2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

iRoCo: Intuitive Robot Control From Anywhere Using a Smartwatch

Fabian C Weigend[1], Xiao Liu[1], Shubham Sonawani[1], Neelesh Kumar[2], Venugopal Vasudevan[2] and Heni Ben Amor[1]

***Abstract*— This paper introduces iRoCo (intuitive Robot Control) – a framework for ubiquitous human-robot collaboration using a single smartwatch and smartphone. By integrating probabilistic differentiable filters, iRoCo optimizes a combination of precise robot control and unrestricted user movement from ubiquitous devices. We demonstrate and evaluate the effectiveness of iRoCo in practical teleoperation and drone piloting applications. Comparative analysis shows no significant difference between task performance with iRoCo and gold-standard control systems in teleoperation tasks. Additionally, iRoCo users complete drone piloting tasks 32% faster than with a traditional remote control and report less frustration in a subjective load index questionnaire. Our findings strongly suggest that iRoCo is a promising new approach for intuitive robot control through smartwatches and smartphones from anywhere, at any time. The code is available at `www.github.com/wearable-motion-capture`**

I. Introduction

Human motion tracking and state estimation are essential components of many robotics approaches, such as teleoperation [1], exoskeleton control [2], learning from demonstration [3], and in particular human-robot collaboration [4]. A major challenge faced in these fields is the reliance on costly motion capture systems, e.g, OptiTrack [5], which are restricted to stationary setups and require elaborate calibration procedures. For more flexibility, non-optical motion capture systems utilizing Inertial Measurement Units (IMUs) have gained traction [6], [7]. However, most IMU-based methods still require precise placement of specialized units on the user's body and intricate calibration procedures [8], [9].

To address these drawbacks and enable human pose estimation outside of the lab, recent works have investigated the potential of smart devices for IMU-based motion capture. These approaches promise truly ubiquitous human-robot collaboration, as motion capture through smart devices is widely accessible and intuitive for untrained users [10], [11]. Studies have specifically examined the use of a single smartwatch for motion capture [12]–[14], along with the development of human-robot control interfaces based on wearable technology [10], [11], [15]. However, existing works in smartwatch motion capture impose constraints on user movements and orientation, thereby reducing their utility and practicality [11]–[13].

This paper expands upon previous work by augmenting the sensor data from a smartwatch with a connected smartphone.

[1]Weigend, F, Liu, X, Sonawani, S & Ben Amor, H are affiliated with SCAI, Arizona State University {fweigend, xliu330, sdsonawa, hbenamor}@asu.edu. [2]Kumar, N & Vasudevan, V are affiliated with Corporate Functions-R&D, Procter and Gamble {kumar.n.40,vasudevan.v}@pg.com

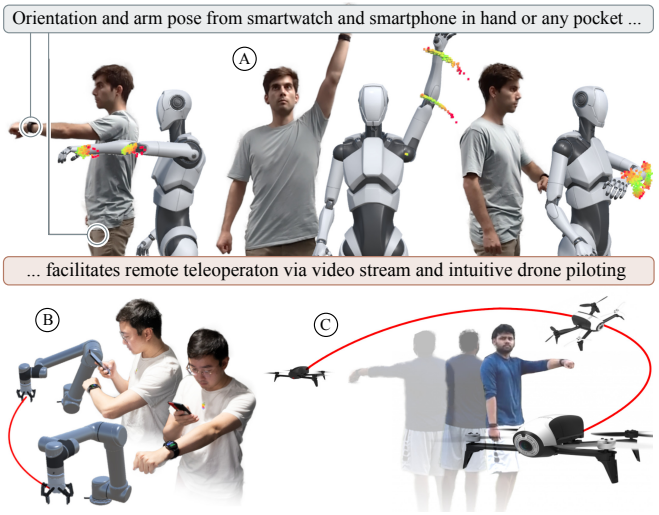

Fig. 1. **A)** We present an **i**ntuitive **Ro**bot **Co**ntrol (iRoCo) framework, which achieves robust body orientation and arm pose estimations from a smartwatch and a smartphone. **B)** The system allows teleoperation through video streaming to a smartphone, enabling tasks such as pick-and-place operations. **C)** We demonstrate iRoCo's potential for intuitive drone piloting.

Using the combined data, we leverage advancements in neural state estimation techniques, specifically differentiable recursive filters [16]–[18], to allow users to move freely in their environment, thereby addressing limitations of previous approaches for smartwatch motion capture. We employ a differentiable Bayesian state estimator, augment it with a tailored control modality and present the entire system as **i**ntuitive **Ro**bot **Co**ntrol (iRoCo). As illustrated in Figure 1, iRoCo optimizes the balance between unrestricted movements and reliable pose estimations for various human-robot interaction tasks. We demonstrate the advantages of iRoCo by evaluating the user experience and comparing it to gold-standard baselines in teleoperation and drone piloting tasks. We summarize our contributions as:

- Introducing iRoCo, an intuitive robot control interface enabling robot control anytime and anywhere from a single smartwatch and smartphone.
- iRoCo integrates a differentiable filter algorithm for refined human pose estimation with uncertainty.
- We design a tailored control modality for ubiquitous and intuitive teleoperation by a human operator.
- We demonstrate the advantages of iRoCo in two applications: teleoperated pick-and-place tasks and drone piloting.

