A Motion Retargeting Method for Effective Mimicry-based Teleoperation of Robot Arms

Daniel Rakita, Bilge Mutlu, Michael Gleicher
Department of Computer Sciences, University of Wisconsin–Madison
1210 West Dayton Street, Madison, WI 53706 USA
{rakita, bilge, gleicher}@cs.wisc.edu

ABSTRACT

In this paper, we introduce a novel interface that allows novice users to effectively and intuitively tele-operate robot manipulators. The premise of our method is that an interface that allows its user to direct a robot arm using the natural 6-DOF space of his/her hand would afford effective direct control of the robot; however, a direct mapping between the user's hand and the robot's end effector is impractical because the robot has different kinematic and speed capabilities than the human arm. Our key technical idea that by *relaxing* the constraint of the direct mapping between hand position and orientation and end effector configuration, a system can provide the user with the feel of direct control, while still achieving the practical requirements for telemanipulation, such as motion smoothness and singularity avoidance. We present methods for implementing a motion retargeting solution that achieves this relaxed control using constrained optimization and describe a system that utilizes it to provide real-time control of a robot arm. We demonstrate the effectiveness of our approach in a user study that shows novice users can complete a range of tasks more efficiently and enjoyably using our relaxed-mimicry based interface compared to standard interfaces.

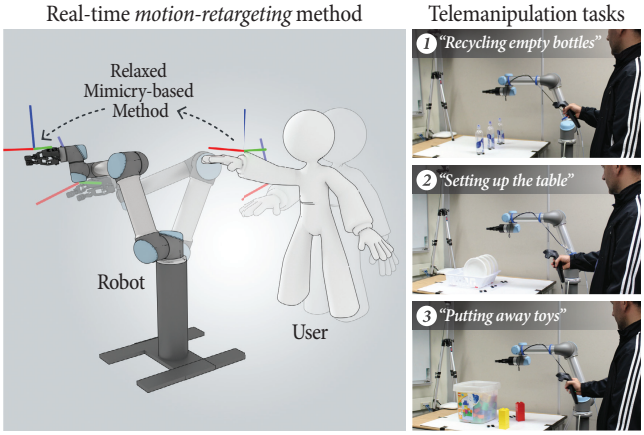

Figure 1: We propose a mimicry-based telemanipulation method that uses relaxed constraints on the mapping between a user and a robot arm as an effective real-time control mechanism and evaluate it in three telemanipulation tasks that follow a home care scenario.

1. INTRODUCTION

Telemanipulation systems, where a human user controls a robot arm from a distance, are valuable in scenarios where automation is impractical, where human judgment is essential, or where having the user engaged in the task is desirable [25]. However, controlling a robot arm from a distance introduces a number of challenges, including limitations in the capabilities of human operators and technical challenges in effectively translating human operator commands into robot actions. While shared control strategies are often used to improve performance by providing higher-level control through automation, such strategies have limited applicability, as they require prior knowledge of user goals, target object locations, and feasible robot trajectories [29, 35, 24, 6, 1, 36, 8]. Therefore, many applications involving telemanipulation must rely on direct control of robot arms. While successful real-world applications of direct control exists, such as tele-surgery, these interfaces are designed for expert users with substantial skill and training. Broadening applications and the potential users of telemanipulation, such as family members providing remote home care for an elder parent, requires interfaces that are intuitive and effective to novice users.

Our goal is to provide a direct control interface that will allow novice users to effectively teleoperate robot arms and will support applications such as remote home-care or tele-nursing that require considerable human judgment and involvement without the opportunity for extensive training in system operation [18, 14]. We posit that enabling users to work in the "natural" space of their arms will allow them to draw on their inherent kinesthetic sense and ability to perform tasks in controlling a robot. That is, mapping the movement of the user's arm, particularly the position and orientation of their hand, to the movement of the robot can allow intuitive and effective control of the robot without significant training. Capturing arm motion has become practical with the recent emergence of a variety of technologies, including visual tracking (e.g., the Microsoft Kinect[1]) and wireless controllers (such as the HTC Vive[2] shown in Figure 1). However, translating captured motion into robot movements in a manner that successfully meets the demands of robot control remains a key challenge. Furthermore, because we lack satisfactory solutions to this challenge, the premise that such a mapping between human and robot arms serves as effective interface for direct control has not been validated.

The key technical challenge stems from differences in the characteristics and capabilities of robot and human arms, making a direct

[1]Microsoft Kinect: https://dev.windows.com/en-us/kinect
[2]HTC Vive: http://www.vive.com/us/

DOI: http://dx.doi.org/10.1145/2909824.3020254

mapping unworkable for a number of reasons. First, kinematic differences between the arms mean that small movements of the human arm may require large, abrupt movements of the robot arm. Second, joint and velocity limits make smooth tracking of the user's movement difficult, leading to robot arm motion that is less predictable and harder to control. Finally, because the physiology of the user's arm may differ greatly from that of the robot arm, and the user has no kinesthetic sense of the configuration of the robot's arm, the user may inadvertently guide the robot into kinematic singularity. A practical telemanipulation system must not only mimic the user's movements well enough that they have the feeling of direct control, but it must also generate smooth, continuous robot motions that avoid singularity.

In this paper, we present a telemanipulation method that addresses this challenge to provide novice users with an easy-to-learn and easy-to-use direct control interface. We argue that the benefits of using the user's natural arm space as the input space may be obtained without considering position and orientation of the user's hand as a hard constraint. By appropriately *relaxing* the mapping between the human input and the robot's movement, we can approximate direct control solutions that support an easy-to-use interface while offering the flexibility to achieve motion with minimized joint velocities, smooth positional tracking, and no kinematic singularities. To accomplish these goals, we developed an optimization-based inverse kinematic solver that allows for relaxed constraints that can be adjusted dynamically based on what information is currently important. We have implemented this method in a prototype system that allows naïve users to control a UR5[3] robot arm using input devices including a wireless 6-DOF controller (Figure 1). We conducted a user study to demonstrate the feasibility of this method and to compare its effectiveness in enabling novice users to perform manipulation tasks using a 6-DOF stylus and a touch-screen interface.

The specific contributions of our work include (1) a mimicry-based robot arm control method that serves an easy-to-learn and easy-to-use direct manipulation interface; (2) solutions to technical challenges involved in effectively mapping human and robot arm motions and their implementation into an integrated robot control system (§4 and §5); and (3) a demonstration of the feasibility, effectiveness, and user perceptions of the control method relative to alternative, commonly used control interfaces through a user study (§6).

