

An Avatar Robot Overlaid with the 3D Human Model of a Remote Operator

Ravi Tejwani¹, Chengyuan Ma¹, Paolo Bonato² and H. Harry Asada³, *Fellow, IEEE*

Abstract—Although telepresence assistive robots have made significant progress, they still lack the sense of realism and physical presence of the remote operator. This results in a lack of trust and adoption of such robots. In this paper, we introduce an Avatar Robot System which is a mixed real/virtual robotic system that physically interacts with a person in proximity of the robot. The robot structure is overlaid with the 3D model of the remote caregiver and visualized through Augmented Reality (AR). In this way, the person receives haptic feedback as the robot touches him/her. We further present an Optimal Non-Iterative Alignment solver that solves for the optimally aligned pose of 3D Human model to the robot (shoulder to the wrist non-iteratively). The proposed alignment solver is stateless, achieves optimal alignment and faster than the baseline solvers (demonstrated in our evaluations). We also propose an evaluation framework that quantifies the alignment quality of the solvers through multifaceted metrics. We show that our solver can consistently produce poses with similar or superior alignments as IK-based baselines without their potential drawbacks.

I. INTRODUCTION

In a situation when a person (such as an older adult) needs physical assistance, a care giver would be required to provide that support. In remote interactions, physical assistance should be delivered without degrading the quality of service. However, providing quality remote service is more challenging than in-person care. This could be realized by using a physical agent, i.e. a robot, being operated from a remote site. However, it might be uncomfortable or even terrifying to have such robot touch a care recipient's body. Rather, it is desirable that a remote caregiver appears to be in contact with the recipient while the robot is providing the physical assistance.

Advancements in telepresence assistive robots have enabled care givers to remotely monitor and assist patients during the rehabilitation process by observing and communicating with patients in real-time, even if they are not physically present in the same location. These robots have been used in a variety of rehabilitation services, including exercises, stretches, and massages [1]–[3]. They have also helped patients to improve their mobility and balance by providing feedback on their movements and posture [4]–[6].

¹Ravi Tejwani and Chengyuan Ma are with Department of Electrical Engineering and Computer Science (EECS), Massachusetts Institute of Technology, Cambridge, MA 02142 USA {tejwanir, macy404}@mit.edu

²Paolo Bonato is with Harvard Medical School Department of Physical Medicine and Rehabilitation at Spaulding Rehabilitation Hospital, Charlestown, MA 02129 USA pbonato@mgh.harvard.edu

³H. Harry Asada is with Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02142 USA asada@mit.edu

The research on visual representation of the remote therapist on to the robot has been limited in their ability to display the remote therapist in the form of 2D screens [1], [2], [7] or using virtual and augmented reality to display the remote person [8]–[10]. Even the state of the art telepresence avatar robots [11]–[13] simply project the avatar (3D human model) of the remote therapist on to the screens. Hence, there is a fundamental disconnect in the research literature between the visual display of the remote care giver on the tele-operated robot and the actual remote physical therapy performed by the robot.

In this paper, we aim to bridge this gap by conceptualizing the development of an Avatar Robot as a mixed real/virtual assistive tele-operated robot – a physical robot that is overlaid with the 3D Human model of a remote care giver. Therefore, the care receiver might feel as though he is in contact with the remote operator.

Our contributions include:

- 1) *Avatar Robot System*: To the best of our knowledge, we propose the first general purpose end-to-end robotic system that provides the integration of the trajectory control of the robot, cross-device pose synchronization, robot tracking, alignment solver(s), and online 3D projection overlay using an Augmented Reality (AR) device; it achieves efficient and reactive real-time pose synchronization, alignment computation, and overlay projection.
- 2) *Alignment solver*: An optimal non-iterative alignment solver that solves for the optimally aligned pose of 3D Human model to the robot (shoulder to the wrist). We show how existing iterative IK solvers such as Jacobian DLS [14] and FABRIK [15] can be adapted for this purpose. Compared to these baselines, our solver is intuitive, concise, deterministic, singularity-free, and superior in alignment metrics. We also provide theoretical guarantees of the solver;
- 3) *Empirical metrics and evaluation*: Set of evaluation metrics that qualitatively measures the quality of human-robot alignment from multiple perspectives including visual overlap, structural deviation, and model distortion. Our evaluation metrics could serve as a benchmark for evaluating future telepresence robots and their performance.

II. AVATAR ROBOT SYSTEM

An Avatar Robot is a mixed real/virtual robotic system that physically interacts with a person in proximity of the robot. As shown in Fig. 1, a remotely operated robot and

the 3D model of a remote operator are incorporated into our system. The robot structure is overlaid with the 3D model of the remote caregiver using our solvers and visualized through Augmented Reality (AR). In this way, the person receives haptic feedback as the robot is in contact with him/her. Simultaneously, the person would visually observe the caregiver’s presence through 3D rendering. Thus, care receivers will have a highly immersive experience when haptic stimuli and visual observations are synchronized.

Problem Given a 3D model of a human, obtained from the 3D scan of the physical therapist (Fig. 1a), and a dual arm collaborative YuMi robot (Fig. 1b) which can provide physical assistance to the person. Our objective is to optimally align and overlay them across their joints, axes and rotations as the Avatar Robot interacts with the person.(Fig. 1c).

Human Model We prepare the 3D human model from a photogrammetry scan of a Physical Therapist from Spaulding Rehabilitation Center, Charlestown, MA. The therapist was in T-pose at the time of scan, and we manually annotated the joints of the model so the model can be posed dynamically. In this research, we are interested in dynamically computing the pose of the model’s arms to create optimal alignment with the robot arms, so joints of particular concern are the shoulder, elbow, and wrist joints on both sides.

