

DexPilot: Vision-Based Teleoperation of Dexterous Robotic Hand-Arm System

Ankur Handa^{*†} Karl Van Wyk^{*†} Wei Yang[†] Jacky Liang^{†‡} Yu-Wei Chao[†]
Qian Wan[†] Stan Birchfield[†] Nathan Ratliff[†] Dieter Fox[†]

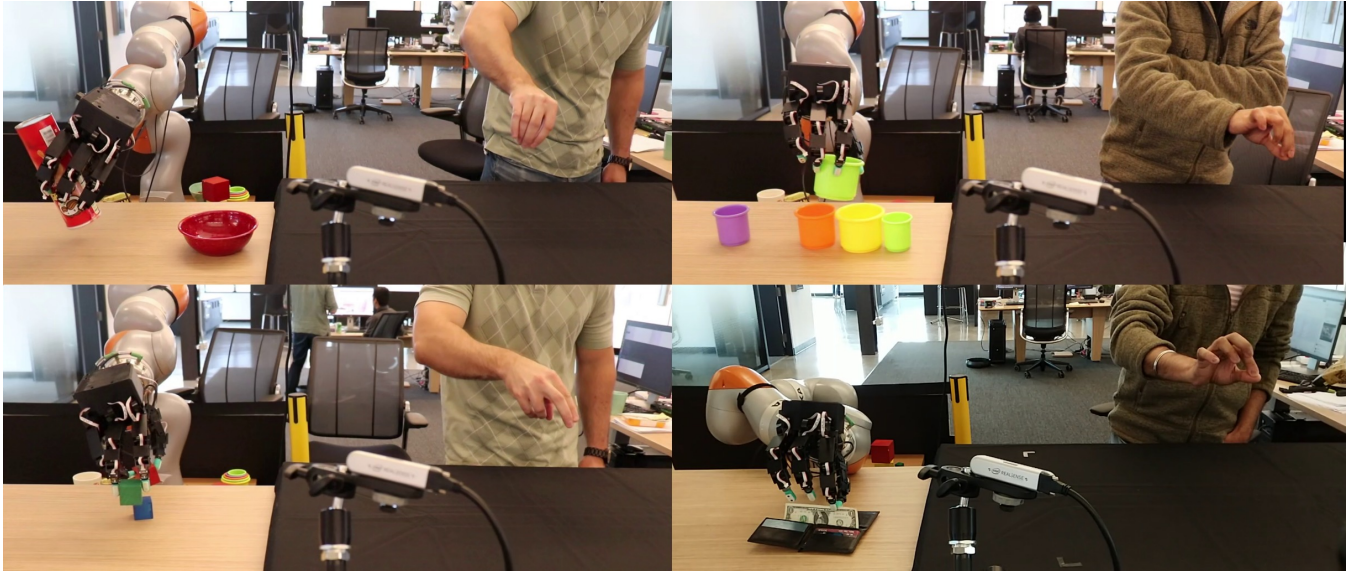


Fig. 1. DexPilot enables teleoperation across a wide variety of tasks, e.g., rectifying a chips can and placing it inside the red bowl (upper-left), inserting cups (upper-right), concurrently picking two cubes with four fingers (lower-left), and extracting paper money from a wallet (lower-right). Videos are available at <https://sites.google.com/view/dex-pilot>.

Abstract—Teleoperation offers the possibility of imparting robotic systems with sophisticated reasoning skills, intuition, and creativity to perform tasks. However, teleoperation solutions for high degree-of-actuation (DoA), multi-fingered robots are generally cost-prohibitive, while low-cost offerings usually offer reduced degrees of control. Herein, a low-cost, depth-based teleoperation system, DexPilot, was developed that allows for complete control over the full 23 DoA robotic system by merely observing the bare human hand. DexPilot enabled operators to solve a variety of complex manipulation tasks that go beyond simple pick-and-place operations and performance was measured through speed and reliability metrics. DexPilot cost-effectively enables the production of high dimensional, multi-modality, state-action data that can be leveraged in the future to learn sensorimotor policies for challenging manipulation tasks. The videos of the experiments can be found at <https://sites.google.com/view/dex-pilot>.

I. INTRODUCTION

Robotic teleoperation has been researched for decades with applications in search and rescue [1], space [2], medicine [3], prosthetics [4], and applied machine learning [5]. The primary motivation of teleoperation is to allow a robot system to perform complex tasks by harnessing the cognition, creativity, and reactivity of humans through a human-machine interface (HMI). Through the years, many advancements have been made in this research field including

the incorporation of haptic feedback [6, 7] and improved human skeletal and finger tracking [8, 9, 10, 11]. Nevertheless, the cost of many high DoA teleoperation or tracking systems is prohibitive. This research bridges this gap by providing a low-cost, markerless, glove-free teleoperation solution that leverages innovations in machine vision, optimization, motion generation, and GPU compute. Despite its low-cost, the system retains the ability to capture and relay fine dexterous manipulation to drive a highly actuated robot system to perform a wide variety of grasping and manipulation tasks. Altogether, four Intel RealSense depth cameras and two NVIDIA GPUs in combination with deep learning and nonlinear optimization produced a minimal-footprint, dexterous teleoperation system. Despite the lack of tactile feedback, the system is highly capable and effective through human cognition. This result corroborates human gaze studies that indicate that humans learn to leverage vision for planning, control, and state prediction of hand actions [12] prior to accurate hand control. Therefore, the teleoperation system exploits the human ability to plan, move, and predict the consequence of their physical actions from vision alone; a sufficient condition for solving a variety of tasks. The main contributions are summarised below:

