Probabilistic Differentiable Filters Enable Ubiquitous Robot Control with Smartwatches

Fabian C Weigend, Xiao Liu, Heni Ben Amor

***Abstract*— Ubiquitous robot control and human-robot collaboration using smart devices poses a challenging problem primarily due to strict accuracy requirements and sparse information. This paper presents a novel approach that incorporates a probabilistic differentiable filter, specifically the Differentiable Ensemble Kalman Filter (DEnKF), to facilitate robot control solely using Inertial Measurement Units (IMUs) from a smartwatch and a smartphone. The implemented system is cost-effective and achieves accurate estimation of the human pose state. Experiment results from human-robot handover tasks underscore that smart devices allow versatile and ubiquitous robot control. The code for this paper is available at `github.com/ir-lab/DEnKF` and `github.com/wearable-motion-capture`.**

I. Introduction

Examining the human-robot relationship is a central concern in the field of artificial intelligence and robotics. To facilitate reliable human-robot collaboration, accurate estimations of the state of humans and robots are crucial. One challenge is that numerous robotics systems still rely heavily on costly motion capture systems in estimating the human pose, thereby restricting their suitability primarily to stationary setups or complex calibration procedures.

The gold standard for motion capture are cameras [1]. These systems may feature multiple cameras on multiple base stations or in one device, e.g, Microsoft Kinect v2 [2] or Intel RealSense. These systems usually require a dedicated setup and suffer from line-of-sight issues. Alternatively, non-optical systems enable motion capture through Inertial Measurement Units (IMUs) [3], [4]. For this purpose, the Xsens motion capture system [5] is commonly used [6]. Also, Sony's recent Mocopi [7] promises new opportunities for non-visual real-time human motion capture. A downside of most IMU-based tracking is that they require users to carefully place multiple specialized units on their body and a calibration procedure.

Recent research investigates the opportunities of omnipresent IMUs in smartphones and smartwatches for motion capture. Some even limit themselves to a single device [8], [9]. These approaches have promising potential for human-robot collaboration because motion capture through smart devices is ubiquitous and natural to the user [10]. However, leveraging the data of a single smartwatch for motion capture of sufficient accuracy for robot control is challenging, and previous work had to constrain the user to a constant body forward-facing direction [10].

All Authors are with the School of Computing and Augmented Intelligence, Arizona State University {fweigend, xliu330, hbenamor}@asu.edu

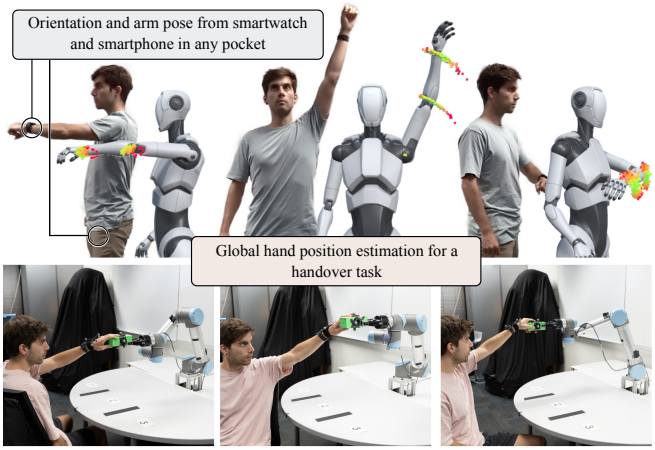

Fig. 1. **Top**: Our Differentiable Ensemble Kalman Filter (DEnKF) achieves robust body orientation and arm pose estimations from the sensor data of a single smartwatch and a smartphone. The user can place the phone in any pocket. **Bottom**: Orientation predictions are also accurate when the user sits. We evaluate pose predictions on test data and in a human-robot collaboration handover task.