Structural Correspondence between the Human Model and the Robot Since our robot has anthropomorphic arms, with clearly identifiable “upper arm” and “forearm” segments, our task is to align the human model’s shoulder, elbow, and wrist joints to the corresponding *reference positions* on the robot, which we choose to be the center of axis 2, 3, 6 of the robot respectively. A side-to-side comparison between the arms of our human model and the robot as well as the reference positions that we chose is shown in Fig. 2.

As a preliminary step of alignment, we also compute the position and rotation of the entire human model such that the segment connecting its shoulder joints overlaps and share the same midpoint with the segment connecting the shoulder reference positions on the robot. This step ensures the body /shoulder of the human model is aligned with the body of the robot at large and allows us to compute the best-aligning pose of two arms separately.

Defining Input and Output for the Solvers The input to an alignment solver consists of an joint positions of the human model in its default (unaligned) configuration and the reference joint positions of the robot. The output will be an *arm pose* for each side, which include the rotations of joints as well as stretch factor of forearm and upper arm of the human model. Allowing stretching parts of the human model risks making the output pose un-anthropomorphic, but it is necessary for better alignment as the proportions of robot and human arms are inevitably different (see Fig. 2). We assume that some extent of scaling is acceptable. We will revisit incorporating scaling and other constraints in the description of the solvers.

Pipeline The Avatar Robot System integrates the alignment solvers into an end-to-end real-time workflow that

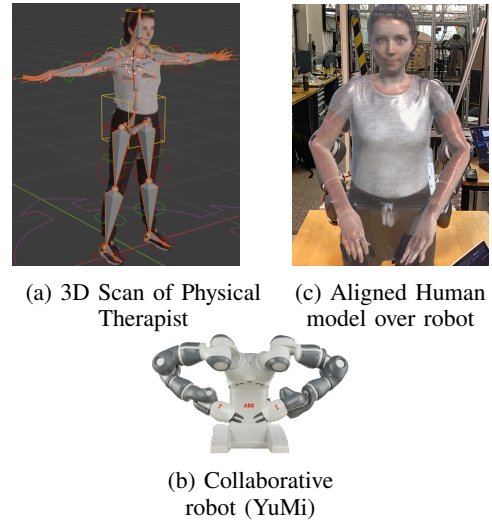


Fig. 1: Aligning Human model with Collaborative robot

comprises trajectory control of the robot, cross-device pose synchronization, robot tracking, alignment solving, and overlay projection. The pipeline of robot System is presented in Fig. 3. It consists of three components: the robot, the control station, and the augmented reality (AR) device.

The robot, a dual-arm collaborative robot [16], runs a pose server and a generic motion server in parallel on each arm, and communicates with the control station through Ethernet. The control station stores control scripts that issues specific trajectory commands to the robot to perform various tasks, e.g., emulating a rehabilitation therapy. It also collects joint angles and gripper orientations from both arms at real-time and hosts a relay server to stream this information through the local wireless network. We use HoloLens 2 [17] as the AR device. The pose of the robot base is tracked using the camera mounted on the AR device with a computer vision model. The raw pose estimation is further passed to a low-

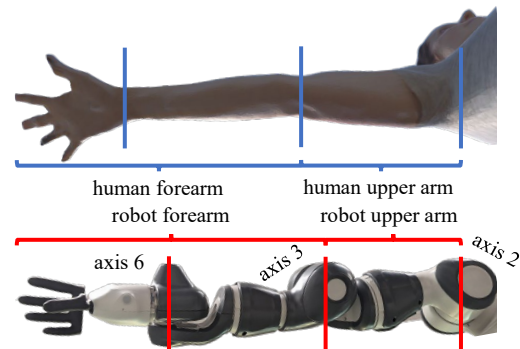


Fig. 2: Side-by-side comparison of our robot arm and the arm of our human model. We set the center of axis 2, 3, and 6 of our robot as the shoulder, elbow, and wrist reference points for the human model. The difference in proportions of arm segments between the human model and the robot makes scaling necessary in order to achieve good alignment.