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

II. Related Work

Smart device-based human-robot systems provide a wide range of possibilities to enhance the flexibility and robustness of human-robot interactions. These include navigating mobile robots using a single smartwatch [15], automated monitoring of vital body functions through smartwatches [19], [20], and the coordination of multi-robot systems [10]. The utilization of wearable devices has demonstrated the potential to alleviate user fatigue and reduce mental workload in various robot control tasks [10]. However, numerous applications heavily rely on accurate gesture recognition by human users. Consequently, these systems suffer from reduced precision, despite their intuitive nature.

Aware of these challenges, the works of [12]–[14] proposed approaches to make more accurate arm pose predictions from smartwatch sensor data. In previous work, we presented approaches to utilize such arm pose predictions for teleoperation and robot control through voice commands [11]. While these systems are promising for applications in robotics, they impose calibration procedures and fixed body-forward facing constraints on the user [12], [13]. Or, they limit inference to the same environment where the model was trained [14].

To address the limitations associated with human arm pose estimation from smart devices, we propose the utilization of differentiable filters (DFs) [16]–[18]. DFs are a subclass of algorithms derived from the Deep State-Space model [16], [21], [22] and offer a solution to combine the principles of Bayesian recursive filtering while learning state transition and measurement models from data. Previous research [16] has demonstrated that DFs can effectively learn noise profiles through end-to-end training, and their efficacy has been evidenced in various real-world applications [23]–[25]. In our study, we employ the Differentiable Ensemble Kalman Filter (DEnKF) [17] as the stateful model for accurate arm pose estimation and uncertainty measurement. By utilizing both a smartwatch and a smartphone, our system enables users to have greater freedom of movement. The DEnKF is leveraged as the inference model, ensuring stable and precise pose estimations throughout our experiments.

III. Methodology

This section describes data collection, defines states, observations, and the implementation details of our DEKnF for human pose state estimations.

A. Data Collection, Observation, and State

For data collection, we developed smartwatch (Wear OS) and smartphone (Android) apps to stream sensor data, and made use of the optical motion capture system Optitrack [5] for capturing ground truth. As depicted on the left in Figure 2, the human subjects wore a 25-marker Optitrack suit, a smartwatch on their left wrist, and kept a smartphone in their pocket. After starting the data recording with watch, phone, and Optitrack, we asked subjects to perform random arm motions and encouraged them to vary their body orientation or move around within the optically trackable area. Recorded

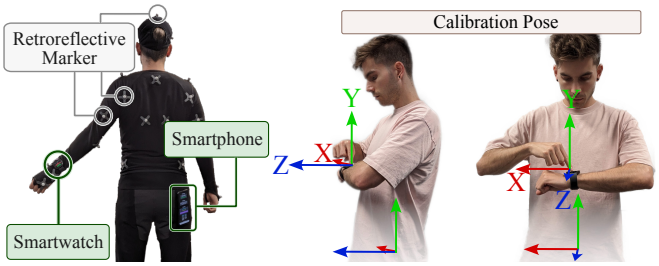

Fig. 2. **Left**: Data collection to record ground truth OptiTrack poses together with smartwatch and smartphone sensor data. **Right**: Starting our app on the watch calibrates the watch and phone orientation. The start pose sets the local coordinate system and defines the forward direction.

data comprises of calibrated observation data from watch and phone together with ground-truth state data from OptiTrack. We define calibrated observations and states in the following.

Observation: The raw observation $\mathbf{y}$ consists of the values $\mathbf{y} = [\Delta t, \boldsymbol{\theta}_{\text{sw}}, \boldsymbol{v}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \rho, \mathbf{r}_{\text{h}}]^\top$, with $\mathbf{y} \in \mathbb{R}^{22}$. The following values are measured by the smartwatch: The virtual rotation vector sensor ($\boldsymbol{\theta}$) by Wear OS provides a global orientation with respect to the magnetic North and gravity vector. We transform the provided quaternion into the form of a continuous six-dimensional rotation representation (6DRR), which is well-suited for training neural networks [26]. Therefore, $\boldsymbol{\theta}_{\text{sw}} \in \mathbb{R}^6$. The measurements $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\phi} \in \mathbb{R}^3$ are the IMU readings. Further, we integrate linear acceleration measurements between two observations over Δt and denote them as velocities $\boldsymbol{v} \in \mathbb{R}^3$. The value ρ is the atmospheric pressure sensor. Finally, $\mathbf{r}_{\text{h}}$ is measured by the smartphone: This measurement is the rotation around the calibrated up-axis of the virtual rotation vector sensor provided by Android. To turn this rotation into a continuous rotation representation, we estimate $\mathbf{r}_{\text{h}}$ as the sine and cosine of the rotation angle. Therefore, $\mathbf{r}_{\text{h}} \in \mathbb{R}^2$.