- Markerless, glove-free and entirely vision-based teleoperation system that dexterously articulates a highly-actuated robotic hand-arm system with direct imitation.
- Novel cost function and projection schemes for kinematically retargeting human hand joints to Allegro hand

^{*} Equal Contribution

[†]NVIDIA, USA

[‡]CMU, Pittsburgh, PA, USA

@{ahanda, kvanwyk, dieterf}@nvidia.com

joints that preserve hand dexterity and feasibility of precision grasps in the presence of any tracking error.

- Demonstration of teleoperation on a wide variety of tasks involving fine manipulations and dexterity, *e.g.*, pulling out paper currency from wallet and grasping two cubes with four fingers as shown in Fig. 1.
- System assessment across two trained human demonstrators — also called pilots — revealed that high task success rates can be achieved despite the lack of tactile feedback.

II. RELATED WORK

Teleoperation via the human hand is traditionally done using either pure vision or sensorized glove-based solutions. Early vision-based solutions include Dorner *et al.* [13] who used colour markers on the hand to track the positions of joints and hand pose. Later, [14] and [15] propose extensions of the colour glove to track the joints and pose of the hand for augmented reality (AR) and virtual reality (VR) applications. Alternatively, [16] tracked hands with an array of motion capture cameras with data association between tracked markers via deep learning for various virtual reality applications. Li *et al.* [17] also proposed vision-based teleoperation through deep learning that calculated Shadow robotic hand joint angles from observed human hand depth images. However, this approach was not extended to a full robotic system and the mapping results had performance issues as later discussed in Section VII-A. In contrast to vision-based methods, glove-based solutions offer a different paradigm for human hand measurement. For example, [18] and [19] showed that low-cost solutions for teleoperation with a glove can be constructed, though they tend to be quite bulky and have wires tethered to them. Few commercial, glove-based solutions also exist including CyberGlove [7] and HaptX [6]. These approaches do provide the user with tactile feedback in the form of physical resistance and spatial contact, but this benefit comes with heightened cost and additional bulk to the user. Experience with DexPilot revealed the importance of ergonomics of solution to reduce the rate of user fatigue on multi-trial and long-range demonstrations. Still, these other solutions are important to investigating how relaying tactile feedback to human pilots improve teleoperation capability, considering that numerous neurophysiological studies have uncovered the importance of tactile feedback in human sensorimotor control of the human hand [12, 20, 21].

III. HARDWARE

The teleoperation setup comprises a robot system and an adjacent human pilot arena as shown in Fig. 3. The robot system was a KUKA LBR iiwa7 R800 series robot arm and a Wonik Robotics Allegro hand. The Allegro hand was retrofitted with four SynTouch BioTac tactile sensors at the fingertips and 3M TB641 grip tape applied to the inner surfaces of the phalanges and palm. The rubbery surfaces of both the BioTac sensors and 3M tape improves friction of the hand while the BioTacs themselves produced 23 signals that can later be used to learn sensorimotor control from demonstrations. The studio is a black-clothed table surrounded by four calibrated and time-synchronized Intel Realsense D415 RGB-D cameras spatially arranged to cover a workspace of $80 \times 55 \times 38 \text{ cm}^3$. It is directly adjacent

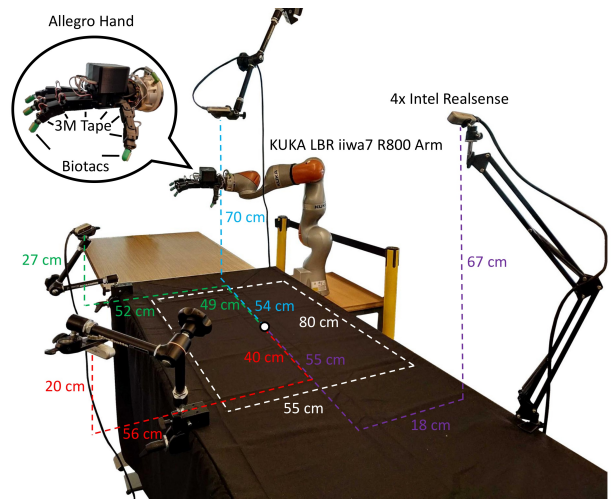


Fig. 3. The studio is composed of four RGBD cameras pointing towards the table over which the user moves their hand with the hand-arm system in close proximity to enable line-of-sight teleoperation.

to the robot to improve line-of-sight and visual proximity since teleoperation is entirely based on the human vision and spatial reasoning.