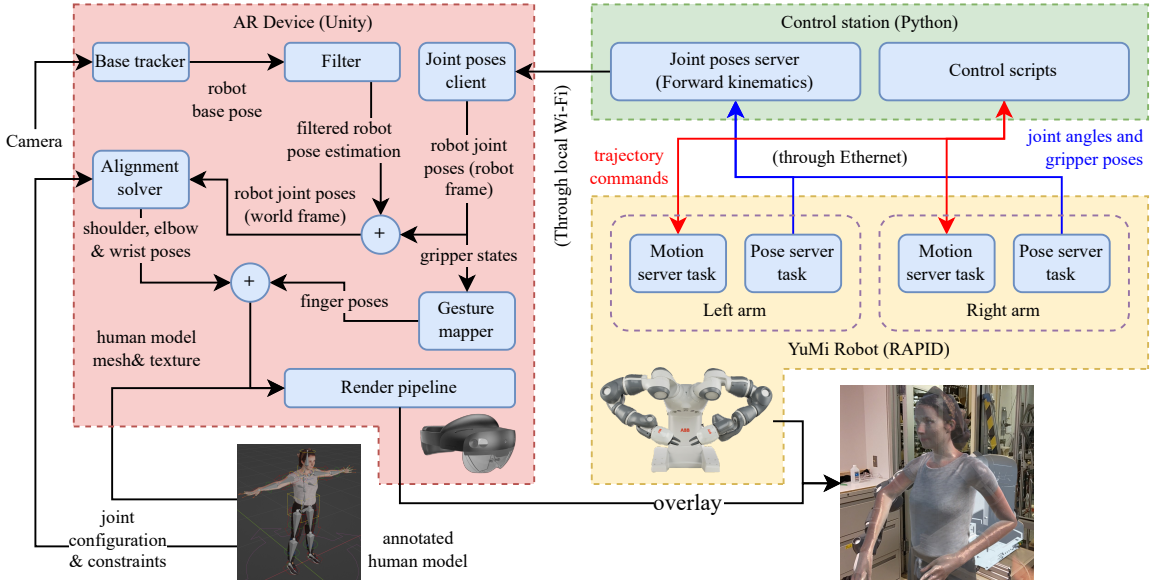


Fig. 3: The pipeline of our Avatar Robot System. A motion server and a pose server run in parallel on both arms of the robot. The motion server executes commands from scripts on the control station. The control station also runs a relay that aggregates joint angles from both arms and stream dual-arm joint poses to the AR device over WLAN. The AR device converts local-frame joint positions to world-frame by them combining with the pose estimate of the robot from its cameras. An alignment solver then solves for the human model pose; A gesture mapper maps gripper positions to hand gestures of the human model. With the arm poses and gesture, the renderer renders the human model accurately over the robot. Joint poses are synchronized between robot, AR device and the control station at 100Hz, while the rendering is at 60Hz.

pass filter to reduce noises and stuttering. The filtered pose is combined with the robot-frame joint coordinates retrieved from the control station real-time to compute the world-frame joint positions. Our system then computes the shoulder, elbow, and wrist reference positions for the human model and aligns its body with the robot. The reference positions are then passed to an alignment solver, whose pose output is then fed to the rendering pipeline which renders the human model and projects it over the robot in front of the patient.

For completeness, we also designed a gripper mapper module that maps the state of robot gripper into the gesture of the model. Our gripper is based on a 1DoF linear gripper with fingers customized to resemble thumb and fingers, so we map the position of the gripper linearly to the angle between the thumb and index finger of the model. In practice, this system can align a human model pose with a robot in real-time with minimal latency and high accuracy.

III. SOLVERS

In this section, we first explore how existing iterative Inverse Kinematics (IK) solvers can be adapted to perform the aforementioned alignment task. Specifically, we show adaptation of two IK solvers: Jacobian Damped Least Square [14], commonly used in robotics, and FABRIK [15], commonly used in humanoid animation. These solvers serve as our baselines. We then introduce our optimal non-iterative alignment (ONIA) solver.

A. Inverse Kinematic Solvers

1) *Jacobian (adapted)*: We first adapted the Jacobian method with damped least square (DLS) for the alignment task. Given a parameterization θ and the current end effector position \mathbf{p} , the effect of perturbing θ on \mathbf{p} can be linearly approximated by $\Delta\mathbf{p} = \mathbf{J}\Delta\theta$ in \mathbf{p} 's neighborhood, where \mathbf{J} is the Jacobian. Provided with a desired $\Delta\mathbf{p}$, Jacobian with DLS computes $\Delta\theta$ as follows,

$$\Delta\theta = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I})^{-1}\Delta\mathbf{p}. \quad (1)$$

where the hyper-parameter λ is the damping constant that must be chosen with care to keep the solver both responsive and stable.

We model the human arm as the shown in Fig. 4. In particular, we parameterize the swing of the upper arm around the shoulder as an exponential map [18], [19]. To incorporate the reference position of the elbow in Jacobian-based algorithms, we start by noting that the Jacobian method is equivalent to gradient descent to the objective

$$\frac{1}{2}\|\mathbf{p} - \mathbf{x}\|^2 \quad (2)$$

where \mathbf{x} is the target position. For a joint, \mathbf{p}_e , we want to be close to its reference, \mathbf{x}_e , we add the term $\frac{1}{2}w_e^2\|\mathbf{p}_e - \mathbf{x}_e\|^2$ to the objective,

$$\frac{1}{2}\|\mathbf{p} - \mathbf{x}\|^2 + \frac{1}{2}w_e^2\|\mathbf{p}_e - \mathbf{x}_e\|^2 = \frac{1}{2}\left\|\begin{bmatrix} \mathbf{p} \\ w_e\mathbf{p}_e \end{bmatrix} - \begin{bmatrix} \mathbf{x} \\ w_e\mathbf{x}_e \end{bmatrix}\right\|^2 \quad (3)$$

where w_e is a small constant so $\frac{1}{2}w_e^2\|p_e - x_e\|^2$ becomes a secondary objective. We find $w_e = 0.1, \lambda = 0.2$ to work best.

2) *FABRIK (adapted)*: Forward And Backward Reaching Inverse Kinematics (FABRIK) is a heuristic-based IK solver that received wide use in humanoid animation due to its simplicity and its natural poses [15]. Every iteration of FABRIK consists of a forward and backward reach that traverse the kinematic chain in different directions. It repositions a joint on the line connecting its current position and the preceding joint's updated position.

Our adaptation presented here enhances plain FABRIK by allowing every joints to have a reference positions. It produces a pose that minimizes the maximum deviation of any joint to its reference¹. Under our adaptation, a joint with a reference position is given a deviation tolerance ϵ . In every iteration, the joint is re-positioned not on the segment connecting its child and itself, but on the segment connecting the child and the point closest to the joint *within a radius- ϵ sphere around the reference position*. This heuristic is intuitive, local, minimal, easy to implement, and works well empirically. The choice of ϵ is critical: if too large, the constraint is too loose to be useful; if too small, the solver may be unstable due to infeasibility. We determine the optimal ϵ by binary search.

Both Jacobian and FABRIK allow incorporating joint rotational constraints into the solution. Hence, we explored adding basic human joint constraints into both baselines: We allow the upper arm to swing for at most 85° in all directions and twist for $\pm 75^\circ$ relative to the T-pose. The hinge joint permits rotation from 0 to 150° inward, with the T-pose as the zero reference. These constraints are more permissive than anatomical ground truth [18] so as not to over-constrain our model, as alignment is also a tight constraint on its own. We allow a stretch factor from 0.8 to 1.3 for both upper and forearm for reasons discussed in section II.