2. RELATED WORK

In this work, we draw on prior work in the robotics literature on kinematic mapping, human-to-robot motion remapping, and control principles. We also draw on the animation literature for work exploring retargeting motion to new characters of different scales or structures.

Human-to-robot Motion Remapping — Moving robots using human motion has a long history dating back to the early turn-crank master-slave device developed by Goertz [12] (for a survey on teleoperation devices and methods, see the work by Hokayem and Spong [15]). Since then, research has explored various scenarios that involve controlling robots using human motion. Pollard et al. [27] present a method to transfer human motion data and accompanying stylistic motion qualities to a humanoid robot, despite the discrepancy between human and robot degrees of freedom, joint velocity limits, and joint rotation ranges. Suleiman et al. [32] present a robot imitation approach and show that their analytical solution to the optimization converges quickly and effectively. Our work uses an optimization framework to address degree-of-freedom and joint-velocity-limit differences, though our approach focuses on real-time robot arm control applications, as opposed to the off-line methods mentioned above.

Robot Control with Human Motion — Prior work has used human motion in various robot control settings. Dragan et al.[9] explore teleoperation with customizable interfaces, where both the task context and assistive scaffolding (e.g., input goal estimation) are identified as important factors to support teleoperation and manipulation. Human pose states are one possible way to interface with the robot and serve as a starting point for inferring user intent in this framework. A large body of work in robotic surgery investigates systems that use specialized input devices to map and scale the surgeon's motions to a robot to perform minimally invasive procedures (see the work by Lanfranco et al. [19] for an overview). While our work addresses some similar problems, such as keeping the end effector path smooth and steady when performing dexterous tasks, analogous to the work by Taylor et al. [33], overall, mapping motions in free space at the scale of the user's hand space presents separate challenges. Also, work in haptics explores methods to make telemanipuations more tangible by detecting the user's motion and applying resistive forces with a haptic feedback device to simulate where objects would be in the user's workspace. We note that our method would integrate well with such devices as they become more feasible at the human-scale (See Okamura [26] for an outline on the history of such techniques).

Direct Control — Direct control involves specifying joint values in an unambiguous fashion from input to output device. For a comprehensive overview of teleoperation classifications, Niemeyer et al. [25]. While many believe direct control to be too tedious for users to complete tasks, some prior work suggests that users prefer direct control for some tasks [17]. Our approach aims to gain benefits of both direct and shared control: we want the user to feel like he/she has control over all robot movements in a direct control manner, while the system makes automatic adjustments that adapt to the environment and achieve attractive motion qualities analogous to shared control methods. Specialized input devices such as the ROBOpuppet system by Eilering et al. [10] make direct control more efficient by specifying poses on a miniature version of the robot in order to puppet the full robot. Because of the intuitive nature of this control method, we would like to compare mimicry-based control against this puppeteering approach in future work.

Retargeting Methods in Animation — Because our goal is to map motion from a human arm to a structurally dissimilar robot arm, we turn to the animation literature, where retargeting motion between different characters is a well studied problem. Because articulated characters and robots both consist of links and joints, principles can often be adapted between the two fields. Gleicher [11] first looked at retargeting motion capture data to new characters of different sizes and geometry using "spacetime constraints." We draw on some of the optimization techniques used in this line of work to map motion between kinematically dissimilar human and robot arms. A motion retargeting technique that works in real-time was proposed by Shin et al. [30], which allows character puppetry for live performances. We expand on the idea of importance-based inverse kinematics proposed in this work in order to fade motion goals in and out based on what is currently important using dynamically relaxed constraints.

3. TECHNICAL OVERVIEW

Our goal is to give the user the feeling of direct control from the user's hand to the robot, while achieving this effect with practical robot motions. A system must, therefore, map between measured configurations of the hand to configurations of the robot in real-time

[3]UR5: https://www.universal-robots.com/products/ur5-robot/

in a manner that achieves both objectives. At each instant in time, our system estimates the hand configuration, maps its estimate to a goal configuration of the robot, and executes robot movements to move toward this goal. The middle step of performing the mapping, known as *retargeting*, is the core of our approach. We divide retargeting into two steps (Figure 2). First, we map the measured position and orientation of the hand (6-DOF) from the space of the hand into a corresponding position and orientation in the robot's workspace. Second, we compute a configuration of the robot (i.e., joint angles) that achieves the position and orientation goal for its end effector.

In our current implementation, the first step (spatial mapping) is kept simple in order to preserve the direct nature of the interface. Positions are measured relative to the starting configuration so that the initial hand position corresponds to the initial robot configuration. Orientation displacements with respect to initial frames are copied directly, so that the end effector goal matches the hand orientation. A clutching mechanism disconnects positional motions of the input device, effectively re-centering the mapping. If the workspace of the robot has a significantly different scale than that of the human arm, the positional displacement is appropriately scaled.

The mapping from end effector goals to a joint configuration is commonly known as *inverse kinematics* (IK) (see Buss [3] for a review of common IK methods). Given the constraint of an end effector goal, an IK solver determines the corresponding joint angles that achieve the goal. However, telemanipulation must consider practical goals beyond simply matching the end effector configurations. A simple approach of directly mapping the 6-DOF hand configuration to a goal for the end effector at each time step fails, as motions of the user's hand often leads to a series of configurations that do not provide a valid robot motion. The differences in kinematics between the user's arm and the robot mean that a series of hand configurations from a smooth arm motion may require complex, large, and potentially discontinuous, robot motions. We confirmed this behavior empirically in experiments with a variety of IK techniques such as Jacobian Pseudoinverse [3], DLS [3], S-DLS [4], SQP [2], and IKfast. [4] In our pilot studies, the naïve approach failed in all trials: user hand motions lead to robot motions that violated velocity limits. In contrast, we have observed no instances of this failure in trials that used our approach. While some IK methods are capable of achieving secondary goals using redundant degrees of freedom, such a strategy requires a robot with extra degrees of freedom, and also makes the other goals secondary.

Our approach considers that the practical needs of creating viable motions may be more important than achieving precise end effector goals. We therefore frame the IK problem as an optimization where each goal can be appropriately weighted and provide methods for dynamically tuning the weights during movement to emphasize different objectives. *Our key insight* is that exactly matching the end effector goal is not always required to give the feeling of direct control. For example, when the user makes a fast large-scale motion, it may be more important to have the robot move in a smooth, continuous, and singularity-avoiding manner than to exactly reproduce the trajectory. The user may not notice small deviations in end effector position or orientation when moving quickly. In fact, they may be more likely to notice a lack of smoothness, fast joint deviations (that may be required to make small end effector movements), or if the robot becomes immovable due to a singularity. Therefore, our methods automatically *relaxes* the end effector goals (i.e., de-emphasizes them with respect to others), so that the robot can achieve movement-oriented goals.