IV. ARCHITECTURE

To produce a natural-feeling teleoperation system, an imitation-type paradigm is adopted. The bare human hand motion — pose and finger configuration — is constantly observed and measured by the visual perception module. The human hand motion is then relayed to the robot system in such a way that the copied motion is self-evident. This approach enables the human pilot to curl and arrange their fingers, form grasps, reorient and translate their palms, with the robot system following in a similar manner. DexPilot relies heavily on Dense Articulated Real-Time Tracking (DART) [22], which forms the backbone of tracking the pose and joint angles of the human hand.

In the following, we explain the main components of the overall system [23]: 1) DART for hand tracking (Section V). 2) deep neural networks for human hand state estimation and robustifying DART (Section VI). 3) human hand state conversion through nonlinear optimization to Allegro hand states (Section VII-A). 4) motion generation and control through Riemannian Motion Policies (RMPs) and torque-level impedance controllers (Section VII-B). The full system architecture and component connections are shown in Fig. 4. Overall, the system produces a latency of about one second.

V. DART

DART is used for continuous pose and joint angle tracking of the human hand. DART requires an articulated model of the hand that is registered against an input point cloud. The human hand model was obtained from [24] and turned into a single mesh model in [25]. With CAD software, the fingers of the mesh model were separated into their respective proximal, medial, and distal links, and re-exported as separate meshes along with an associated XML file that describes their kinematic arrangement. In total, the human hand model possesses 20 revolute joints: four joints per finger with one abduction joint and three flexion joints.

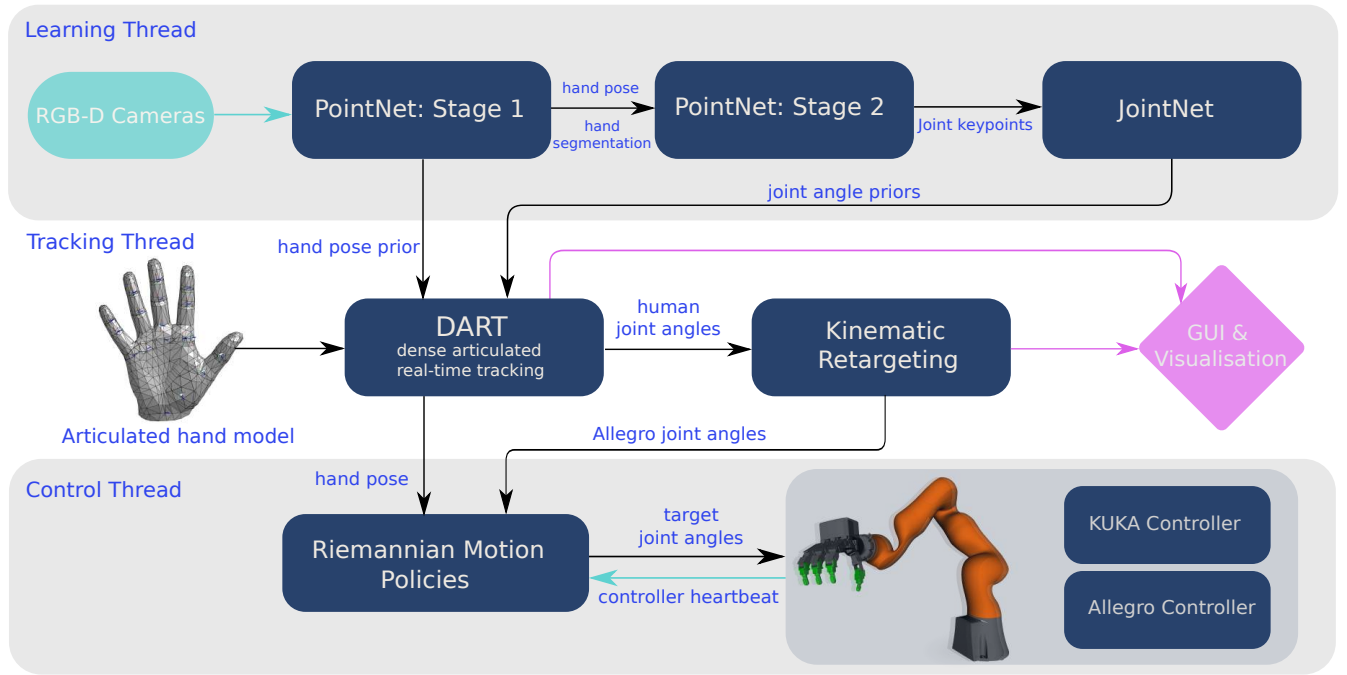


Fig. 4. The system is comprised of three threads that operate on three different computers. The learning thread provides hand pose and joint angle priors using fused input point cloud coming from four cameras from the studio. The tracking thread runs DART for hand tracking with the priors as well as kinematic retargeting needed to map human hand configuration to allegro hand. The control thread runs the Riemannian motion policies to provide the target joint commands to the KUKA and allegro hand given the hand pose and joint angles.