B. Optimal Non-Iterative Alignment Solver

It is helpful when the input robot arm pose is not anthropomorphic to use IK solvers that are generic and consider joint constraints. However, when the robot arm poses are anthropomorphic, the benefits of these baseline IK solvers fails to outweigh their limitations: they are prone to singularities, and require careful tuning of its hyper parameters to be stable. We introduce an Optimal, Non-Iterative, Alignment solver (ONIA) that solves for the optimally aligned pose from shoulder to the wrist non-iteratively. ONIA is stateless, achieves optimal alignment and faster than the baseline solvers (demonstrated in the evaluation section). We describe ONIA's design below.

Nomenclature: We begin by defining our notations. Let \mathcal{R}_x^y denote the rotation that rotates x to y around $x \times y$. In other

¹We make the remark that a joint with a reference is different from an end effector with a target, because an end effector is assumed to be always able to reach its target in FABRIK but it may very well be infeasible for every joint to be at its reference position, and FABRIK is unstable when confronted with infeasibility. Hence, plain FABRIK's ability to handle multiple end effector does not help in our scenario.

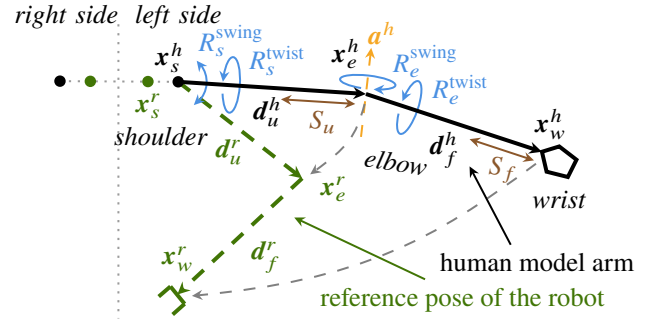


Fig. 4: The kinematics model. We model the shoulder joint of a human model as a ball-and-socket joint which can be decomposed into a swing and a twist [18]. The elbow joint is composed of a hinge followed by a small of twist around the forearm. The shoulder of the human model and the robot were assumed to be aligned initially (see section II). The notations are used by the description of our non-iterative solver below.

words, \mathcal{R}_x^y is the rotation such that

$$\mathcal{R}_x^y x = y, \quad \mathcal{R}_x^y (x \times y) = x \times y. \quad (4)$$

In addition, let

$$(x)_{\perp y} = x - \frac{x \cdot y}{\|y\|^2} y \quad (5)$$

denote the projection of x onto the plane normal to y .

As outlined in section II. The input to ONIA has two parts: the joint positions of an arm of the human model (in its default unaligned pose), and the corresponding reference positions of the robot. For clarity, through out this section, we'll use the superscripts h to denote joints from the human model and r to denote references positions of the robot. We will also use subscripts s, e, w, u, f for houlder, elbow, wrist, upper arm, and forearm. All joints and rotations below are of the human model unless explicitly specified otherwise. Fig. 4 visualizes the symbols we used below and presents the kinematic chain of the human model which ONIA works on.

ONIA aligns the given arm of the human model to an arm of the robot from shoulder to wrist through the following steps:

a) *Computing arm axes*: ONIA begins by computing the *arm axes* of both the human model and the robot, which are vector representations of the upper and forearm segments. Let x_s^h, x_e^h and x_w^h be the positions of the human model's shoulder joint, and let x_e^r and x_w^r be the reference elbow and wrist positions of the robot. The elbow axes are computed as

$$d_u^h = x_e^h - x_s^h, \quad d_f^h = x_w^h - x_e^h, \quad (6)$$

$$d_u^r = x_e^r - x_s^r, \quad d_f^r = x_w^r - x_e^r. \quad (7)$$

Note d_u^r starts at x_s^h instead of x_s^r (see Fig. 4). Because the human model's shoulder width differs from the robot's shoulder width. The human model's and robot's shoulder joints cannot be perfectly aligned at the same time. Hence, we use x_s^h as the basis for guiding rotation of the upper arm.

b) *Aligning direction of the upper arm:* The direction of the upper arm is aligned with the reference axis \mathbf{d}_u^r by swinging it around the shoulder,

$$R_s^{\text{swing}} = \mathcal{R}_{\mathbf{d}_u^h}^{\mathbf{d}_u^r}. \quad (8)$$

c) *Aligning scale of the upper arm:* We further stretch(shrink) the upper arm segment so that its elbow joint reaches the reference position \mathbf{x}_e^r . This is done by computing the scaling factor S_u ,

$$S_u = \frac{\|\mathbf{d}_u^r\|}{\|\mathbf{d}_u^h\|}, \quad (9)$$

such that

$$\mathbf{d}_u^r = R_s^{\text{swing}}(S_u \mathbf{d}_u^h). \quad (10)$$

d) *Aligning the elbow axis:* Further we twist the the upper arm so that its *elbow axis* aligns with the *reference elbow axis* of the robot. The elbow axis of a human model's arm is the axis around which the elbow rotates, if treated as a hinge joint. We annotate the elbow axis when preprocessing the model and denote it as \mathbf{a}^h . The reference elbow axis of the robot is the cross product of its reference arm axes,

$$\mathbf{a}^r = \mathbf{d}_u^r \times \mathbf{d}_f^r. \quad (11)$$

Consequently, the twist component of the human arm's shoulder joint is

$$R_s^{\text{twist}} = \mathcal{R}_{\mathbf{a}_\perp^h}^{\mathbf{a}^r}, \quad (12)$$

$$\text{where } \mathbf{a}_\perp^h = \left(R_s^{\text{swing}} \mathbf{a}^h \right)_\perp \mathbf{d}_u^r. \quad (13)$$

Essentially, we rotate the swung upper arm (and hence its elbow axis) around itself so the projection of the elbow axis onto the plane normal to the axis of rotation, \mathbf{d}_u^r , coincides with the reference elbow axis \mathbf{a}^r . This minimizes the angle between the human model's elbow axis and the reference elbow axis: in general, given vectors $\mathbf{u}, \mathbf{v}, \mathbf{k} \in \mathbb{R}^3$, where \mathbf{v} can rotate around \mathbf{k} . The angle between \mathbf{v} and \mathbf{u} is minimized when $(\mathbf{v})_\perp \mathbf{k}$ and $(\mathbf{u})_\perp \mathbf{k}$ meet.