[4]IKfast: http://openrave.org/docs/0.8.2/openravepy/ikfast/

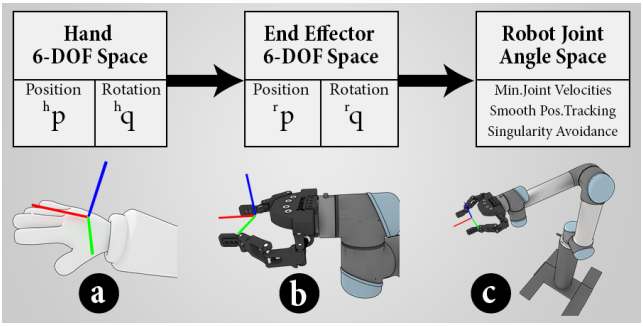

Figure 2: An overview of our method. (a) Given a goal position and orientation in the user's hand frame, (b) we map these goals to the end effector tool space of the robot, (c) and we solve for joint angles that sufficiently meet the desired end effector goals, while also achieving other attractive motion qualities.

Our inverse kinematics solver poses the computation of joint angles from a given end effector position as a constrained optimization. The hard constraints used are singularity avoidance and joint limits, while end effector position and orientation goals are treated as an objective function to be minimized. Other aspects of the robot's motion, such as smoothness in both joint and end effector space, are also used as objectives. The relative weighting of the various objectives is adjusted for each computation based on what the system determines is important at that instant.

4. RETARGETING METHOD

The core of our approach is the inverse kinematic solver that finds a goal configuration for the robot (i.e., joint angles) given information about the goal position and orientation of its end effector. At each step, our system obtains a measurement of the user's hand configuration, maps this to a position and orientation in the robot's workspace, employs the IK solver to determine the appropriate joint angles, and sends this configuration to the robot.

We denote the robot configuration as Θ, an n–vector of joint angles (where n is the number of robot joints). The robot configuration is adjusted at each system update; thus, Θ implicitly depends on a time t. We have a preset initial configuration for the robot, referred to as Θ_0. We denote the goal position for the robot's end effector by the three-vector ${}^r\mathbf{p}$ and the goal orientation for the end effector by the quaternion ${}^r\mathbf{q}$. We denote the user's hand position and orientation as ${}^h\mathbf{p}$ and ${}^h\mathbf{q}$, respectively. A subscript zero denotes an initial configuration; for example, ${}^h\mathbf{p}_0$ signifies the original position of the hand. The coordinate systems of the input device and the robot are aligned for both position and rotation.

At each step, ${}^r\mathbf{p}$ and ${}^r\mathbf{q}$ are determined based on the measured configuration of the user's hand $({}^h\mathbf{p}, {}^h\mathbf{q})$. The end effector goal position is found by matching the displacement of the user's hand with respect to its initial position: ${}^r\mathbf{p} = {}^r\mathbf{p}_0 + s({}^h\mathbf{p} - {}^h\mathbf{p}_0)$. If the workspace of the robot is of significantly different scale than that of the human arm, the positional displacement can be appropriately scaled using the term s. The goal orientation for the end effector ${}^r\mathbf{q}$ is calculated by representing the same rotation of the hand in the robot's end effector frame. From an input stream of quaternions $\mathbf{q}$, we first capture the user's initial hand orientation and save it as ${}^h\mathbf{q}_0$. We represent each new hand orientation that comes in with respect to that initial frame by taking the quaternion displacement: ${}^h\mathbf{q} = \mathbf{q}^{-1} * {}^h\mathbf{q}_0$, where $*$ denotes quaternion multiplication. Next, we interpret the user's hand rotation in the end effector frame of

the robot, seen as a part of step "b" in Figure 2. We represent the orientation of the robot using a quaternion by converting the rotation matrix of the end effector frame. We calculate the goal orientation for the end effector by rotating the hand's quaternion into the robot's initial frame through quaternion multiplication: ${}^{r}\mathbf{q} = {}^{h}\mathbf{q} * {}^{r}\mathbf{q}_0$.

We express the IK problem as a constrained optimization:

$$\begin{aligned} \Theta = \arg\min_{\Theta} \ & \mathbf{f}(\Theta) \text{ s.t. } \mathbf{c}(\Theta) \geq \mathbf{0} \\ & l_i \leq \Theta_i \leq u_i, \forall i \end{aligned} \tag{1}$$

where $\mathbf{c}(\Theta)$ is a set of constraints, l_i and u_i values define the upper and lower bounds for the robot's joints, and f is an objective function. Our challenge is to encode our goals for the mapping within the constraints and objectives. Singularity avoidance is encoded as a constraint. Other goals are encoded in the objective function.

Our objective function is the weighted sum of four terms:

$$\mathbf{f}(\Theta) = \omega_j f_j(\Theta) + \omega_e f_e(\Theta) + \omega_p f_p(\Theta) + u\omega_o f_o(\Theta) \tag{2}$$

where f_j is a joint smoothness term, f_e is an end effector smoothness term, f_p is a positional matching term, and f_o is an orientation matching term. The constants ω_i are corresponding weighting terms, and u is a scalar *importance value* that is computed to adjust the weightings between the various factors. For all experiments in this paper, we use $\omega_j = 3$, $\omega_e = 2$, $\omega_p = 10$, and $\omega_o = 5$ We discuss the various objective terms in §4.2, and the computation of u in §4.1.

4.1 Importance-Based Inverse Kinematics

Central to our objective function is the idea that we adjust the relative weights of the objective terms based on what is important at the moment. This objective is accomplished by computing the value of u in Equation 2 before the optimization begins for each step. The idea of dynamically adjusting objective weights in an IK solver based on which constraints are currently important in motion retargeting was introduced by Shin et al. [30].

The original importance-based approach determined the importance of end-effector goals based on proximity to objects; when the end-effector is close to an object, its configuration is more likely to be important because it will interact with the object. Lacking information about objects in the scene, we use a different importance measure: as the velocity of the hand increases, the precise configuration becomes less important. This measure may be seen as an approximation to the distance-based heuristic: as the user's hand approaches an object, they will naturally slow down to allow for precise manipulations of it. The velocity heuristic is also motivated by the observation that, when the hand and robot end-effector move quickly, it is less likely that orientation errors will be noticed or that precise orientation is required.