VI. ESTIMATING HAND POSE AND JOINT ANGLES WITH NEURAL NETWORKS

DART is a model-based tracker that relies on non-linear optimisation and initialisation (typically from the previous frame or an initial guess). If this initialisation is not within the basin of convergence, the tracker can fail to provide the correct solution. When tracking the human hand with DART on point cloud data, the hand model often snaps on to spurious local minima leading to tracking failures every few minutes. Therefore, to reliably track the human hand over long periods of time — as needed for teleoperation — it is desirable to have reliable hand pose priors, clean hand segmentation, and a multi-view camera studio to prevent the hand model from snapping on to unexpected local minimas. Although various datasets [9, 10] exist that provide the necessary annotations for hand pose and segmentation, these datasets were not suitable primarily due to different sensor characteristics and lack of hand poses that were necessary for dexterous manipulation with DexPilot.

Data collection with DART happens in two phases [23]. In the first phase, we use a coloured fabric glove to obtain hand-pose priors and collect a large scale data with joint position annotations. We train a neural network to map the point cloud from depth sensors to the corresponding joint positions. In the second phase, we use DART and the latest trained neural network serving priors to generate increasing amounts of data. The network is perpetually updated with the latest dataset to generate increasingly better priors for DART, ultimately extending the range over which DART can operate without any failures. The hand pose neural network is a PointNet++ based [26] architecture which operates directly

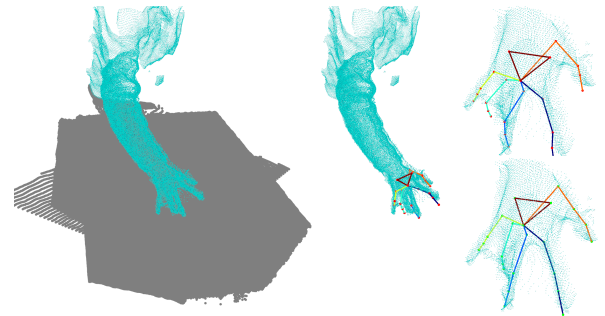


Fig. 5. The input point cloud has points both from the table as well as the human body and arm. A plane was fit to remove points on the table and the remaining points are input to the first stage of PointNet that recovers the pose of the hand. The second stage refines the pose and provides a more accurate result. The hand images on the right show the result from stage 1 (above) and stage 2 (below).

on the fused point cloud data obtained by back-projecting depth images from extrinsically calibrated depth cameras into a single global reference frame with annotations provided by DART. Since the fused point cloud contains both the points on table as well as human body and arm, it is imperative to first localise the hand. Points are removed from the table by fitting a plane and feeding the remaining points containing the arm and human body to the network that localises the hand as well as provides the hand pose (Fig. 5). Our network architecture is based on [11] which estimates hand joint positions via a vote-based regression scheme to the 3D positions of specified keypoints on the hand, a technique reminiscent of spatial-softmax often used in 2D keypoint localisation. It is trained to predict 3D coordinates of 23 keypoints specified on the hand — four joint keypoints for

each of the five fingers and three keypoints on the back of the hand for hand pose estimation. The loss function is the Euclidean distance between predicted and the ground truth keypoints. Additionally, an auxiliary segmentation loss is included for hand segmentation. For efficiency reasons, input point cloud of size $N \times 3$ is sub-sampled uniformly to a fixed 8192×3 size before feeding it to the network. While reasonable hand pose estimation and segmentation can be achieved via this network, high quality predictions for the 20 joint keypoints on the fingers is hard. The uniform sub-sampling used at the input means that points on the fingers are not densely sampled, and therefore a second stage refinement [23] is needed which resamples points on the hand from the original raw point cloud given the hand pose and segmentation from the first stage. The second stage is trained on the same loss functions, but uses only the points sampled on the hand instead [23] and accurately predicts the 23 keypoints — 4 keypoints per finger and 3 at the back of the palm. To enable robustness to any inaccuracies, random perturbations were added to the hand pose for second stage. Fig. 5 shows that the second stage refinement improved the system. Overall, both stages of the network are trained on 100K point clouds collected over a batch of 30-45 minutes each for 7-8 hours in total by running DART that provided annotations for keypoints, joint angles and segmentation. The training takes 15 hours on a single NVIDIA Titan Xp GPU. To provide joint angle priors for fingers for DART, a third neural network is trained that maps keypoint locations predicted by two-stage network to corresponding joint angles. This neural network, called JointNet, is a two-layer fully connected network that takes input of size 23×3 and predicts 20-dimensional vector of joint angles for fingers [23].