The overall rotation of the shoulder joint is the composition of the swing followed by the twist,

$$R_s = R_s^{\text{twist}} R_s^{\text{swing}}. \quad (14)$$

We are now done with the shoulder joint and the upper arm and now move on to the elbow joint and the forearm.

e) *Aligning the direction of the forearm:* Similar to (8), we align the forearm — now $R_s \mathbf{d}_f^h$ after rotating the shoulder — by swinging it around the elbow so that it lines up with the reference forearm axis,

$$R_e^{\text{swing}} = \mathcal{R}_{R_s \mathbf{d}_f^h}^{\mathbf{d}_f^r} R_s. \quad (15)$$

f) *Aligning the scale of the forearm:* We further stretch (shrink) the forearm, similar to (9), so that wrist joint of the human model reaches the reference wrist position. The scaling factor of the human model's forearm is computed as

$$S_f = \frac{\|\mathbf{d}_f^r\|}{\|\mathbf{d}_f^h\|}. \quad (16)$$

g) *Aligning the wrist:* We set the rotation of the wrist R_w to be the same as the end effector rotation of the robot.

h) *Reacting to wrist twist:* Finally, we twist the forearm slightly around itself proportional to the twist of the wrist joint. This leads to more realistic rendering by enabling the forearm skin near the elbow to move in response to the wrist twist. We develop the notion of *wrist twist* through the following insight: Suppose in the default configuration of the human model, there is some “wrist vector” \mathbf{w} that is roughly perpendicular to the palm. Then, after the wrist joint rotates by R_w , the twist component of R_w is the rotation around the forearm axis that minimizes the angle between \mathbf{w} and $R_w \mathbf{w}$. Similar to (12), this is the rotation that brings the projection of the two vectors together. With this insight, we formulate the proportional twist of the elbow joint as follows,

$$R_e^{\text{twist}} = \left(\mathcal{R}_{\mathbf{w}_\perp^h}^{\mathbf{w}_\perp^r} \right)^{\alpha_e} \quad (17)$$

$$\text{where } \mathbf{w}_\perp^h = \left(R_e^{\text{swing}} \mathbf{w} \right)_\perp \mathbf{d}_f^r, \mathbf{w}_\perp^r = (R_w \mathbf{w})_\perp \mathbf{d}_f^r. \quad (18)$$

We set $\alpha_e = 0.4$ empirically. Together, the overall rotation of the elbow joint is

$$R_e = R_e^{\text{twist}} R_e^{\text{swing}}, \quad (19)$$

This concludes ONIA. Note that in the above procedure, ONIA starts with the default configuration of the human arm and is not dependent on the current poses. Being non-iterative makes the algorithm easy to analyze and ensures a one-to-one, deterministic mapping from the robot poses to the human model poses.

ONIA produces optimal alignment in two regards. First, the swinging ((8) and (15)) and scaling ((9) and (16)) ensures that the elbow and wrist joints of the human model reaches their reference positions on the robot *exactly*. In addition, the twist in (12) *minimizes* the angle between the human model's elbow axis and the robot's elbow axis.

IV. EVALUATION

A. Metrics

We define the following metrics in order to evaluate the performance of the solvers in our Avatar Robot System:

1) **Overlay ratio** of a solver at a certain frame is defined as

$$O := \frac{\text{area of robot arms overlaid by human model}}{\text{total area of robot arms}}, \quad (20)$$

where “area” is computed from the patient's perspective, i.e. the area of projection into patient's eye view. This is an intuitive metric of visual alignment: a value of 1 would place all robot arms behind the human model and only the human model would be visible. We compute the overlay ratio by replaying sessions in simulations with a special shader (see Fig. 7).

2) **Elbow (wrist) deviation:** The elbow deviation Δx_e and the wrist deviation Δx_w are defined as the distances of the human model's elbow and wrist joints from their