This work builds upon [10] and incorporates the sensor data of a connected smartphone to enable ubiquitous robot control without body orientation constraints. Recursive Bayesian filters, particularly Kalman filters, play a pivotal role in tasks such as predicting the future movements of human interaction partners [11], tracking objects over time [12], and ensuring stability during robot locomotion [13]. We propose that advances in state estimation, specifically the Differentiable Ensemble Kalman Filter (DEnKF) [14], allow to enhance the accuracy and stability of motion capture using ubiquitous smart devices.

As depicted in Figure 1, using a probabilistic differentiable filter provides us with a distribution of solutions, aiding stability and adding an important measure of uncertainty for robot control. The proposed filter allows to strike the balance between less-constrained movements while still achieving stable and effective pose estimations suitable for human-robot collaboration.

II. Methodology

We present our approach by describing our data collection and defining our states and observations in Section II-A. Then, we define the used Differentiable Ensemble Kalman Filter (DEKnF) for human pose state estimations in Section II-B.

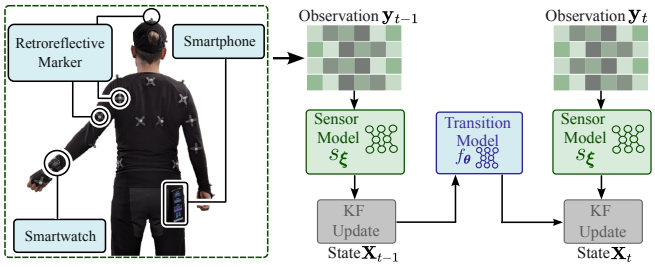

Fig. 2. **Left**: Our data collection setup with smart devices and Optitrack system. **Right**: The DEnKF model structure. The sensor model projects raw observations to the observation space, the stochastic transition model forwards the ensemble one step in t, and the KF update step corrects the state.

A. Data Collection, Observation and State

For data collection, we develop two apps and make use of a research-grade motion capture system. The apps for the smartwatch and smartphone stream sensor data to a remote machine.

Observation: We define that the raw observation $\mathbf{y}$ consists of the following values $\mathbf{y} = [\Delta t, \boldsymbol{\theta}_{\text{sw}}, \boldsymbol{v}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \rho, \mathbf{r}_h]^\top$, with $\mathbf{y} \in \mathbb{R}^{\mathbf{22}}$, where $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\phi} \in \mathbb{R}^3$ are the IMU readings. Namely, these are the average values of the linear acceleration, gravity, and gyroscope since the previous observation Δt seconds ago. The averaging is necessary because the apps stream data at around 80 Hz and the accelerometer and gyroscope record measurements at a faster rate. We also integrate linear acceleration measurements between two observations and denote them as velocities $\boldsymbol{v}$. In addition, the apps record the virtual rotation vector sensor ($\boldsymbol{\theta}$), which is provided by Android and Wear OS. We record this orientation in form of a continuous six-dimensional rotation representation (6DRR), which is well-suited for training neural networks [15].

Calibration: When the user starts the app on their smartwatch, they usually hold their arm up, parallel to the chest and hip. We use this start position to calibrate $\boldsymbol{\theta}$ with the first initial orientations $\boldsymbol{\theta}_{\text{init}}$, which gives us the calibrated orientation $\boldsymbol{\theta}_{\text{sw}}$. The value ρ with $\rho \in \mathbb{R}$ is the atmospheric pressure sensor. We also calibrate this one by subtracting the reading when the user started the app from all subsequent ones. Finally, the phone provides the body-forward facing direction. Also here, we record the orientation sensor of the phone and calibrate it with the first observation in the starting position. Further, we assume that the user has their forward-facing direction parallel to the watch at the starting position. This allows us to estimate an offset rotation from the phone to the forward direction, bringing it into the same global reference frame as the watch. From this, we denote the body orientation as the sine and cosine of the calibrated up-axis rotation of the phone, represented as $\mathbf{r_h} \in \mathbb{R}^{\mathbf{2}}$ in our raw observation $\mathbf{y}$.