All the neural networks are trained on data collected within the limits of the studio work volume across multiple human hands, ensuring accurate pose fits for this application and enabling sensible priors for DART. Qualitatively, the hand tracker works well for hands geometrically close to the DART human hand model. Overall, average keypoint error on a validation set of seven thousand images of differing hand poses and finger configurations was 9.7 mm (comparable to results reported in [11], but on a different dataset) and joint error was 1.33 deg per joint.

VII. ROBOT MOTION GENERATION

A. Kinematic Retargeting

Teleoperation of a robotic hand that is kinematically disparate from the human hand required a module that can map the observed human hand joints to the Allegro joints. There are many different approaches to kinematic retargeting in the literature. For instance, a BioIK solver was used to match (between the human and Shadow hand) the positions from palm to the fingertips and medial joints, and the directionality of proximal phalanges and thumb distal phalange in [17]. The optimized mapping was used to label human depth images to learn end-to-end a deep network that can ingest a depth image and output joint angles for the Shadow hand. Although interesting, the result produced retargeting results that are not useful for precision grasps (e.g., pinching) where gaps between fingertips need to be small or zero. Motion retargeting is also popular in the animation field. For instance,

a deep recurrent neural network was unsupervised trained to retarget motion between skeletons [27]. Although the synthesized motion looks compelling, it is unclear whether these approaches work well enough for teleoperative manipulation where the important task spaces for capturing grasping and manipulation behavior are likely not equivalent to those that capture visual consistencies. The approach herein prioritizes fingertip task-space metrics because distal regions are of highest priority in grasping and manipulation tasks as measured by their contact prevalence [28], degree of innervation [12], and heightened controllability for fine, in-hand manipulation skill [29]. Moreover, the joint axes and locations between the two hands are strikingly different, and therefore, no metrics directly comparing joint angles between the two hands are used. To capture and optimize for the positioning of fingertips, both distance and direction among fingertips are considered. Specifically, the cost function for kinematic retargeting [23] is chosen as:

$$\mathcal{C}(q_h, q_a) = \frac{1}{2} \sum_{i=0}^N s(d_i) \|r_i(q_a) - f(d_i) \hat{r}_i(q_h)\|^2 + \gamma \|q_a\|^2,$$

where q_h, q_a are the angles of the human hand model and Allegro hand, respectively, $r_i \in \mathbb{R}^3$ is the vector pointing from the origin of one coordinate system to another, expressed in the origin coordinate system (see Fig. 6). Furthermore, $d_i = \|r_i(q_h)\|$ and $\hat{r}_i(q_h) = \frac{r_i(q_h)}{\|r_i(q_h)\|}$. The switching weight function $s(d_i)$ is defined as

$$s(d_i) = \begin{cases} 1, & d_i > \epsilon \\ 200, & d_i \leq \epsilon \wedge r_i(q_h) \in \mathcal{S}_1 \\ 400, & d_i \leq \epsilon \wedge r_i(q_h) \in \mathcal{S}_2, \end{cases}$$

where $\mathcal{S}_1$ and $\mathcal{S}_2$ are vector sets defined in Table I, The distancing function, $f(d_i) \in \mathbb{R}$, is defined as

$$f(d_i) = \begin{cases} \beta d_i, & d_i > \epsilon \\ \eta_1, & d_i \leq \epsilon \wedge r_i(q_h) \in \mathcal{S}_1 \\ \eta_2, & d_i \leq \epsilon \wedge r_i(q_h) \in \mathcal{S}_2, \end{cases}$$

where $\beta = 1.6$ is a scaling factor, $\eta_1 = 1 \times 10^{-4} m$ closes the distance between a primary finger and the thumb, and $\eta_2 = 3 \times 10^{-2} m$ forces a minimum separation distance between two primary fingers when both primary fingers are being projected with the thumb. These projections ensure that contacts between primary fingers and the thumb are close without inducing primary finger collisions in a precision grasp. This was found to be particularly useful in the presence of visual finger tracking inaccuracies. Importantly, the vectors r_i not only capture distance and direction from one task space to another, but their expression in local coordinates further contains information on how the coordinate systems, and thereby fingertips, are oriented with one another. The coordinate systems of the human hand model must therefore have equivalent coordinate systems on the Allegro model with similarity in orientation and placement. The vectors shown in Fig. 6 form a minimal set that produced the desired retargeting behavior. Finally, $\gamma = 2.5 \times 10^{-3}$ is a weight on regularizing the Allegro angles to zero (equivalent to fully opened the hand). This term helped greatly with reducing redundancy in solution and ensured that the hand

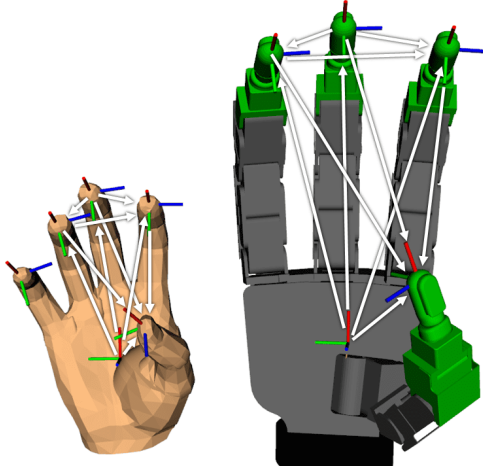


Fig. 6. Task space vectors between fingertips and palm for both the human hand model and Allegro hand used for retargeting optimization.

