. For our gains we chose $K_r = 0.001$ and $K_c = 0.001$. These gains work very well, but in order to take into account added weight to the robot, from tools or objects it is holding, the force in the negative z-direction could be factored into the equation so that the compliance gain would be inversely proportional to the weight.

\ " ' .

To control the arms of the Hubo robot via teleoperation a Polhemus FASTRAK motion tracking device is used, which utilizes 6-DOF sensors. FASTRAK provides three position values and three orientation values of a small sensor relative to a reference frame as it moves through space. These readings are given in real time with virtually no latency (4 ms). FASTRAK allows for up to four sensors to be used simultaneously. To control both of Hubo's arms two FASTRAK sensors are used that map the pose (location and orientation) of a human user's hands to the pose of Hubo's hands. Thus, this allows for real time teleoperation of Hubo's hands by a human operator.

The FASTRAK system returns homogeneous transform matrices of the sensors' respective poses for each instance in time. To obtain calibrated relative position readings of the operator's hands the first sensor readings are used as offset location values used to correct all proceeding sensor readings. These corrected transformations are given to the inverse kinematics algorithm to get Hubo's reference joint values. These joint values are fed in to the joint control algorithm, which moves Hubo's hands to the same relative pose as the operator's hands.

IV. EXPERIMENTAL SETUP

A task that our team is focused on is cutting through walls. These potential abilities give humanoid robots a large advantage over mobile ground robots during search and rescue missions in hazardous environment. To show that Hubo is capable of using power tools to cut through a wall we have equipped Hubo with a cordless, straight-handled drill, see Fig. 3. Using this drill we demonstrate that Hubo can cut through a cardboard wall. In this setup the middle of the drill bit is the location of the end-effector coordinate system with the x-direction pointing out the end of the drill bit. The

transformation from the hand coordinate frame to the end-effector with the drill is

$${}^6T_E = \begin{bmatrix} y \cos \phi & 0 & \sin \phi & l_E \cos \phi \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & -l_E \sin \phi \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (13)$$

where $\phi = \pi / 4$ and $l_E = 1$ cm.

In order for the drill to be the most effective at cutting through the cardboard the drill bit should be orthogonal to the surface. This means the drill bit should have an orientation that is aligned with neck coordinate frame of the robot given that the robot is standing square to the wall. The workspace in both the vertical and horizontal directions of the end-effector is limited to this orientation. The limitation in the vertical direction is less of a concern because the legs can be used to move the entire body of Hubo up and down as described in Section III-A. The size of this workspace changes for different distances in front of Hubo. In order for Hubo to cut the largest hole possible we want to find the distance that Hubo should stand from the wall that maximizes the horizontal workspace of the end-effector with the desired orientation.

Fig. 3. Drill end-effector coordinate frames

To find this optimal distance simulations were performed to map the workspace of the end-effector. The results of these simulations are shown in Fig. 4. It was found that a distance of 484 mm from the neck to the wall gives the largest horizontal workspace at 390 mm. This is at a vertical distance of 75 mm above the neck coordinate frame. The joint values for the extreme points in the horizontal direction are the following: far right (minimum y value) $s = [-0.9, 0.1, 0.6, 1.8, 0.6, 0.84, 82.9, 4]$ and the far left (maximum y value) $s = [-1.8, 0.1, 1.2, 1.3, 0.3, 1.2, 8.9, 3, -1.1, 18.4, 8, 2.6]$.

The orientation of the drill is set to have the drill bit orthogonal to the wall. In addition, the distance from the wall is also set to give the maximum horizontal workspace. Therefore, using teleoperation via FASTRAK to control the orientation and the distance in the x-direction is not possible. The teleoperation in this case only defines the displacement in the y-direction and the z-direction.

In this experiment the wall is made from cardboard and is set 48 cm in front of Hubo. A human operator controlling Hubo via FASTRAK cuts a rectangular hole in the cardboard wall. Both the input from the human is recorded as well as the actual end-effector location. The results are discussed in the next section.