Calibration: We require a calibration to bring the virtual rotation sensors of watch and phone into the reference frame of the human body, and to define forward and up directions. For a seamless calibration procedure, we propose the start pose as depicted in Figure 2. When the user starts the app on the smartwatch, they hold their lower arm parallel to the chest and hip. We then record subsequent watch orientation measurements $\boldsymbol{\theta}_{\text{sw}}$ and phone orientation measurements $\mathbf{r}_{\text{h}}$ relative to their orientations in this start pose. Further, because our simulations run in the Unity engine, we transform them into a right-handed coordinate system with the Y-axis up and Z-axis forward. In addition to the orientations, we also calibrate atmospheric pressure ρ by recording subsequent measurements relative to the initial measurement in start pose. Therefore, calibrated $\boldsymbol{\theta}_{\text{sw}}$, $\mathbf{r}_{\text{h}}$, and ρ measurements are recorded relative to their initial values in the initial calibration pose when the user started streaming.

State: Given our calibrated observations, we define state as a vector holding the arm pose and forward-facing direction of the human and their velocities. We utilize upper arm rotation ($\mathbf{q}_{\text{u}}$) and lower arm rotation ($\mathbf{q}_{\text{l}}$) together with their velocities $\dot{\mathbf{q}}_{\text{u}}$ and $\dot{\mathbf{q}}_{\text{l}}$ in the continu-

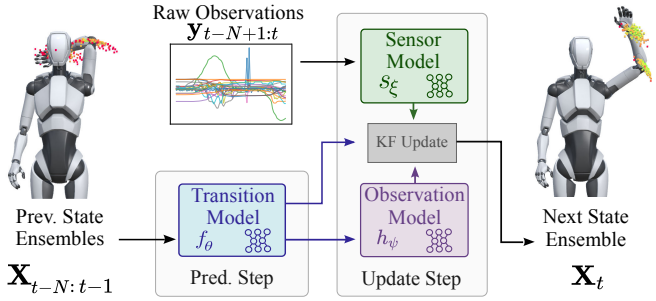

Fig. 3. The DEnKF model structure. In the Prediction Step, the stochastic Transition Model forwards the ensemble $\mathbf{X}_{t-N:t-1}$ one step in time ($\tilde{\mathbf{X}}_t$). In the following Update Step, the Sensor Model projects raw observations to the observation space and the Observation Model projects $\tilde{\mathbf{X}}_t$ to the observation space, such that the KF Update corrects $\tilde{\mathbf{X}}_t$ and we obtain $\mathbf{X}_t$.

ous 6DRR. Further, we facilitate the body-forward facing direction as the sine and cosine of the rotation around the body up-axis as $\mathbf{q}_\mathrm{h} \in \mathbb{R}^2$ and the angular velocity as Euler angles $\dot{\mathbf{q}}_\mathrm{h} \in \mathbb{R}$. The entire state is denoted as $\mathbf{x} = [\mathbf{q}_\mathrm{u}, \mathbf{q}_\mathrm{l}, \mathbf{q}_\mathrm{h}, \dot{\mathbf{q}}_\mathrm{u}, \dot{\mathbf{q}}_\mathrm{l}, \dot{\mathbf{q}}_\mathrm{h}]^\top$, where $\mathbf{x} \in \mathbb{R}^{27}$.

B. Differentiable Ensemble Kalman Filter

To model the temporal transition between human pose states $\mathbf{x}$ and to map raw observations $\mathbf{y}$ to the state space, we utilize a Differentiable Ensemble Kalman Filter (DEnKF) [17]. This state estimation approach enables us to learn and infer the dynamics of the human pose over time, while efficiently incorporating and processing the observed data. As depicted in Figure 3, the DEnKF keeps the core algorithmic steps of an Ensemble Kalman Filter (EnKF) [27] while leveraging the capabilities of stochastic neural networks (SNNs) [28]. There are two steps in DEnKF, the *Prediction Step* propagates the state one step further in time, and the *Update Step* corrects the state based on newly collected observations. We define the learned observation for DEnKF as $\tilde{\mathbf{y}} = [\mathbf{q}_\mathrm{u}, \mathbf{q}_\mathrm{l}, \mathbf{q}_\mathrm{h}, \dot{\mathbf{q}}_\mathrm{u}, \dot{\mathbf{q}}_\mathrm{l}, \dot{\mathbf{q}}_\mathrm{h}]^\top$. Let $\mathbf{X}_{0:N}$ denote the states of N steps in t with a number of $E \in \mathbb{Z}^+$ ensemble members, we initialize the filtering process with $\mathbf{X}_{0:N} = [\mathbf{x}^1_{0:N}, \ldots, \mathbf{x}^E_{0:N}]$.