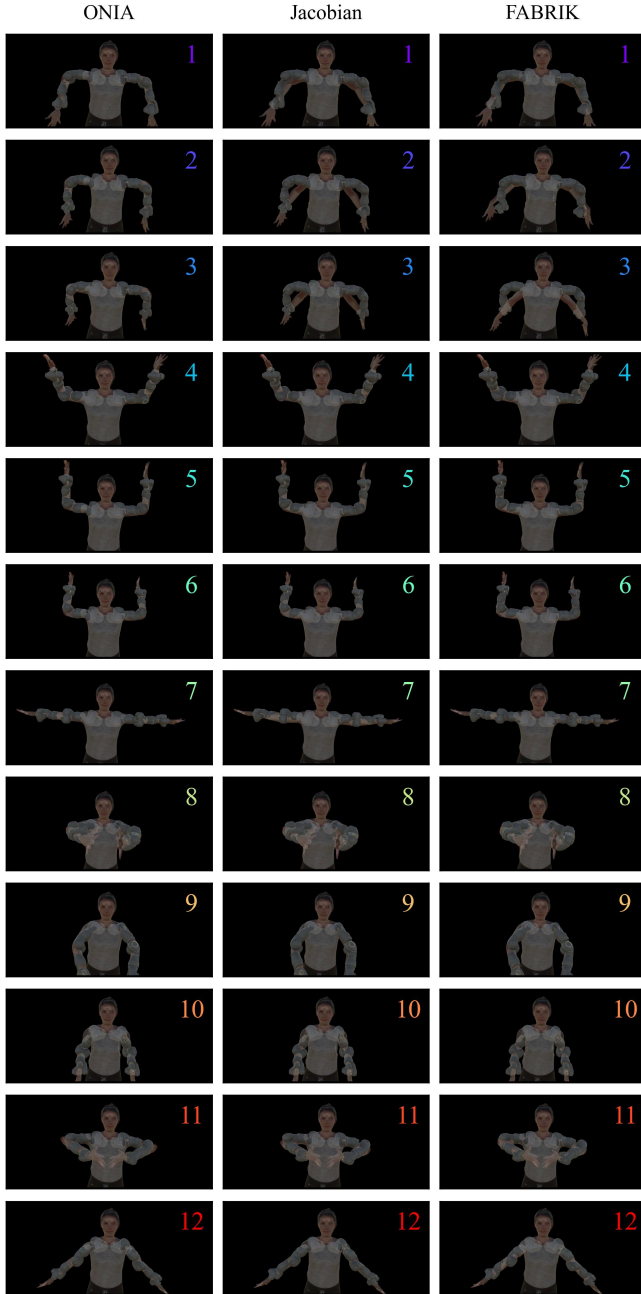


Fig. 5: We evaluated the performance of the solvers on a set of 12 poses. We chose these poses because they are representative of the space of possible arm poses. Pose 1 to 3 all cover “hands down” but differ in forearm orientations. Pose 4 to 6 likewise covers “hands up”. Pose 8 to 11 covers “arms in front,” and pose 7 and 12 are T and A-poses. Note that for the unnatural “hands down” poses, the Jacobian and FABRIK choose natural-looking approximations while ONIA achieves best alignment. FABRIK was trapped in a singularity for pose 3. For other poses, all solvers achieve similar alignment, with ONIA being the fastest. See Fig. 6 for the metrics evaluated at each pose for every solver. Detailed descriptions and results for each pose and each solver are available on our online appendix: <https://avatar-robots.github.io/poses>

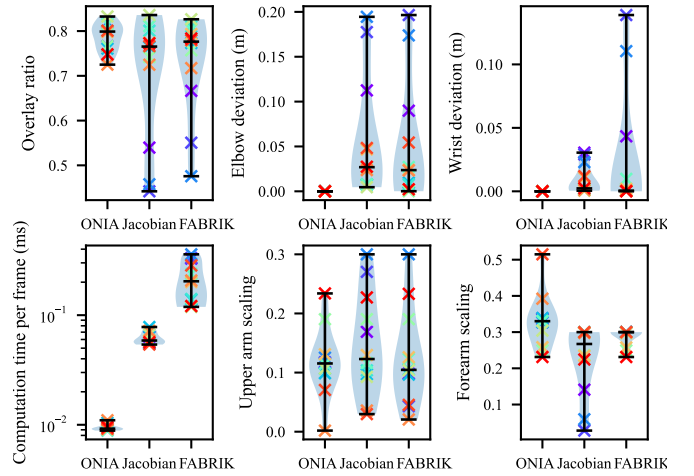


Fig. 6: Violin plots of performance metrics of solvers evaluated from 12 poses. The poses are presented in Fig. 5 and marker colors here matches those in Fig. 5. The minimums, maximums and medians are marked with black bars. We see that ONIA is the fastest of all, achieves the best overlay ratio, and scales the upper arm most conservatively and most consistently.

corresponding reference positions of the robot arm,

$$\Delta x_e := \|\mathbf{x}_s^h + R_s(S_u \mathbf{d}_u^h) - \mathbf{x}_e^r\|, \quad (21)$$

$$\Delta x_w := \|\mathbf{x}_s^h + R_s(S_u \mathbf{d}_u^h) + R_e(S_f \mathbf{d}_f^h) - \mathbf{x}_w^r\|. \quad (22)$$

Smaller deviation should imply better alignment. ONIA is optimal in a sense by ensuring $\Delta x_e = \Delta x_w = 0$, but these metrics still help compare other solvers.

- 3) **Upper & forearm shrink/stretch:** The scaling factors of the human model’s arm segments are S_u (9) and S_f (16). Scaling less aggressively keeps the human model more anthropomorphic. In our plots we present them by their absolute differences to 1 (i.e. $|S_u - 1|$ and $|S_f - 1|$), the smaller the difference, the better.
- 4) **Computation time per frame:** The average time a solver spent on computing alignment pose per frame. Admittedly, the solvers are unlikely to be the performance bottleneck of the entire system since rendering tend to be more resource-intensive. Still, this metric reflects the complexity of different solvers and a faster, simpler one should be preferred if multiple solvers align the human model equally well.

We compute these metrics by logging the robot’s position, arm poses, and the patient’s perspective (camera transform) of a session and replaying them later in simulation. This replay-based approach allows us to compare multiple solvers controlling for the robot trajectory and patient movement. We evaluate our solvers with these metrics in two ways.

B. Evaluating solvers on a set of static poses

We evaluated the performance of the three solvers through a set of 12 static poses, as shown in Fig. 5. Poses are chosen in group, with each group representative of one partition of the space of all possible robot arm poses. Poses 1 to 3 covers the general case of “hands down”, and vary in whether the hand