We compute the importance value based on velocity as:

$$u = \frac{v_{max} - \|\mathbf{v}_h\|}{v_{max}} \tag{3}$$

where v_{max} is a pre-calibrated maximum hand velocity per update, and $\|\mathbf{v}_h\|$ is the magnitude of the current hand velocity vector. The system described in §5 uses a v_{max} value of 0.04. As seen in Equation 2, the importance weight u only affects the orientation matching objective, not position matching.

4.2 Optimization Terms and Constraints

The following sections provide further detail on the terms used in our objective function as well as how we use the nonlinear optimization constraints and bounds to supplement our mimicry-based control approach.

Joint Velocity Minimization — Standard kinematic remapping techniques for mimicry-based control may result in motion that is too discontinuous to be used for control applications, because small positional or rotational changes in the user's hand may lead to distant configurations in the robot's joint space. Thus, we would like to favor successive solutions that provide smooth joint position changes by relaxing the constraints on end effector position and rotation goals.

To achieve smooth positional changes in real-time, we add a term in our objective function that attempts to minimize the magnitude of joint velocities:

$$f_j(\Theta) = \|\Theta'\|^2 \tag{4}$$

We approximate the joint velocity value by using backwards finite differencing.

Position Tracking — When controlling a robot with arm motion, it is important that the robot smoothly and accurately tracks the relative position of the user's hand so that the robot motion is more predictable and easier to control. Because we want users to feel like the robot is completing tasks the same way they would with their own arm, robot motion that lags behind in task space or introduces jagged end effector paths will likely break this immersive experience. However, because the robot's joint limits and velocities may differ from the user's arm, it may be difficult to smoothly track the user's relative hand path over time. We address this challenge by introducing two terms in our optimization objective function. First, we add a term that attempts to match the end effector position to the goal position ${}^{r}\mathbf{p}$. The calculation of this goal position is described at the head of §4. We take our preset initial configuration for the robot Θ_0 and calculate the original end effector position by using the forward kinematics equation of the robot, denoted here as $FK(\Theta_0)$. We formulate the following objective term to minimize the error between the goal position and current displacement of the end effector with respect to its initial position:

$$f_p(\Theta) = \|{}^{r}\mathbf{p} - FK(\Theta)\|^2. \tag{5}$$

We would also like the end effector path to be smooth in order to achieve robot motion that is more predictable and easier to track. This idea follows previous findings that smoothing in the end effector space improves user reaction times to robot arm motions [16]. Smoothing in the end effector position space has the additional benefit of filtering input signal noise in real-time, making the system sufficiently adept for dexterous adjustments. We achieve this effect in a similar fashion to Equation 4, although this calculation minimizes the end effector velocity in Cartesian space. We formulate this minimization as an optimization term as follows:

$$f_e(\Theta) = \|FK'(\Theta)\|^2. \tag{6}$$

Orientation Matching — Because the orientation of a person's hand is important in many tasks, such as pouring tea, placing a lego piece on a board, or unscrewing a light bulb, equipping the robot arm with the ability to mimic the orientation of the user's hand is critical to accomplish tasks in the same way. To achieve this goal, we must converge on the correct end effector orientation in optimization.

We denote the orientation of the robot's end effector (represented as a quaternion) as ${}^{e}\mathbf{q}$, which is computed as a function of Θ using the forward kinematics equations of the robot. We minimize the difference between this actual orientation and the goal orientation ${}^{r}\mathbf{q}$. We measure the difference between orientations as the magnitude of the rotation vector between them [21], $disp(\mathbf{q}_1, \mathbf{q}_2) = log(\mathbf{q}_1^{-1} * \mathbf{q}_2)$. The objective term is therefore:

$$f_o(\Theta) = \|disp({}^{r}\mathbf{q}, {}^{e}\mathbf{q})\|^2 \tag{7}$$

We scale the orientation term using u in the objective function as seen in Equation 3 based on our assumption that orientation is most important as users slow their motion as their hands approach an object. Thus, the importance of the end effector orientation

dynamically fades in as an object is approached. Conversely, when the user is moving his/her hand quickly, the orientation goal is relaxed so that the robot has extra degrees of freedom available to smoothly track this fast moving position goal.

Kinematic Singularity Avoidance — Kinematic singularities are a well studied robotics problem [13]. These objectionable robot poses occur when the Jacobian matrix $\mathbf{J}(\Theta)$ that maps joint velocities to end effector tool velocities, i.e., $\dot{\mathbf{x}} = \mathbf{J}(\Theta)\dot{\Theta}$, loses full rank. Under these circumstances the chain may lock since an instantaneous change in one of the end effector DOFs is unattainable. When the Jacobian matrix is near singular, small changes in the end effector tool space can induce large, diverging velocities in joint angle space, which is unsafe for many applications including direct control.

Because the user's arm physiology may differ greatly from the structure of the robot arm, and the user has no kinesthetic sense of the robot's objectionable kinematic singularity poses, the user may unknowingly guide the robot into such a singularity. To reconcile this problem, we ensure that the robot automatically maintains a safe distance from such configurations. Much previous work exists on kinematic singularity avoidance for control, especially for 7-DOF redundant mechanisms [5]. Our simplified singularity avoidance technique fits conveniently as a constraint in the optimization framework and works for robot arms of various DOFs.

We use the manipulability measure proposed by Yoshikawa [34] to define how far the robot is from a singularity. Using this distance, we specify a constraint as follows:

$$\mathbf{c}(\Theta) := \sqrt{|det(\mathbf{J}(\Theta)\,\mathbf{J}(\Theta)^T)|} - s_{min} \geq 0 \qquad (8)$$

Here, s_{min} signifies a lower bound for the manipulability measure. A good value for this bound may depend on the robot arm used, so some analysis of the mechanism's manipulability ellipse may be worthwhile. We have observed a value of 2^{-24} to suffice for our system described in §5.

Self Collision Avoidance and Stability — To avoid self collisions, we set the optimization bounds l_i and u_i for all joints in Θ_i in Equation 1 to values where self collisions are not likely, while still allowing maximum task flexibility. Effective values will depend on the robot platform, and we note that using joint limits as a method to avoid self collisions may not generalize to all robot geometries. Stability is a key concern for any teleoperation system (see Lawrence [20] for a more detailed discussion). The upper and lower bounds on Θ additionally ensure that the joint values can never diverge.

5. SYSTEM DETAILS

We have realized our approach in a system designed to provide sufficient performance, flexibility, and safety in order to demonstrate its value in a human participants study.