Prediction Step: This step is modeled using the Transition Model as shown in Figure 3. We use a window of N and we leverage the stochastic forward passes from a trained state transition model to update each ensemble member:

$$\tilde{\mathbf{x}}^i_t \sim f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}^i_t | \mathbf{x}^i_{t-N:t-1}), \ \forall i \in E. \tag{1}$$

Matrix $\tilde{\mathbf{X}}_t = [\tilde{\mathbf{x}}^1_t, \cdots, \tilde{\mathbf{x}}^E_t]$ holds the updated ensemble members which are propagated one step forward through the state space. Note that sampling from the transition model $f_{\boldsymbol{\theta}}(\cdot)$ implicitly introduces a process noise.

Update Step: Given the updated ensemble members $\tilde{\mathbf{X}}_t$, a nonlinear observation model $h_{\boldsymbol{\psi}}(\cdot)$ is applied to transform the ensemble members from the state space to observation space. The observation model is realized via a neural network with weights $\boldsymbol{\psi}$:

$$\mathbf{H}_t\tilde{\mathbf{X}}_t = \left[h_{\boldsymbol{\psi}}(\tilde{\mathbf{x}}^1_t), \cdots, h_{\boldsymbol{\psi}}(\tilde{\mathbf{x}}^E_t)\right], \tag{2}$$
$$\mathbf{H}_t\mathbf{A}_t = \mathbf{H}_t\tilde{\mathbf{X}}_t - \left[\frac{1}{E}\sum_{i=1}^{E} h_{\boldsymbol{\psi}}(\tilde{\mathbf{x}}^i_t), \cdots, \frac{1}{E}\sum_{i=1}^{E} h_{\boldsymbol{\psi}}(\tilde{\mathbf{x}}^i_t)\right].$$

$\mathbf{H}_t\tilde{\mathbf{X}}_t$ is the predicted observation, and $\mathbf{H}_t\mathbf{A}_t$ is the sample mean of the predicted observation at t. EnKF treats observations as random variables [27].

As shown in Figure 3, we incorporate a Sensor Model that can learn projections between the learned observation and raw observation space. To this end, we leverage the methodology of SNN to train a stochastic sensor model that takes N steps of the raw observation and predicts the current learned observation using $s_{\boldsymbol{\xi}}(\cdot)$:

$$\tilde{\mathbf{y}}^i_t \sim s_{\boldsymbol{\xi}}(\tilde{\mathbf{y}}^i_t | \mathbf{y}_{t-N+1:t}), \ \forall i \in E, \tag{3}$$

where $\mathbf{y}_{t-N+1:t}$ represents the noisy observations. Sampling yields observations $\tilde{\mathbf{Y}}_t = [\tilde{\mathbf{y}}^1_t, \cdots, \tilde{\mathbf{y}}^E_t]$ and sample mean $\bar{\mathbf{y}}_t = \frac{1}{E}\sum_{i=1}^{E} \tilde{\mathbf{y}}^i_t$.

DEnFK then proceeds with the Kalman Filter Update (KF Update in Figure 3). To this end, the innovation covariance $\mathbf{S}_t$ can then be calculated as:

$$\mathbf{S}_t = \frac{1}{E-1}(\mathbf{H}_t\mathbf{A}_t)(\mathbf{H}_t\mathbf{A}_t)^T + r_{\boldsymbol{\zeta}}(\tilde{\mathbf{y}}_t), \tag{4}$$

where $r_{\boldsymbol{\zeta}}(\cdot)$ is the measurement noise model implemented using MLP. We use the same way to model the observation noise as in [16], $r_{\boldsymbol{\zeta}}(\cdot)$ takes a learned observation $\tilde{\mathbf{y}}_t$ in time t and provides stochastic noise in the observation space by constructing the diagonal of the noise covariance matrix. The final estimate of the ensemble $\mathbf{X}_t$ can be obtained by performing the measurement update step:

$$\mathbf{A}_t = \tilde{\mathbf{X}}_t - \frac{1}{E}\sum_{i=1}^{E}\tilde{\mathbf{x}}^i_t, \ \mathbf{K}_t = \frac{1}{E-1}\mathbf{A}_t(\mathbf{H}_t\mathbf{A}_t)^T\mathbf{S}_t^{-1}, \tag{5}$$
$$\mathbf{X}_t = \tilde{\mathbf{X}}_t + \mathbf{K}_t(\tilde{\mathbf{Y}}_t - \mathbf{H}_t\tilde{\mathbf{X}}_t),$$

where $\mathbf{K}_t$ is the Kalman gain. In inference, the ensemble mean $\bar{\mathbf{x}}_t = \frac{1}{E}\sum_{i=1}^{E}\mathbf{x}^i_t$ is used as the updated state. The neural network structures for all learnable modules are described in Table I, where the ReLU activation function is applied within all hidden layers and a liner activation function is used for the output layer in each sub-module.

TABLE I
DENKF LEARNABLE SUB-MODULES.

$f_{\boldsymbol{\theta}}$:	1×SNN(256, ReLU), 1×SNN(512, ReLU), 1×fc(S, -)
$h_{\boldsymbol{\psi}}$:	2×fc(32, ReLU), 2×fc(64, ReLU), 1× fc(O, -)
$r_{\boldsymbol{\zeta}}$:	2×fc(16, ReLU), 1×fc(O, -)
$s_{\boldsymbol{\xi}}$:	2×SNN(256, ReLU), 2×SNN(64, ReLU), 1×fc(O, -)

fc: fully connected, S, O: state and observation dimension.