TABLE I

DESCRIPTION OF VECTOR SETS USED IN KINEMATIC RETARGETING.

Set	Description
S_1	Vectors that originate from a primary finger (index, middle, ring) and point to the thumb.
S_2	Vectors between two primary fingers when both primary fingers have associated vectors $\in S_1$, e.g., both primary fingers are being projected with the thumb.

never entered strange minima that was difficult to recover from (e.g., the fingers embedding themselves into the palm). Various mappings from human hand to allegro as produced by our kinematic retargeting are show in Fig. 7.

For implementation, the above cost function is minimized in real-time using the Sequential Least-Squares Quadratic Programming (SLSQP) algorithm [30, 31] of the NLOpt library [32]. The routine is initiated with Allegro joint angles set to zero, and every solution thereafter is initiated with the preceding solution. Moreover, the forward kinematic calculations between the various coordinate systems of both the human hand model and Allegro hand are found using the Orocos Kinematics and Dynamics library [33]. Finally, a first-order low-pass filter is applied to the raw retargeted joint angles in order to remove high-frequency noise present in tracking the human hand and to smooth discrete events, like the projection algorithm inducing step-response changes in retargeted angles.

B. Riemannian Motion Policies

Riemannian Motion Policies (RMPs) are real-time motion generation methods that calculate acceleration fields from potential function gradients and corresponding Riemannian metrics [34, 35]. RMPs combine the generation of multi-priority Cartesian trajectories and collision avoidance behaviors together in one cohesive framework. They are used to control the Cartesian pose of the Allegro palm given the observed human hand pose while avoiding arm-palm collisions with the table or operator using collision planes. Given these objectives, the RMPs generated target arm joint trajectories that were sent to the arm's torque-level impedance controller at 200 Hz. The kinematically retargeted

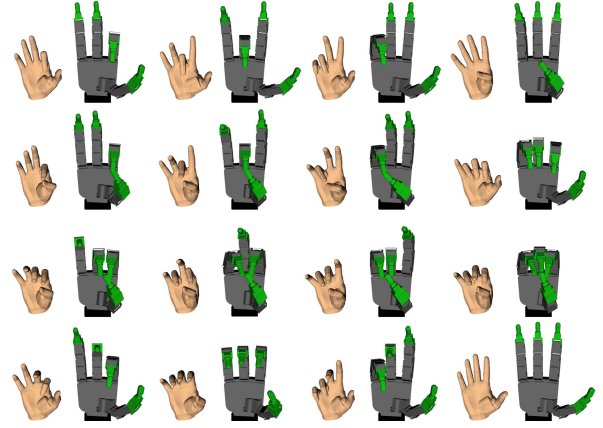


Fig. 7. Canonical kinematic retargeting results between the human hand model and the Allegro hand model.

Allegro angles are sent to the torque-level joint controller at 30 Hz. A teleoperation instance is initialized by registering the studio cameras with the robot base coordinate system via an initial, static robot pose and the initial observation of the human hand. For convenience, the human hand model axes and robot end-effector axes are approximately aligned such that direction of movements are preserved between human hand motion and robot motion.

VIII. EXPERIMENTS

The DexPilot system is systematically tested across a wide range of physical tasks that test precision and power grasps, prehensile and non-prehensile manipulation, and finger gaiting (see Fig. 8). The description of the tasks are as follows:

- 1) **Pick and Place (P&P) Brick:** Pick brick and place in red bowl.
- 2) **P&P Pringles Box:** Pick chips box and place in red bowl standing straight.
- 3) **P&P Spam:** Pick spam can and place in red bowl.
- 4) **Pouring:** Pick cup and pour beads into red bowl.
- 5) **Cup Insertion:** Pick and insert concentric cups.
- 6) **Blocks (L):** Stack large blocks (blue, red, yellow).
- 7) **Blocks (M):** Stack medium blocks (blue, red, green).
- 8) **Blocks (S):** Stack small blocks (blue, red, green).
- 9) **Wallet:** Open wallet and extract paper currency.
- 10) **Container:** Open plastic container, extract and open cardboard box.
- 11) **Peanut Jar:** Open peanut jar and place lid on table.
- 12) **Brick Gaiting:** Pick up and in-hand rotate brick 180 degrees and place back down.
- 13) **Tea:** Pull open tea drawer, extra single bag of tea and place on table, close tea drawer.

Before benchmarking, the pilots went through a warm-up training phase where they tried to solve the task with 3-5 non-consecutive attempts. Later, five consecutive test trials are conducted by each pilot for each task to avoid preferential selection of results and pilots were graded based on their performance. The performance metrics for these tasks include mean completion time (CT) and success rate which capture speed and reliability of the teleoperation system. The system is tested with two pilots and the performance measures are reported in Figs. 9 - 10 (see videos at <https://sites.google.com/view/dex-pilot>).