Input Device — For natural input, our system uses an HTC Vive controller, a wireless, handheld device that provides precise position and orientation information with low latency across a sufficiently large workspace. The controller also includes a trigger, which we use to operate the gripper, and a number of other buttons (one of which we use as a clutch). While other motion capture techniques, such as video tracking, may provide even more natural movement, the robustness and ease of setup of the Vive controller made it convenient for a user study, as it required no per-user calibration and did not fail during any of our experiments.

Robot Manipulator and Controller — For manipulation, our system uses a Universal Robots UR5 robot equipped with a Robotiq 85 gripper. These devices are commonly used in collaborative industrial settings. The UR5 is a 6-DOF arm with approximately the same workspace as a human arm but with very different kinematics. We operate the robot using position control, sending joint target commands directly to its controller via a network connection. The controller receives commands at up to 125 Hz, and our system sends goal targets as quickly as possible. Upon receiving a target, the robot moves its joints toward the goal as quickly as possible (given the safety speed limits). When an updated configuration is sent to the controller, the previous goal is over-written with the new target. The robot controller utilizes virtual safety planes and speed limits.

The Robotiq gripper involves a two-finger compliant design. To simplify operation, our system only sends open and close commands to the gripper, freeing the user from having to precisely control its configuration or force. The compliant design of the gripper affords some flexibility in grasping approach.

System Architecture — We have implemented our approach as a distributed system. Input devices are connected to a Windows PC, which runs an interface layer developed within the Unity game engine. This environment provides a flexible, low-latency interface to various input devices. The interface streams controller values over a socket connection to the retargeting engine. The retargeting engine runs on a second PC running the Linux operating system. It receives input device information, computes target configurations for the robot, and communicates these targets to the robot controller via a network connection. While the distributed setup across two computers may increase system latency, this architecture afforded flexibility in interfacing to different input devices and robot hardware. Each component of the system runs asynchronously, processing commands as quickly as possible.

Our retargeting engine can be connected to a simulator, rather than the actual robot, which allowed us to confirm the infeasibility of using the end effector goals directly with a standard IK solver.

Optimization Solver — Our retargeting engine is written in Python. To solve the non-linear constrained optimization, we use the `slsqp` solver included in the `scipy` package. The solver implements a variant of sequential quadratic programming. We provide it with derivatives of the constraint and objective functions computed with automatic differentiation via the `ad` package. Each step is executed as an independent optimization, although we use the results of the previous step as an initial condition. Our solver takes 8 ms per solution, and on a testbed of 20,000 trials, the solver converged 100% of the time. The average IK position and rotation deviations over five test motions are 4.3 mm and 0.012 radians respectively, with maximal deviations of 12.0 mm and 0.021 radians. Precise end effector control is considered more important when the user's hand slows down to afford dexterous maneuvers such as picking up or rotating objects, inducing average deviations of 0.9 mm and 0.007 radians when the user's hand velocity is below 0.1 m/s.

While local minima are a risk in our approach, we have not observed it to be a problem in practice. Because each optimization search starts at the previous solution, a local minimum will likely still be a good solution in the neighborhood of the previous configuration, meaning that the system will not be greatly disturbed.

Latency Measurement — A drawback of our architecture is that its distributed setup may lead to latency. While each component is efficient, the total system latency aggregates between parts. Rather than reporting component performance, we have measured the end-to-end system latency as a measure of the latency experienced by users. For this measurement, we recorded a high speed (240 Hz) video of the controller and the robot, starting with both at a complete stop. We marked the frames where movement in each was first visible. While this approach has some subjectivity in identifying the frames with first movement, it does include the entire system latency including the robot's inertia. We were able to reliably measure 130-140 ms.

Latency is known to degrade the effectiveness of teleoperation

systems [25]. Previous work suggests that latency does not become a factor until it becomes much worse than our system's performance (Rayman et al. [28] suggest 400 ms, while Lum et al. [23] suggest 250 ms). However, as these guidelines are based on tele-surgery applications, the effects of latency on novice users are still unknown. In future work, we will explore ways to further reduce our latency.

6. USER STUDY

In order to show that a teleoperation interface using the natural space of the human hand can facilitate novice users' control of a robot arm, we developed a set of physical tasks and conducted a human-participants study to compare user performance and experience with our system against various input devices. For input type, we selected an appropriate implementation.

6.1 Hypothesis

Our principle hypothesis is that *relaxed mimicry control* will outperform alternative, commonly used control methods on task performance and user experience for novice users due to its effectiveness, intuitiveness, and learnability.

6.2 Experimental Design, Tasks, & Procedure

To test our hypothesis, we designed a 3 × 1 within-participants experiment in which naïve participants used three control paradigms (relaxed mimicry, Geomagic Touch,[5] and the UR5 native touchscreen interface) in a random order to complete three physical tasks.

Control Methods — We compare the relaxed mimicry control to a Geomagic Touch *6-DOF stylus* and the native *touchscreen interface* of the UR5. The Geomagic Touch device is a 6-DOF stylus device commonly used for robot control, 3D sculpting, or surgery simulation. In fact, prior work suggests that highly complex operations, such as simulated surgery, can be completed with such stylus devices by trained experts [31]. However, we note that our goal is not to test expert use of this input method, but chose the 6-DOF stylus as a comparison baseline, because it is a commonly used method to achieve Cartesian control of a robot's end effector. Additionally, it allows for fluid manipulation of all 6-DOFs, thus serving as an apt comparison to test our hypothesis that the input space of the user's hand is effective and intuitive for robot control.

Users controlled the robot with the Geomagic 6-DOF stylus device by moving and rotating the stylus attached to the device base, which the robot automatically matched through its relative position and orientation. We used the same optimization-based IK solver to drive the robot using the 6-DOF stylus; however, we tuned the weights and motion scaling to tailor the solver to be compatible with this device. The weights were chosen to provide sufficient relaxation required to avoid invalid motions from reasonable inputs.

The touchscreen controller was the native on-screen interface on the UR5 teach pendant. Using this method, participants controlled the robot by holding down arrow symbols on the screen using a stylus that each corresponded to one of the six 6-DOFs of the end effector. We included this control method as a comparison baseline, because a touchscreen interface is supplied with many popular robot platforms. We do note that the most common use case for such interfaces is to pose the robot for teaching waypoints as opposed to real-time control. The three methods can be seen in Figure 3.