Training: We train the entire framework in an end-to-end manner using a mean squared error (MSE) loss between the ground truth state $\hat{\mathbf{x}}_t$ and the estimated state $\bar{\mathbf{x}}_t$ at every timestep. We also supervise the intermediate modules via loss gradients $\mathcal{L}_{f_{\boldsymbol{\theta}}}$ and $\mathcal{L}_{s_{\boldsymbol{\xi}}}$. Given ground truth at time t, we

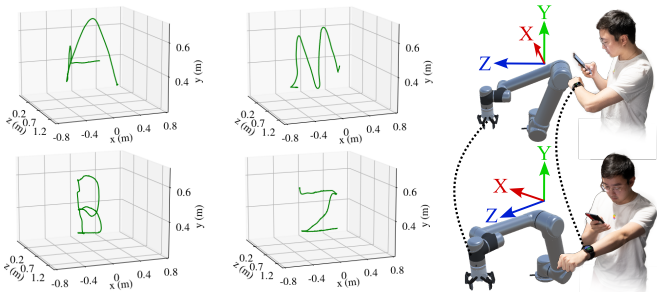

Fig. 4. We propose a control modality where the body forward-facing direction together with the estimated wrist position on the sagittal plane determine the final end-effector position relative to the robot's base. This allows fine-grained control as demonstrated by writing the letters A, M, B, Z with the robot end-effector.

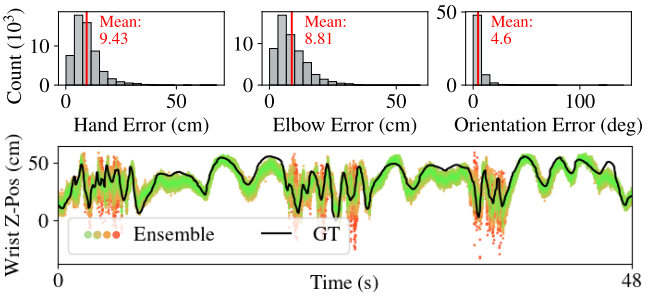

Fig. 5. **Top:** Prediction error distributions of DEnKF on the test dataset. **Bottom:** An example for widening ensemble distributions, i.e., higher uncertainty, when the user moves fast. Ensemble members are colored according to their distance from the mean. GT is the Ground Truth.

apply the MSE loss gradient calculated between $\hat{\mathbf{x}}_t$ and the output of the state transition model to $f_{\boldsymbol{\theta}}$ as in Eq. 6. We apply the intermediate loss gradients computed based on the ground truth observation $\hat{\mathbf{y}}_t$ and the output of the stochastic sensor model $\tilde{\mathbf{y}}_t$:

$$\mathcal{L}_{f_{\boldsymbol{\theta}}} = \|f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_{t-N:t-1}) - \hat{\mathbf{x}}_t\|_2^2, \quad \mathcal{L}_{s_{\boldsymbol{\xi}}} = \|\tilde{\mathbf{y}}_t - \hat{\mathbf{y}}_t\|_2^2. \quad (6)$$

All models in the experiments were trained for 50 epochs with batch size 256, a learning rate of $\eta = 10^{-4}$, and an ensemble size of 32.

C. Control Modality

We introduce a control modality to leverage human pose estimations for fine-grained robot control. As depicted in Figure 4, we utilize the predicted body forward-facing direction $\mathbf{q}_\mathrm{h}$ to set the local coordinate system for the user. The target robot end-effector position then is the projected wrist position on the sagittal plane (defined by the Z-axis and Y-axis). On the left in Figure 4, two human subjects used this modality to guide the end-effector of a Universal Robot 5 (UR5) within a $1.6\,\mathrm{m} \times 0.6\,\mathrm{m} \times 1\,\mathrm{m}$ area to write letters. The trajectories of the end-effector were recorded with OptiTrack. This completes our iRoCo system: the DEnKF algorithm to track the human pose from smartwatch and smartphone data together with the established modality for fine-grained robot control.

IV. Evaluation

This section discusses the evaluation of the pose estimation accuracy of the DEnKF algorithm and outlines the composition of our training and test datasets.

A. Training and Test Datasets

Training and test data was collected using the methodology described in Section III-A and was approved by the institutional review board (IRB) of ASU under the ID STUDY00017558. We recorded data in sessions of approximately 20 min length. In the rare event of magnetometer sensor drift during a session, we asked the human subject to recalibrate the devices and start over. We recorded at least 4 sessions per human subject. Our training dataset consists of 970,493 data points collected from five human subjects. For the test dataset, the participants performed a series of common movements including crossing the arm on the chest and behind the head, waving, and walking in a figure-eight in a separate recording session. Further, we encouraged participants to perform movements typical for teleoperation, e.g., slow wrist movements in a straight line or orientation changes with a constant arm pose.