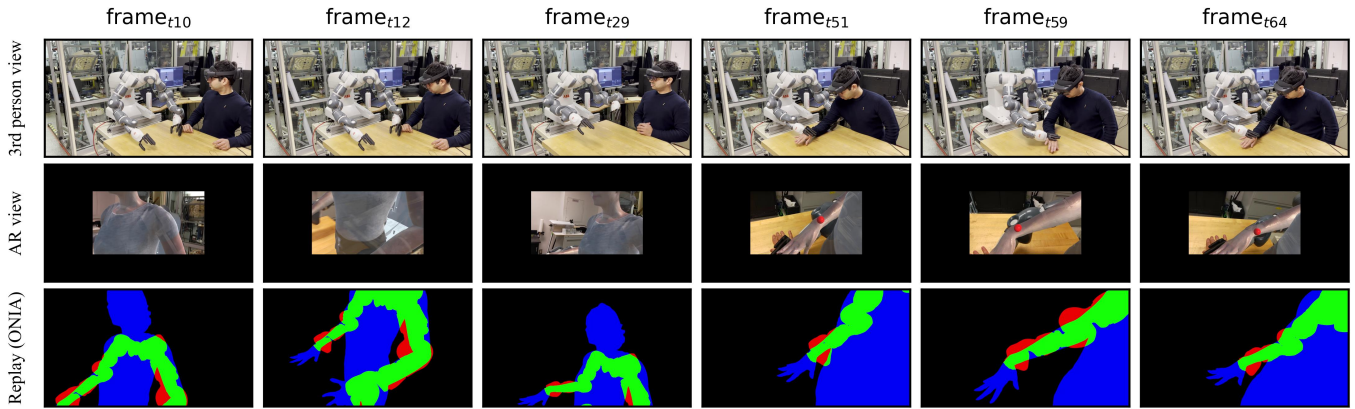


Fig. 7: The key frames of the emulated therapy session. The first row shows the session as seen from an external camera; The second row shows the recording of the AR device the user is wearing, which, due to limitation of the recording software, only captures a small central portion of actual user view; The third row shows the replay scene of the session with ONIA, rendered with a special shader, where green represents parts of the robot arm covered by the human model, red represents uncovered area of the robot arm, and blue represents the rest of the human model. The overlay ratio is then the green area over the red and green area combined. See Fig. 8 for the quantitative analysis. The full video of the session with detailed results can be found in our online appendix: <https://avatar-robots.github.io/sessions/emulated-therapy-session>

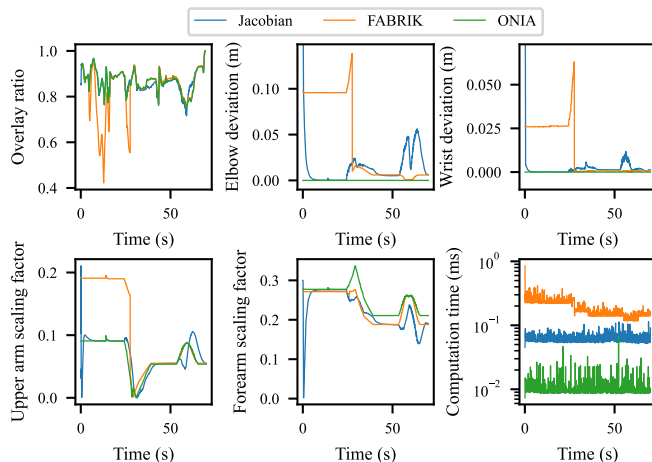


Fig. 8: Metrics of all solvers evaluated throughout the therapy session. ONIA has the highest overlay ratio in average and runs the fastest. It scales the upper arm similar to FABRIK but scales the forearm more aggressively, consistent with section IV-B. FABRIK exhibits poor alignment metrics for initially, and we find Jacobian to be lagging behind robot movements even with the smallest damping necessary to keep it stable.

points forward, upright, or backward; Likewise, poses 4 to 6 covers the case of “hands up.”; Poses 8 to 11 covers different cases where the arms are in front of the body; Poses 7 and 12 are the standard T-pose and A-pose.

We compute metrics for every pose and each solver. The overlay ratio was evaluated from looking at the robot and human model from the front (as shown in Fig. 5). We present the results in Fig. 6.

We noticed that even though computational complexity varies greatly, all three solvers achieve similar alignment. Jacobian and FABRIK has similar elbow and wrist deviations.

Alignment was poor for both baselines on poses 1 to 3 (the “hands down”) poses, because these poses are unnatural for human and lie on the boundary of human anatomical constraints. Incorporating these constraints, both baseline solvers produces only natural-looking approximation to the given robot arm pose. Whether a user prefers visually aligned or natural-looking pose is the subject of future user studies. Still, even when *excluding the 3 outliers*, ONIA achieves the highest median overlay ratio (hence best visual alignment)². Even without explicit scaling constraints, ONIA solver scales the upper arm less aggressively and more consistently (low variance) than both baselines. As of forearm, ONIA tend to scale more aggressively while Jacobian has the most consistent scaling. FABRIK was trapped at a singularity for pose 12, as shown in Fig. 5, with the arm straight and the back of the elbow facing upwards. In general, we find that FABRIK is more likely to suffer from such singularities than Jacobian. Being non-iterative, ONIA runs the fastest, followed by Jacobian and FABRIK which are each an order of magnitude more computationally expensive than one another.

C. Evaluating the Avatar Robot System through an emulated physical therapy session

We further evaluated our solvers through an emulated therapy session: ‘shoulder flexion with elbow extension therapy’³ In the emulated therapy, we program the robot to follow a trajectory that guides the user’s forearm to rotate around his elbow joint. The key frames of the session are shown in Fig. 7, and the metrics evaluated at every frame for every solver are presented in Fig. 8.

²Median overlay ratios for 3 solvers excluding poses 1-3 are not shown in Fig. 6. They are .798, .773, and .783 for ONIA, Jacobian and FABRIK respectively.