Tasks — To ensure the applicability and generalizability of our findings to wide range of physical manipulation tasks, we developed three tasks that followed a home-care scenario (Figure 3), as a setting that involves a novice user in an unstructured environment, both key motivations for our work. Participants were presented with

[5]Geomagic Touch: http://www.geomagic.com/en/

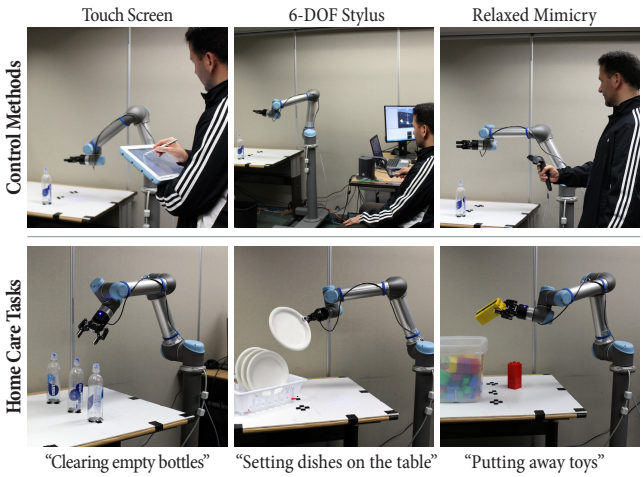

Figure 3: The three control methods (top row) and three home-care tasks (bottom row) used in our user study.

the story of logging in to the robotic teleoperation system to care for a friend or family member by clearing off a table by disposing trash or putting away unused objects and setting the table up for dinner. In the first task, which served as a simple pick-and-place task, the participant picks up three empty disposable water bottles and dropped them into a recycling bin. Our second task involved taking two plates from a dish rack and placing them on the table. Because the plates start vertically, the task tests the rotational usability of the input modalities by requiring a rotation to set the plates correctly on the table. In the third task, the participant picked up two LEGO Quatro towers and dropped them in a toy bin, as a pick-and-place task with increased complexity. Because the square pieces only fit within the fingers of the gripper if the grasp is orthogonal to the end effector facing vector, this task further constrained the orientation of the end effector compared to grasping cylindrical bottles.

Procedure — Following informed consent, participants were provided with detail on the study goals and tasks. Before using each control method, participants viewed a short training video. The videos outlined how to use the control method and specified important safety guidelines. The video-based introduction was used to standardize how participants were trained on each control method. After expressing readiness, participants were presented with a training task of picking up a single water bottle and dropping it in a recycling bin in order to get accustomed to the control method. This training task was selected due to its simplicity and the preparation it offered for the first study task. Each participant performed the training task a minimum of three times and had the option of performing it up to two additional times. If the participant task times did not decrease or become stable by the third trial, the experimenter suggested that the participant performed one or two additional trials.

After the training, participants performed tasks in the order, *three bottles*, *plates*, then *legos*, for each control method. This order ensured that task difficulty increased through the experiment. After each task, the experimenter reset the robot to the same initial configuration to standardize the starting point. After completing the training and experimental tasks using a control method, participants filled out a questionnaire pertaining to that method.

To ensure physical safety, participants stood out of the envelope of the robot at all times. Additionally, four virtual safety planes surrounding the robot were defined in the robot's low-level controller, which prompted it to halt if its end effector reached any of the planes. The training videos advised the participants to stay away

Table 1: Descriptive statistics for all measures.

	Touch Screen		6-DOF Stylus		Mimicry	
Measure	**M**	**SD**	**M**	**SD**	**M**	**SD**
Aggregate (s)	433.79	102.5	296.38	145.68	124.75	47.98
Bottles (s)	160.75	36.04	110.20	63.34	53.33	22.97
Plates (s)	133.46	45.60	68.41	50.10	27.67	13.31
Legos (s)	139.58	45.47	117.75	72.56	43.75	21.71
Ease of Use	5.63	1.41	4.51	1.23	5.67	0.95
Satisfaction	5.96	1.10	5.97	0.84	6.52	0.53
Enjoyment	4.81	1.50	5.13	1.45	5.99	1.09

Table 2: Inferential statistics for all measures. T, S, and M denote *Touch Screen*, *6-DOF Stylus*, and *Mimicry*, respectively.

					Pairwise Comparisons		
Measure	**DF**	R^2	f	p	**M-S**	**M-T**	**S-T**
Aggregate (s)	2	.99	80.4	<.0001	<.0001	<.0001	<.0001
Bottles (s)	2	.99	53.5	<.0001	<.0001	<.0001	<.0001
Plates (s)	2	.99	55.9	<.0001	.0006	<.0001	<.0001
Legos (s)	2	.99	27.7	<.0001	<.0001	<.0001	.2300
Ease of Use	2	.97	8.84	.0006	.0019	.9900	.0021
Satisfaction	2	.99	3.40	.0420	.0700	.0700	1.000
Enjoyment	2	.89	7.40	.0018	.0250	.0020	.5700

from the safety planes during the trials. If a safety plane was reached, the robot returned to its initial configuration, which required the participant to restart the task.

6.3 Measures & Analyses

To assess participants' task performance, we measured *task time* for each task and an aggregate time representing the total time it took participants to complete all tasks. Participants were given unbounded time to complete each task.

To measure participants' perception of the different control methods, we administered a questionnaire based on prior research on measuring user preferences [7, 22], including scales to measure *satisfaction*, *ease-of-use*, and *enjoyment*. Each scale included three items measured on a seven-point rating scale (1 = strongly disagree; 7 = strongly agree). Ease of use was measured using items "The control method made it easy to accomplish the task," "I felt confident controlling the robot," and "I could accurately control the robot" (Cronbach's $\alpha = 0.76$). The satisfaction scale included items "Controlling the robot was easy to understand," "I would like to control a robot like this in the future," and "I found the control method useful" (Cronbach's $\alpha = 0.74$). Finally, enjoyment was measured using items "Controlling the robot was fun," "I felt satisfied while controlling the robot," and "I felt happy while controlling the robot" (Cronbach's $\alpha = 0.75$).

We analyzed data from all measures using one-way repeated measures analyses of variance (ANOVA), including control method, *relaxed mimicry*, *6-DOF stylus*, or *touchscreen*, as a within-participants variable. All pairwise comparisons used Tukey's HSD test to control for Type I error in multiple comparisons.

6.4 Participants

We recruited 24 participants (11 male, 13 female) from the University of Wisconsin–Madison campus with ages 18–46 ($M = 22.875$, $SD = 6.05$). Participants reported low-to-moderate familiarity with robots ($M = 2.92$, $SD = 1.61$, measured on a seven-point scale). Eleven participants reported participating in prior robotics research studies. The study took 60 minutes, and each participant received $10 for their time.