We augmented training and test data by retrospectively rotating the calibrated smartwatch orientation $\boldsymbol{\theta}_\mathrm{sw}$ and smartphone body orientation $\mathbf{r}_\mathrm{h}$ by a random angle around the up-axis. This augmentation artificially simulated additional body orientations. This was feasible because the remaining sensor measurements, such as the accelerometer or barometer, are in the watch's reference frame and therefore unaffected by global orientation changes. As a result of this augmentation process, our training dataset amounts to 4,259,746 data points and our test dataset amounts to 26,688 data points.

B. Model Performance

To evaluate human pose estimation accuracy on the test dataset we employed Eucledian distance and angular difference metrics. To this end, we processed the DEnKF state ensemble outputs because they quantify the state uncertainty through the distribution of their ensemble members. For example, in Figure 5 the distributions widen when the wrist Z-position changes fast, indicating a lower confidence. To obtain a singular Euclidean distance to the ground truth, we averaged the rotation values of all ensemble members for the lower arm $\mathbf{q}_\mathrm{l}$, upper arm $\mathbf{q}_\mathrm{u}$, and hip $\mathbf{q}_\mathrm{h}$. Assuming a default fixed lower arm length l_l, fixed upper arm length l_u, and fixed distance from hip to shoulder, we utilized forward kinematics to determine the relative XYZ coordinates of the wrist and elbow. This provided us with intuitive Euclidean distances for our evaluation and comparison with related work. Figure 5 summarizes the performance of the DEnKF model on the test dataset. On average, hand positions were off by 9.43 cm, elbow positions by 8.81 cm, and body orientation by 4.6 deg.

Table II compares our results with other related works [12]–[14]. The method of [13] requires inference in the same environment where the training data was collected, therefore, it is not applicable *anywhere*. Further, [13] focuses on pose predictions for the wrist, omitting the rest of the

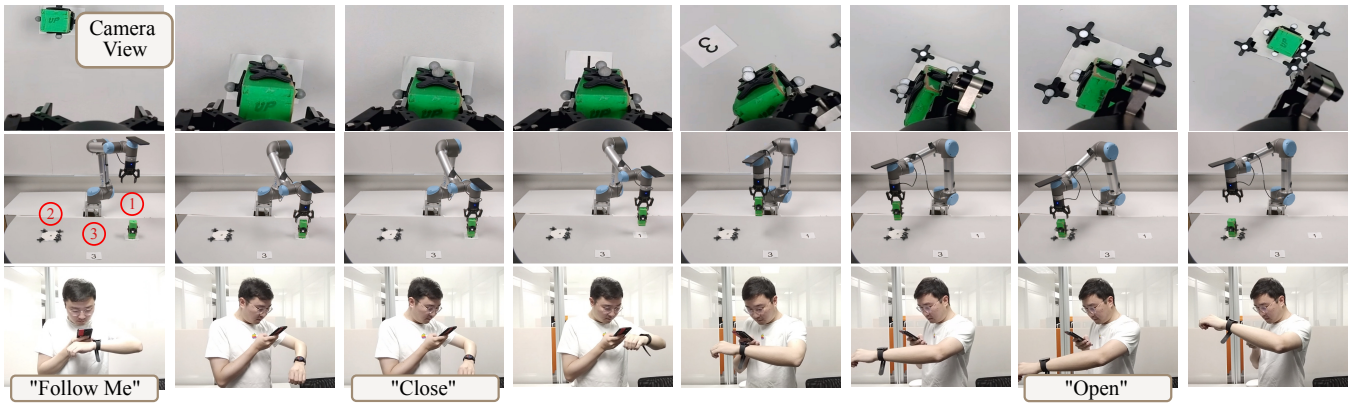

Fig. 6. An example for our teleoperation pick-and-place task. The task had 3 possible locations for cube and target. Human subjects watched the first-person camera view on the phone and controlled the robot through body movements and voice commands.

TABLE II
MODEL COMPARISON WITH RELATED WORK

Method	Anywhere	Free Forward-Facing Dir.	Wrist (cm)	Elbow (cm)	Hip (deg)
[14]	×	✓	10.93	-	-
[13]	✓	×	8.50	8.50	-
[12]	✓	×	9.20	7.90	-
Ours	✓	✓	9.43	8.81	4.60

body pose including elbow or hip orientation. Methods of [14] and [12] demonstrate lower errors for wrist and elbow but fix the user to a constant forward-facing direction. In contrast, iRoCo also provides an estimate of the Hip pose and allows for ubiquitous pose estimation regardless of location or changes in body orientation. Like all related works we compare to, iRoCo is real-time capable. Our framework achieves inference speed at ∼62 Hz on a system using an Intel® Xeon(R) W-2125 CPU and NVIDIA GeForce RTX 2080 Ti.

V. REAL-ROBOT APPLICATIONS

To further assess iRoCo's capabilities to facilitate ubiquitous robot control, we apply it in teleoperation and outdoor drone piloting tasks.