³The full video of the emulated therapy session can be found in our online appendix: <https://avatar-robots.github.io/sessions/emulated-therapy-session/>

We noticed that ONIA has the best average overlay ratio throughout the session and runs the fastest. It stretches the upper arm of the model similar to FABRIK but stretches the forearm more aggressively. These are all consistent with our previous findings from in section IV-B. Jacobian lagged behind changes in robot pose, as shown in its spikes of elbow/wrist deviation when the robot is moving the patient’s arm. This is caused by the damping factor of DLS. Reducing the damping further than the current value may improve the responsiveness of Jacobian but causes instability. The damping also causes Jacobian to scale the arm segments differently than the other two solvers did. The left-arm FABRIK solver was caught in a singularity similar to the one it encountered in the static pose evaluation. This caused very poor alignment of the left arm which was reflected in abnormal overlay ratio, elbow/wrist deviations, and upper arm stretching. FABRIK recovered from this singularity on its own in the middle of the session and achieved decent alignment for the second half.

V. LIMITATIONS

We list below the limitations of our approach, which can provide insights to future investigations:

- 1) The solvers consider the alignment of each arm separately. Future alignment solvers could include dual arm poses with the torso to improve the alignment.
- 2) ONIA solver does not limit the scaling of arm segments. While evaluations show that it produces scaling comparable to the baselines in most cases, excessive scaling as high as 1.5 is still observed in pose 8 of Fig. 5. It would be ideal if a scaling limit could be incorporated into future ONIA versions without significantly compromising its theoretical guarantees.
- 3) For this research, we only considered the alignment of arms between the human model and robot (shoulder, elbow, and wrist joints). The alignment of the hand gesture (including the fingers) to the robot’s end effector needs to be explored in the future;
- 4) Our evaluation contains objective metrics only. While this allows precise quantitative analysis of solvers, we expect to benefit from follow-up user studies that can provide new insights about patient perceptions. Subjective evaluations will complement our existing metrics and provide a more comprehensive evaluation.

VI. CONCLUSION

We presented an Avatar Robot System as a mixed real/virtual robotic system that aligns the 3D human model (remote operator) to the robot structure. Therefore, the care receiver might feel as though they are in close contact with the remote operator. We proposed a non-iterative alignment solver that optimally aligns arms poses of the human model with robot joints. We also proposed a set of evaluation metrics that quantitatively measures alignment quality. Our evaluation shows that our non-iterative solver consistently achieves alignment quality comparable to the baselines, meanwhile avoiding their limitations such as singularities, instability, and hyper-parameter sensitivity.

REFERENCES

- [1] A. Helal and B. Abdulrazak, “Tecarob: Tele-care using telepresence and robotic technology for assisting people with special needs,” *International Journal of ARM*, vol. 7, no. 3, pp. 46–53, 2006.
- [2] S. Koceski and N. Koceska, “Evaluation of an assistive telepresence robot for elderly healthcare,” *Journal of medical systems*, 2016.
- [3] P. D. Thacker, “Physician-robot makes the rounds,” *JAMA*, vol. 293, no. 2, pp. 150–150, 2005.
- [4] F. Bernardoni, Ö. Özen, K. Buetler, and L. Marchal-Crespo, “Virtual reality environments and haptic strategies to enhance implicit learning and motivation in robot-assisted training,” in *2019 IEEE 16th international conference on rehabilitation robotics (ICORR)*. IEEE, 2019.
- [5] B. R. Ballester, L. S. Oliva, A. Duff, and P. Verschure, “Accelerating motor adaptation by virtual reality based modulation of error memories,” in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2015, pp. 623–629.
- [6] K. Brüttsch, A. Koenig, L. Zimmerli, S. Mérillat-Koeneke, R. Riener, L. Jäncke, H. J. van Hedel, and A. Meyer-Heim, “Virtual reality for enhancement of robot-assisted gait training in children with central gait disorders,” *Journal of rehabilitation medicine*, 2011.
- [7] R. Agarwal, A. W. Levinson, M. Allaf, D. V. Makarov, A. Nason, and L.-M. Su, “The roboconsultant: telementoring and remote presence in the operating room during minimally invasive urologic surgeries using a novel mobile robotic interface,” *Urology*, 2007.
- [8] B. Jones, Y. Zhang, P. N. Wong, and S. Rintel, “Vroom: virtual robot overlay for online meetings,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–10.
- [9] C. Neustaedter, G. Venolia, J. Procyk, and D. Hawkins, “To beam or not to beam: A study of remote telepresence attendance at an academic conference,” in *Proceedings of the 19th acm conference on computer-supported cooperative work & social computing*, 2016, pp. 418–431.
- [10] Z. Erickson, Y. Gu, and C. C. Kemp, “Assistive vr gym: Interactions with real people to improve virtual assistive robots,” in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 299–306.
- [11] C. Lenz and S. Behnke, “Bimanual telemanipulation with force and haptic feedback through an anthropomorphic avatar system,” *Robotics and Autonomous Systems*, 2023.
- [12] R. Luo and e. a. Wang, “Towards robot avatars: Systems and methods for teleinteraction at avatar xprize semi-finals,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [13] “Avatar xprize.” [Online]. Available: <https://www.xprize.org/prizes/avatar/articles/avatar-technology-this-is-just-the-beginning>
- [14] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.
- [15] A. Aristidou and J. Lasenby, “FABRIK: A fast, iterative solver for the inverse kinematics problem,” *Graphical Models*, 2011.
- [16] “ABB’s collaborative robot — YuMi.” [Online]. Available: <https://new.abb.com/products/robotics/robots/collaborative-robots/yumi/irb-14000-yumi>
- [17] “HoloLens 2 — Overview, Features, and Specs - Microsoft.” [Online]. Available: <https://www.microsoft.com/en-us/hololens/hardware>
- [18] P. Baerlocher and R. Boulic, “Parametrization and range of motion of the ball-and-socket joint,” in *Deformable Avatars: IFIP TC5/WG5. 10 DEFORM’2000 Workshop Geneva, Switzerland*. Springer, 2001.
- [19] F. S. Grassia, “Practical parameterization of rotations using the exponential map,” *Journal of graphics tools*, vol. 3, no. 3, pp. 29–48, 1998.