6.5 Results

We outline the results from the analyses of our data, comparing our relaxed-mimicry-based control method to baseline control paradigms over numerous tasks. To facilitate readability, a textual description of the results are provided below, and all statistical details are given in Tables 1–2 and Figures 4–5.

Objective Measures — The analysis showed full support for our hypothesis with respect to objective measures (Figure 4). Relaxed mimicry control significantly outperformed both the geomagic touch and touch screen interface control methods on all tasks for all pairwise comparisons. Controlling a robot using the natural arm space afforded improved task performance on all tasks. On average, the aggregate task was completed three minutes faster than the 6-DOF stylus device and five minutes faster than the touch screen.

As illustrated in the far right graph in Figure 4, learning times from the first three trials of the training task show much steeper learning curves for the 6-DOF stylus and touch screen, suggesting that more learning was required to reach competency with these input methods. Even at the end of training, participants do not get as proficient as they do immediately with our mimicry-based method.

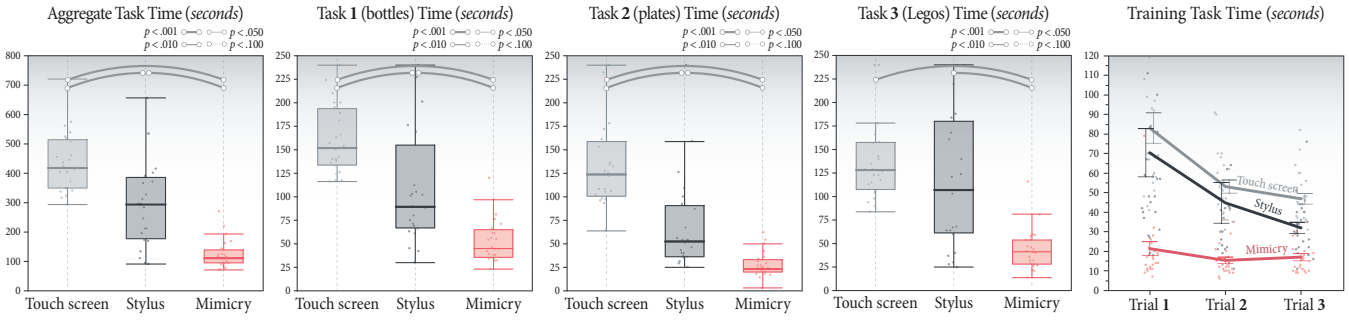

Figure 4: Tukey boxplots of data from the performance measures for each control method across tasks. Far left: aggregate task time derived by adding times from the bottles, plates, and legos tasks. Far right: learning curves for each control method. The extents of the box represent the the first and third quartiles. The line inside the box represents the second quartile (the median). The difference between the first and third quartiles is the interquartile range (IQR). Arc thickness denotes the significance level at $p < .001$, $< .010$, $< .050$, and $p < .100$ in comparisons.

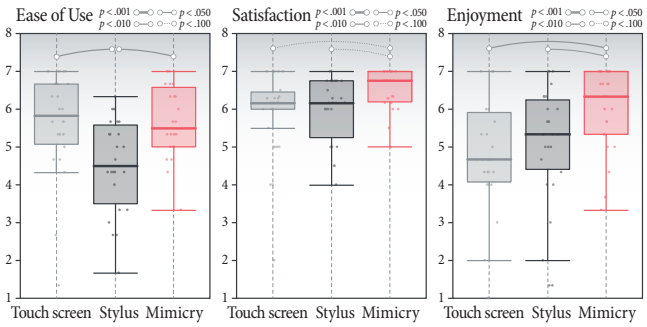

Figure 5: Tukey boxplots of data from subjective measures of *ease of use*, *satisfaction*, and *enjoyment*.

Subjective Measures — Our hypothesis was partially supported by data from subjective measures (Figure 5). Participants found the mimicry-based method to be significantly easier to use than the 6-DOF stylus device, but the touch screen was found to be equally easy to use. The mimicry method was marginally more satisfactory over the 6-DOF stylus device and touch screen. Finally, data from the enjoyment measure fully supported our hypothesis, participants finding the mimicry-based method to be more enjoyable than the 6-DOF stylus and touch screen.

7. DISCUSSION

In this paper, we proposed a method for direct tele-control of a robot arm with natural arm movements as an effective telemanipulation interface for novice users. We have addressed the key technical challenge of creating an appropriate mapping from the user's arm motions to the robot that provides users with a feeling of direct control while producing acceptable robot motions by relaxing optimization end-effector goals. We have implemented the method into a system that uses a commodity 6-DOF input device to control a robot arm and conducted a user study that demonstrates the effectiveness of the system for novice users.

The results from our user study support the premise that controlling a robot using the natural free space of the arm as an input space has usability advantages for novice users. Our results show the effectiveness of the relaxed mimicry-based control over alternative, commonly used control methods, including touch-screen and 6-DOF stylus interfaces, across three different tasks. The results are consistent across objective and subjective measures. The mimicry-based control method was found to be more enjoyable and marginally more satisfactory than these alternative methods. Finally, participants reported the mimicry-based method to be easier to use than the 6-DOF stylus controller but not the touch-screen interface, despite their poor objective performance using the touch-screen. We attribute this finding to the overwhelming familiarity participants have with touch-screen interfaces from their day-to-day use of computers and mobile devices.

Limitations — While the results from our study show that mimicry-based control is a feasible and promising method to provide a direct interface for telemanipulation, the limitations of the present implementation and testing suggest many extensions. Although our present implementation enables a sufficient demonstration of the effectiveness of the method, we have not determined the relative contribution to performance caused by each method component. While our initial simulation-based testing indicated that *some* method was necessary to address practical concerns of movement quality, such as avoiding unwanted rapid movements of the robot or getting stuck in singularities, we have not established whether or not the methods we employed offer the best solutions.

Another limitation of the our work is that the current implementation considers only the 6-DOF rigid configuration of the hand. It does not provide for control over the rest of the arm, or fine control over the robot hand. Such additional control may be important to avoid collisions in more cluttered environments and specify more dexterous manipulations. In the future, we will extend our approach with sensing of the user's arm and hand as well as integrating some degree of shared control for collision avoidance and grasp planning. Our current work considers simple environments; extending to complex and cluttered environments offers a number of challenges beyond collision and manipulation, such as operator visibility.