A. Teleoperation

For the teleoperation task, participants remotely controlled a UR5 to pick-and-place a cuboid. We compared the task completion times and placement accuracy with iRoCo and OptiTrack as the baseline. Data collection was approved under the IRB of ASU under the ID STUDY00018521.

Task Setup: The first column of Figure 6 details the task setup. The UR5 had operated on a tabletop surface with three marked locations. We placed a cuboid at one of these locations and at another a platform indicating the placement target. To control the UR5 remotely, the human subject wore a smartwatch and held a smartphone. We initialized a Zoom session between a smartphone mounted on the UR5 and the smartphone held by the human subject. This provided a first-person view. Further, the smartwatch transcribed voice commands for gripper control.

Task Procedure: The entire Figure 6 depicts the procedure of an example pick-and-place teleoperation task. In the beginning, the human subject started the iRoCo motion capture and then instructed the robot to follow their arm movements ("Follow Me"). This triggered the robot to follow the arm motions of the human subject as defined by our control modality (Section III-C). The subject then maneuvered the end-effector toward the cuboid and grabbed it by closing the gripper with a voice command ("Close"). Then, the subject lifted the end-effector and maneuvered it to the target platform. Once the cuboid hovered above the target platform, they opened the gripper with another voice command ("Open").

TABLE III
PICK-AND-PLACE TASK COMPARISON

Method	Time (s)	Dist. (cm)	Success	Fail
Optitrack	59.8 ± 16.5	4.5 ± 2.9	29	1
iRoCo	71.3 ± 25.6	6.3 ± 6.5	28	2

Results: Five human subjects performed twelve pick-and-place tasks each, six with iRoCo and six with Optitrack. For every task, we altered the cuboid and target platform positions (1-2, 1-3, 2-1, ...). Table III summarizes the results. Three tasks failed because the UR5 knocked over the cuboid. If a task succeeded, we measured the time until task completion ("Follow Me" until "Open"). Further, we measured the distance from the placed cube to the center of the target location using OptiTrack. On average, human subjects completed pick-and-place tasks ∼13 s faster with OptiTrack than with iRoCo and placed the cuboid ∼1.8 cm closer to the target. A paired t-test results in a p-value of 0.018 for completion times and 0.161 for distance measurements, indicating a significant difference in execution times but no significant difference in placement accuracy. This indicates that users might expect slightly longer task execution times when using iRoCo but they are able to achieve a similar level of placement accuracy for the task.

Fig. 7. **Left:** The drone piloting task setup. Targets (T_1, T_2, T_3) were marked with AprilTags [29] to be detected through the camera stream of the drone. **Right:** Human subjects piloted the drone to all three targets in a given random order. This figure depicts the control with iRoCo, where the user only had to point toward the target and adjust elevation and distance with their wrist.

B. Drone Piloting

In this drone piloting task, human subjects used iRoCo to fly a commercial Parrot Bebop 2 drone to three target locations. We compared iRoCo control to the original remote named SkyController as the baseline. this application further highlights the *anywhere* aspect of iRoCo, as we conducted it outside and without any specialized infrastructure. The IRB of ASU approved data collection under the ID STUDY00018450.

Task Setup: The left part of Figure 7 depicts an overview of our task setup. Human subjects stood in the middle of a field and had three targets around them. Targets were colored cardboard sheets marked with AprilTags [29] placed at 4 m, 5 m, and 6 m distance. The task was to hover the drone above these targets in a given random order. When the downward-facing camera stream of the drone recognized the AprilTag ID, we considered the target reached and the drone had to fly to the next one.

Task Procedure: The right part of Figure 7 depicts an example task procedure. We assigned the three targets in random order. In this example, the order was T1, T2, and then T3. Using iRoCo, the Human subject controlled the drone through their body orientation and wrist position. This way, motion capture control with iRoCo enabled the subject to simply orient themselves toward a target and adjust with wrist movements while keeping the target and the drone in view. In the bottom right picture of Figure 7 the drone hovers above the final target and the task is completed.

Results: Ten human subjects completed the task with both iRoCo and with the SkyController. We measured the task completion time as the time from reaching the first target until the last. Post-completion, subjects answered a subjective task load index questionnaire (NASA-TLX) [30]. Table IV summarizes our findings. On average, participants completed the task ∼19 s faster with iRoCo than with the original remote. This seems reasonable because motion capture allowed to keep the drone in view, while, with the remote, subjects tended to alternate between checking the display and looking at the drone to keep a mental map of where the drone is in relation to the targets. We highlight three categories of the NASA-TLX questionnaires, which support this assessment. Subjects reported a lower mental load with iRoCo and a considerably lower frustration score with a low standard deviation. However, the iRoCo system scored higher on the physical load index because the subjects had to control with turns and arm movements.