Fig. 8. Objects used for teleoperation testing. Various tasks performed with these objects can be viewed at <https://sites.google.com/view/dex-pilot>.

Overall, the system can be reliably used to perform a variety of tasks spanning a range of difficulty. Differences in mean CT across tasks indicate the effects of task complexity and horizon scale. Discrepancies in mean CT across pilots per task indicate that there does exist a dependency on pilot behavior. Effects include fatigue, training, creativity, and motivation. The ability to perform these tasks reveal that the teleoperation system has the dexterity to exhibit precision and power grasps, multi-fingered prehensile and non-prehensile manipulation, in-hand finger gaiting, and compound in-hand manipulation (*e.g.*, grasping with two fingers while simultaneously manipulating with the remaining fingers). Certain tasks, *e.g.* Container and Wallet, have longer mean CTs largely due to the fact that these tasks are very long horizon tasks. By contrast, the Blocks (S) task was particularly challenging because of the projection used in kinematic retargeting. While the projection routines enabled grasping of small objects, their release often induced undesirable, antagonistic motion in the object that led to incorrect part placement. Nevertheless, such a rich exhibition of dexterous skill transferred solely through the observation of the bare human hand provides empirical evidence that the approach and architecture herein works well. Although the full teleoperation process for a particular task may not be perfect (*e.g.*, unintentional part loss), the pilot can immediately recover and ultimately complete the task. This data is equally valuable in helping the robot to learn to recover from failures. Additionally, the data can be regarded as play data which is useful to learn long-range planning [36].

IX. DISCUSSION

DexPilot enables a highly-actuated hand-arm system to find a motor solution to a variety of manipulation tasks by translating observed human hand and finger motion to robot arm and finger motion. Importantly, several tasks, like extracting paper money from a wallet and opening a cardboard box within a plastic container, are so complex that hand-engineering a robot solution or applying learning methods directly are likely intractable. Performing these tasks and others through the embodied robotic system reveals that a sensorimotor solution did exist and that these solutions can be generated on-demand for many demonstrations. Furthermore, creating these solutions on the system itself allows for the reading, access, and storage of the 92 tactile signals in the robot's fingertips, 23 commanded and measured joint

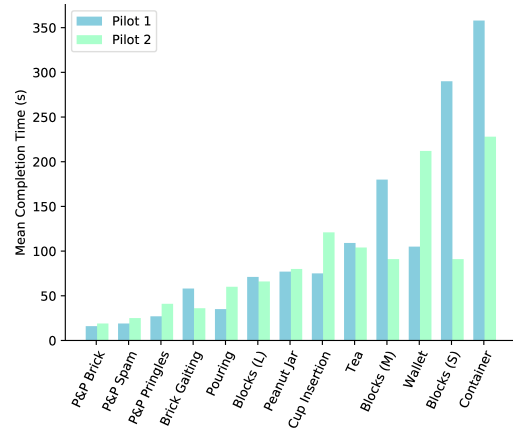


Fig. 9. Mean completion time of teleoperation tasks across two pilots run over five successive trials without reset.

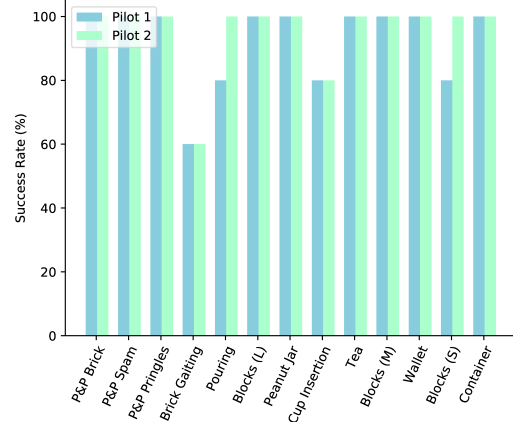


Fig. 10. Success rate of teleoperation tasks across two pilots run over five successive trials without reset.

position and velocity signals through the hand and arm, 23 torque commands throughout the system, and any camera feeds associated with the system. This rich source of data together with demonstrations of tasks will be critical for applications of learning methods that may hopefully learn to solve complex, multi-stage, long horizon tasks much like demonstration data has enabled huge leaps in game AI in StarCraft [37] and Dota2 [38].

Moreover, the current DexPilot system can be improved in a variety of ways in the future. For instance, human hand tracking accuracy can be improved with advancements in deep learning architectures, bigger data sets, and changes to imaging hardware. Ideally, human hand tracking accuracy should be sufficiently improved to greatly reduce the projection distance in the kinematic retargeting approach, enhancing fine finger control and manipulation over small objects and multi-fingered precision grasps. Grasp and manipulation control algorithms [29] could be implemented on the hand that automate force modulating control to reduce burden on the user and minimizing unintentional part drops from the application of incorrect grip forces.