State: Our state entails the arm pose and body forward-facing direction of the human. The ground truth values were recorded with the research-grade optical motion capture system OptiTrack [1]. We record data from participants who wore a 25-marker-upper-body suit along with the smartwatch on their left wrist and a smartphone in their pocket. We collect the upper arm rotation ($\mathbf{q}_{\text{u}}$), lower arm rotation ($\mathbf{q}_{\text{l}}$) also in the continuous 6DRR and the body-forward facing direction $\mathbf{r_h}$. Therefore, the ground truth state for motion capture is denoted as $\mathbf{x} = [\mathbf{q}_{\text{l}}, \mathbf{q}_{\text{u}}, \mathbf{r}_h]^\top$, where $\mathbf{x} \in \mathbb{R}^{14}$. In our Bayesian filtering framework, we define the learned observation $\tilde{\mathbf{y}}$ to be equivalent to $\mathbf{x}$.

B. Differentiable Ensemble Kalman Filter

To model the temporal transition of the human pose $\mathbf{x}$ and map raw observations $\mathbf{y}$ to the state space, we utilize DEnKF [14], [16], [17] as shown in Figure 2. 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. We maintain the core algorithmic steps of an Ensemble Kalman Filter (EnKF) [18] while leveraging the capabilities of stochastic neural networks (SNNs) [19]. 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. Let $\mathbf{X}_{0:N}$ denote the states of N steps in t with number of E ensemble members, we initialize the filtering process with $\mathbf{X}_{0:N} = [\mathbf{x}^1_{0:N}, \ldots, \mathbf{x}^E_{0:N}]$, where $E \in \mathbb{Z}^+$.

Prediction Step: In this step, the Transition Model takes the previous states and predicts the next state. We use a window of N and we leverage the stochastic forward passes from a trained state transition model to update each ensemble member:

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

Matrix $\mathbf{X}_t = [\mathbf{x}^1_t, \cdots, \mathbf{x}^E_t]$ 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 $\mathbf{X}_t$, 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 $\boldsymbol{\psi}$:

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

$\mathbf{H}_t\mathbf{X}_t$ 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. Hence, the ensemble can incorporate a measurement perturbed by a small stochastic noise thereby accurately reflecting the error covariance of the best state estimate [18]. As shown in Figure 2, 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 $s_{\boldsymbol{\xi}}(\cdot)$:

$$\tilde{\mathbf{y}}^i_t \sim s_{\boldsymbol{\xi}}(\tilde{\mathbf{y}}^i_t | \mathbf{y}_t), \ \forall i \in E, \tag{3}$$

where $\mathbf{y}_t$ represents the noisy observation. Sampling yields observations $\tilde{\mathbf{Y}}_t = [\tilde{\mathbf{y}}_t^1, \cdots, \tilde{\mathbf{y}}_t^E]$ and sample mean $\tilde{\mathbf{y}}_t = \frac{1}{E}\sum_{i=1}^{i} \tilde{\mathbf{y}}_t^i$. 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 [20], $r_{\boldsymbol{\zeta}}(\cdot)$ takes an 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 $\mathbf{X}_{t|t}$ can be obtained by performing the measurement update step:

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

where $\mathbf{K}_t$ is the Kalman gain. In inference, the ensemble mean $\bar{\mathbf{x}}_{t|t} = \frac{1}{E}\sum_{i=1}^{E}\mathbf{x}_{t|t}^i$ is used as the updated state.

Prediction Targets: Once the estimated state, comprising the rotation values for both the lower arm and upper arm, is obtained, we utilize forward kinematics with fixed lower arm length l_l and fixed upper arm length l_u to determine the corresponding Cartesian XYZ coordinates of the wrist. This is done due to the constraint of a fixed sagittal plane orientation $\mathbf{r}_h$, where the elbow position is limited to a sphere around the shoulder with a radius of l_u. Additionally, the wrist position must lie on a manifold defined by spheres with a radius of l_l around all possible elbow positions, as described in prior work [21].