TABLE IV
PILOTING COMPARISON (10 HUMAN SUBJECTS)

	Objective	Subjective Task Load Index		
Method	Time (s)	Mental	Physical	Frustration
Remote	59.7 ± 27.8	7.3 ± 3.6	3.7 ± 2.9	4.5 ± 2.4
iRoCo	40.5 ± 10.1	4.7 ± 4.4	6.1 ± 4.3	2.0 ± 1.5

VI. CONCLUSION

This work introduced iRoCo, an intuitive and ubiquitous system to control robots with a smartwatch and smartphone. By incorporating a differentiable ensemble Kalman filter algorithm, it optimizes the balance between motion capture accuracy and unconstrained user movement. We demonstrated its effectiveness in practical teleoperation and drone piloting tasks with time and distance metrics as well as subjective task load metrics. Results confirm that, with only a smartwatch and a smartphone, iRoCo offers new and intriguing possibilities for ubiquitous robot control and human-robot collaboration.

REFERENCES

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, pp. 957–975, 2018.

[2] I. Patzer and T. Asfour, "Minimal sensor setup in lower limb exoskeletons for motion classification based on multi-modal sensor data," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 8164–8170.

[3] S. Schaal, "Learning from demonstration," in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9. MIT Press, 1996.

[4] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 2831–2837.

[5] G. Nagymáté and R. M. Kiss, "Application of optitrack motion capture systems in human movement analysis: A systematic literature review," *Recent Innovations in Mechatronics*, vol. 5, no. 1., p. 1–9., Jul. 2018.

[6] Y. Desmarais, D. Mottet, P. Slangen, and P. Montesinos, "A review of 3d human pose estimation algorithms for markerless motion capture," *Computer Vision and Image Understanding*, vol. 212, p. 103275, 2021.

[7] T. von Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll, "Sparse inertial poser: Automatic 3d human pose estimation from sparse imus," *CoRR*, vol. abs/1703.08014, 2017.

[8] X. Yi, Y. Zhou, and F. Xu, "Transpose: Real-time 3d human translation and pose estimation with six inertial sensors," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–13, 2021.

[9] D. Roetenberg, H. Luinge, P. Slycke, *et al.*, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, pp. 1–7, 2009.

[10] V. Villani, B. Capelli, C. Secchi, C. Fantuzzi, and L. Sabattini, "Humans interacting with multi-robot systems: a natural affect-based approach," *Autonomous Robots*, vol. 44, pp. 601–616, 2020.

[11] F. C. Weigend, S. Sonawani, M. Drolet, and H. B. Amor, "Anytime, anywhere: Human arm pose from smartwatch data for ubiquitous robot control and teleoperation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3811–3818.

[12] S. Shen, H. Wang, and R. Roy Choudhury, "I am a smartwatch and i can track my user's arm," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 85–96.

[13] W. Wei, K. Kurita, J. Kuang, and A. Gao, "Real-time limb motion tracking with a single imu sensor for physical therapy exercises," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 7152–7157.

[14] M. Liu, S. Yang, W. Chomsin, and W. Du, "Real-time tracking of smartwatch orientation and location by multitask learning," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 120–133.

[15] V. Villani, L. Sabattini, G. Riggio, A. Levratti, C. Secchi, and C. Fantuzzi, "Interacting with a mobile robot with a natural infrastructureless interface," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 753–12 758, 2017.

[16] A. Kloss, G. Martius, and J. Bohg, "How to train your differentiable filter," *Autonomous Robots*, pp. 1–18, 2021.

[17] X. Liu, G. Clark, J. Campbell, Y. Zhou, and H. B. Amor, "Enhancing state estimation in robots: A data-driven approach with differentiable ensemble kalman filters," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1947–1954.

[18] X. Liu, Y. Zhou, S. Ikemoto, and H. B. Amor, "α-mdf: An attention-based multimodal differentiable filter for robot state estimation," in *7th Annual Conference on Robot Learning*, 2023.

[19] B.-G. Lee, B.-L. Lee, and W.-Y. Chung, "Smartwatch-based driver alertness monitoring with wearable motion and physiological sensor," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 6126–6129.

[20] H. Wicaksono, I. Sugiarto, P. Santoso, G. Ricardo, and J. Halim, "Towards autonomous robot application and human pose detection for elders monitoring," in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 2022, pp. 772–776.

[21] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[22] A. Klushyn, R. Kurle, M. Soelch, B. Cseke, and P. van der Smagt, "Latent matters: Learning deep state-space models," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[23] B. Wagstaff, E. Wise, and J. Kelly, "A self-supervised, differentiable kalman filter for uncertainty-aware visual-inertial odometry," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2022, pp. 1388–1395.

[24] M. A. Lee, B. Yi, R. Martín-Martín, S. Savarese, and J. Bohg, "Multimodal sensor fusion with differentiable filters," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 444–10 451.

[25] X. Liu, S. Ikemoto, Y. Yoshimitsu, and H. B. Amor, "Learning soft robot dynamics using differentiable kalman filters and spatio-temporal embeddings," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2550–2557.

[26] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 5738–5746.

[27] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[28] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[29] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[30] M. Feick, N. Kleer, A. Tang, and A. Krüger, "The virtual reality questionnaire toolkit," in *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 68–69.