REFERENCES

- [1] A. Norton, W. Ober, L. Baraniecki, E. McCann, J. Scholtz, D. Shane, A. Skinner, R. Watson, and H. Yanco, "Analysis of human-robot interaction at the darpa robotics challenge finals," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 483-513, 2017.

- [2] M. Diftler, T. Ahlstrom, R. Ambrose, N. Radford, C. Joyce, N. De La Pena, A. Parsons, and A. Noblitt, "Robonaut 2 initial activities on-board the iss," in *2012 IEEE Aerospace Conference*. IEEE, 2012, pp. 1–12.
- [3] J. R. Sterbis, E. J. Hanly, B. C. Herman, M. R. Marohn, T. J. Broderick, S. P. Shih, B. Harnett, C. Doarn, and N. S. Schenkman, "Transcontinental telesurgical nephrectomy using the da vinci robot in a porcine model," *Urology*, vol. 71, no. 5, pp. 971–973, 2008.
- [4] K. Z. Zhuang, N. Sommer, V. Mendez, S. Aryan, E. Formento, E. D'Anna, F. Artoni, F. Petrini, G. Granata, G. Cannaviello, W. Raffoul, A. Billard, and S. Micera, "Shared human–robot proportional control of a dexterous myoelectric prosthesis," *Nature Machine Intelligence*, 2019.
- [5] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *ICRA*. IEEE, 2018, pp. 1–8.
- [6] "HaptX," <https://haptx.com/>.
- [7] "CyberGlove Systems," www.cyberglovesystems.com.
- [8] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *PAMI*, 2006.
- [9] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM ToG*, 2014.
- [10] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Big-Hand2.2m benchmark: Hand pose dataset and state of the art analysis," in *CVPR*, 2017.
- [11] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3d hand pose estimation," in *ECCV*, 2018.
- [12] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, p. 345, 2009.
- [13] B. Dorner, "Chasing the colour glove: Visual hand tracking," Ph.D. dissertation, Simon Fraser University, 1994.
- [14] C. Theobalt, I. Albrecht, J. Haber, M. Magnor, and H.-P. Seidel, "Pitching a baseball: tracking high-speed motion with multi-exposure images," in *ACM TOG*, 2004.
- [15] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM ToG*, 2009.
- [16] S. Han, B. Liu, R. Wang, Y. Ye, C. D. Twigg, and K. Kin, "Online optical marker-based hand tracking with deep labels," *ACM TOG*, 2018.
- [17] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network," in *ICRA*, 2019.
- [18] H. Liu, Z. Zhang, X. Xie, Y. Zhu, Y. Liu, Y. Wang, and S.-C. Zhu, "High-fidelity grasping in virtual reality using a glove-based system," in *ICRA*, 2019.
- [19] H. Liu, X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S.-C. Zhu, "A glove-based system for studying hand-object manipulation via joint pose and force sensing," in *IROS*, 2017.
- [20] R. S. Johansson and G. Westling, "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects," *Experimental brain research*, vol. 56, no. 3, pp. 550–564, 1984.
- [21] R. S. Johansson and Å. B. Vallbo, "Tactile sensory coding in the glabrous skin of the human hand," *Trends in neurosciences*, vol. 6, pp. 27–32, 1983.
- [22] T. Schmidt, R. A. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking," in *RSS*. IEEE, 2014.
- [23] A. Handa, K. V. Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "DexPilot: Vision Based Teleoperation of Dexterous Robotic Hand-Arm System," *arXiv preprint arXiv:1910.03135*, 2019.
- [24] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM SIGGRAPH Asia*, 2017.
- [25] Y. Hasson, G. Varol, D. Tzionas, I. Kalevtykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.
- [26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *CoRR*, vol. abs/1706.02413, 2017.
- [27] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargeting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8639–8648.
- [28] S. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, and W. Matusik, "Learning the signatures of the human grasp using a scalable tactile glove," *Nature*, vol. 569, no. 7758, p. 698, 2019.
- [29] K. Van Wyk, M. Culleton, J. Falco, and K. Kelly, "Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand," *IEEE Transactions on Robotics*, 2018.
- [30] D. Kraft, "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [31] —, "Algorithm 733: Tomp–fortran modules for optimal control calculations," *TOMS*, 1994.
- [32] S. G. Johnson, "The nlopt nonlinear-optimization package," <http://github.com/stevengj/nlopt>.
- [33] "Orocos kinematics and dynamics library," https://github.com/orocos/orocos_kinematics_dynamics.
- [34] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, "Riemannian motion policies," *arXiv preprint arXiv:1801.02854*, 2018.
- [35] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "Rmpflow: A computational graph for automatic motion policy generation," *arXiv preprint arXiv:1811.07049*, 2018.
- [36] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," *arXiv preprint arXiv:1903.01973*, 2019.
- [37] "AlphaStar," <https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii>.
- [38] "OpenAI Five Finals," <https://openai.com/blog/openai-five-finals/>.