Our current implementation requires the user to hold a 6-DOF controller and use a button to open and close the gripper. While this input device is reliable and accurate, it may not be as natural as directly measuring the hand using, e.g., motion capture or computer vision technology. In future work, we would like to understand whether these alternative devices provide naturalness that affords an even more effective interface and whether the practical issues in using direct measurement (e.g., noise and accuracy) can be addressed. More generally, while our relaxed mimicry approach may be applied to almost any input device, it will require adaptation and experimentation to understand the user control scenarios in which it can, and should, be applied. Similarly, our present experiments do not consider haptic or tactile feedback. Integrating such feedback into our approach may provide the advantages of both.

Our focus on novice users may over-emphasize the initial learning time. While Figure 4 (far right) suggests that users may learn other methods quickly, the advantages of mimicry persist on the final task of the experiment. Further experimentation would be required to determine at what point novice performance converges to that of an expert and what designs might be most effective for experts.

8. CONCLUSION

In this paper, we present a mimicry-based telemanipulation interface that maps the movements of the user's hand in its natural space to control a robot arm. The premise of the interface is that allowing users to direct a robot using the natural space of his/her own hand would afford intuitive control for novice users. To realize the interface, we developed an IK solver that addresses the issues in mapping human arm motions to robot control. Our solver makes mimicry control feasible in practice by relaxing the mapping from the user's hand to the robot end effector, allowing it to achieve motion goals such as minimized joint velocities, smooth positional tracking, and kinematic singularity avoidance. We created a prototype system that instantiated our mimicry method that was effective and safe enough to test the premise of our proposed interface as a hypothesis in a user study. Results showed the promise of our method by showing significant control performance advantages as well as perceived ease of use and enjoyability benefits over common control paradigms. We will continue to improve our mimicry-based control method in future work and will compare it against more control alternatives.

9. ACKNOWLEDGEMENTS

This research was supported by National Science Foundation award 1208632. We would like to thank Christopher Bodden for contributing to early iterations of this work, as well as Alper Sarikaya, Tomislav Pejsa, Sean Andrist, and Catherine Steffel for valuable feedback.

References

[1] D. Aarno, S. Ekvall, and D. Kragic. Adaptive virtual fixtures for machine-assisted teleoperation tasks. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1139–1144. IEEE, 2005.

[2] P. Beeson and B. Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 928–935. IEEE, 2015.

[3] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.

[4] S. R. Buss and J.-S. Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics, GPU, and Game Tools*, 10(3):37–49, 2005.

[5] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[6] J. W. Crandall and M. A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1290–1295. IEEE, 2002.

[7] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.*, 13(3):319–340, 1989.

[8] A. D. Dragan and S. S. Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

[9] A. D. Dragan, S. S. Srinivasa, and K. C. Lee. Teleoperation with intelligent and customizable interfaces. *Journal of Human-Robot Interaction*, 2(2):33–57, 2013.

[10] A. Eilering, G. Franchi, and K. Hauser. Robopuppet: Low-cost, 3d printed miniatures for teleoperating full-size robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 1248–1254. IEEE, 2014.

[11] M. Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42. ACM, 1998.

[12] R. C. Goertz, R. A. Olsen, and W. M. Thompson. Electronic master slave manipulator, Aug. 5 1958. US Patent 2,846,084.

[13] C. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation*, 6(3):281–290, 1990.

[14] K. Hauser. Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Autonomous Robots*, 35(4):241–254, 2013.

[15] P. F. Hokayem and M. W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, 2006.

[16] M. Huber, C. Lenz, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. Human preferences in industrial human-robot interactions. In *Proceedings of the International Workshop on Cognition for Technical Systems*, pages 4749–4754, 2008.

[17] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal. How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(1):2–14, 2012.

[18] D. Labonte, P. Boissy, and F. Michaud. Comparative analysis of 3-d robot teleoperation interfaces with novice users. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(5):1331–1342, 2010.

[19] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers. Robotic surgery: a current perspective. *Annals of surgery*, 239(1):14–21, 2004.

[20] D. A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, 1993.

[21] J. Lee. Representing rotations and orientations in geometric computing. *IEEE Computer Graphics and Applications*, 28(2):75–83, 2008.

[22] A. Lin, S. Gregor, and M. Ewing. Developing a scale to measure the enjoyment of web experiences. *Journal of Interactive Marketing*, 22(4):40–57, 2008.

[23] M. J. Lum, J. Rosen, H. King, D. C. Friedman, T. S. Lendvay, A. S. Wright, M. N. Sinanan, and B. Hannaford. Teleoperation in surgical robotics–network latency effects on surgical performance. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6860–6863. IEEE, 2009.

[24] K. Muelling, A. Venkatraman, J.-S. Valois, J. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell. Autonomy infused teleoperation with application to BCI manipulation. *arXiv preprint arXiv:1503.05451*, 2015.

[25] G. Niemeyer, C. Preusche, and G. Hirzinger. Telerobotics. In *Springer Handbook of Robotics*, pages 741–757. Springer, 2008.

[26] A. M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[27] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson. Adapting human motion for the control of a humanoid robot. In *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02.*, volume 2, pages 1390–1397. IEEE, 2002.

[28] R. Rayman, S. Primak, R. Patel, M. Moallem, R. Morady, M. Tavakoli, V. Subotic, N. Galbraith, A. van Wynsberghe, and K. Croome. Effects of latency on telesurgery: an experimental study. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 57–64. Springer, 2005.

[29] L. B. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Virtual Reality Annual International Symposium, 1993 IEEE*, pages 76–82. IEEE, 1993.

[30] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Transactions on Graphics (TOG)*, 20(2):67–94, 2001.

[31] M. S. Sorensen, J. Mosegaard, and P. Trier. The visible ear simulator: a public pc application for gpu-accelerated haptic 3d simulation of ear surgery based on the visible ear data. *Otology & Neurotology*, 30(4):484–487, 2009.

[32] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin. On human motion imitation by humanoid robot. In *International Conference on Robotics and Automation, 2008. ICRA 2008. IEEE*, pages 2697–2704. IEEE, 2008.

[33] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. Dejuan, and L. Kavoussi. A steady-hand robotic system for microsurgical augmentation. *The International Journal of Robotics Research*, 18(12):1201–1210, 1999.

[34] T. Yoshikawa. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.

[35] E. You and K. Hauser. Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input. In *Robotics: Science and Systems*, volume 7, page 354, 2012.

[36] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete. Telemanipulation assistance based on motion intention recognition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1121–1126. IEEE, 2005.