Training: DEnKF contains four sub-modules: a state transition model, an observation model, an observation noise model, and a sensor model. The entire framework is trained in an end-to-end manner via a mean squared error (MSE) loss between the ground truth state $\hat{\mathbf{x}}_{t|t}$ and the estimated state $\bar{\mathbf{x}}_{t|t}$ 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 apply the MSE loss gradient calculated between $\hat{\mathbf{x}}_{t|t}$ 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 $\tilde{\mathbf{y}}_t$:

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

All models in the experiments were trained for 50 epochs with batch size 256, and a learning rate of $\eta = 10^{-5}$. We chose the model with the best performance on a validation set for testing. The ensemble size of the DEnKF was set to 32 ensemble members.

III. Evaluation

We discuss the validation accuracy on a separate test dataset. Secondly, we assess the performance of our model by applying it in a human-robot handover tasks with a real UR5 robot.

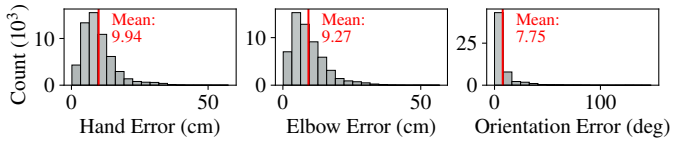

Fig. 3. Prediction error distributions of our DEnKF on the test dataset.

A. Evaluation on Dataset

The training dataset comprises data collected from five human subjects. Written informed consent was obtained and approved by the institutional review board (IRB) of ASU under the ID STUDY00017558. Each subject was instructed to wear the smartwatch on their left arm, the smartphone in one of their pockets and a 25-marker Optitrack suit while performing free random arm motions. The subjects were also encouraged to change their body forward-facing direction and to move around in the area covered by the optical tracking system. In total, we gathered a dataset of 970,493 data points. To further augment the dataset, we artificially adjusted the calibrated smartwatch and smartphone data to simulate new body orientations. This is possible because remaining sensor measurements, e.g., accelerometer or barometer, are in the reference frame of the watch. In total, our augmented training dataset amounts to 4,259,746 data points.

We evaluate the DEnKF prediction accuracy on a test dataset completely separate from the training process. The test dataset comprises 26,688 observations. The participants were asked to perform a series of movements, including arm swing, arm cross on chest and behind the head, arm raise, waving, boxing, clapping, walking in a figure-eight pattern, and jogging in a circular path. Figure 3 summarizes the performance of the DEnKF model on the test dataset. On average, hand positions are off by 9.94 cm, elbow positions by 9.27 cm, and body orientation by 7.75 deg. The DEnKF quantifies the state uncertainty through the distribution of ensemble members, for example, depicted at the top in Figure 1. 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.

TABLE I
Model Comparison with Related Work

Method	Anywhere	Free Forward-Facing Dir.	Wrist (cm)	Elbow (cm)	Hip (deg)
[9]	×	✓	10.93	-	-
[22]	✓	×	8.50	8.50	-
[21]	✓	×	9.20	7.90	-
Ours	✓	✓	9.94	9.27	7.75

We compare our results with other related works [9], [21], [22] in Table I. The method of [22] requires inference in the same environment where the training data was collected, therefore, it is not applicable *anywhere*. Further, [22] focuses on pose predictions for the wrist, omitting the rest of the body pose including elbow or hip orientation. Methods of [9] and [21] demonstrate lower errors for wrist and elbow but fix the

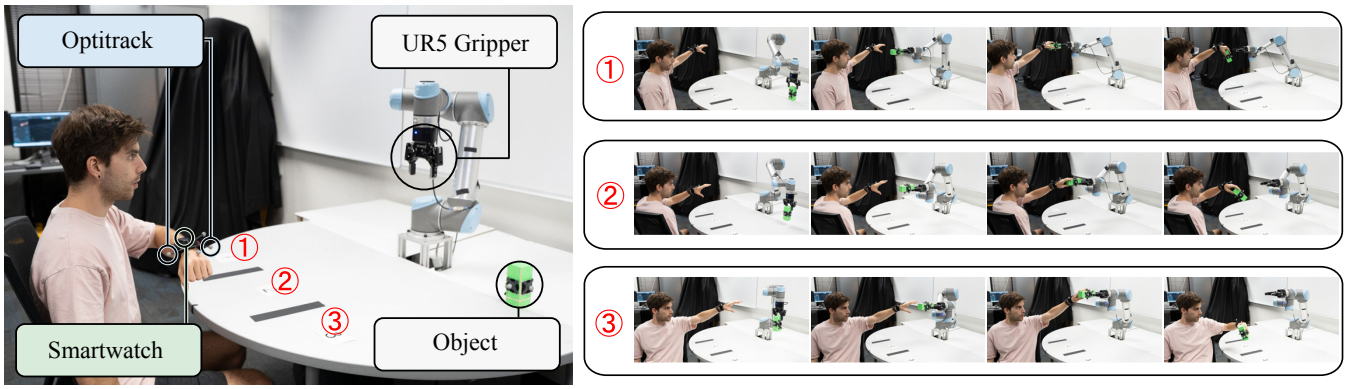

Fig. 4. The handover task system setup is illustrated on the **left**, showcasing the workspace divided into three distinct labeled areas. On the **right**, picture sequences of the handover results are presented for each of these designated zones.

user to a constant forward-facing direction. In contrast, our proposed method using the DEnKF also provides an estimate of the Hip pose and allows for ubiquitous pose estimation regardless of location or changes in body orientation.

B. Handover Task

We demonstrate the efficacy of the trained model in a human-robot handover task. As depicted in Figure 4, participants sit in a chair and engage in handover interactions with the robot. Participants are free to rotate on the chair to experiment with various handover scenarios. The smartphone in their pocket and the smartwatch on their wrist enable estimating the body orientation and arm pose to extract the global hand position.

Task Setup: Participants perform six handover interactions. Each handover interaction is treated as an individual task. At the onset of each task, the participant holds their hand as shown on the left in Figure 4 and initiates the task by issuing a voice command. Like in the work of [10], the smartwatch recognizes the voice command and triggers the robot. Subsequently, the robot grasps the green cube and moves it towards the tracked hand position of the participant. When the cube reaches close proximity to the hand, the participant says "give me the cube" and closes their hand around it. In response to this voice command, the robot releases its grip on the cube, signifying the successful completion of the handover.

To evaluate a range of handover positions, we divide the table surface into three distinct areas labeled as 1, 2, and 3. Each participant is instructed to perform two handovers in each area, with one at a high position and one at a low position. In total, every participant performs all six handovers utilizing smartwatch and smartphone tracking, along with an additional six handovers conducted with the gold-standard Optitrack for our baseline comparison. The order of the tasks and tracking modes is randomized to eliminate any potential biases.

Results: The handover experiment utilizing smartwatches is depicted in Figure 4 (right), illustrating the handover process conducted in zones 1, 2, and 3. We collected data from five human subjects, with each subject performing a total of 12 handover tasks. The evaluation metrics in Table II include the task completion time and the actual 3D distance between hand and cube when the participant gave the voice command to hand over the cube. As summarized in Table II, we observe a relatively small disparity in the task completion time, with an average difference of 1.88 seconds. Further, observed handover 3D distances exhibit a minimal average difference of 0.03 cm when comparing both methods. Altogether, the results suggest that handover tasks with smartwatch and smartphone tracking might take about 1.88 seconds longer but the participant is comfortable to complete the handover at similar distances. It is important to note that all handover tasks were accomplished successfully, with users consistently grasping the cube within the desired zones without any instances of dropping it. The obtained results indicate that the smartwatch system is comparable to the Optitrack system for this task, thus establishing its potential as a cost-efficient alternative.

TABLE II
HANDOVER TASK RESULTS

Method	Time (s)	Dist. (cm)
Wearable	11.74 ± 5.35	7.74 ± 4.68
Optitrack	9.86 ± 2.67	7.77 ± 4.71
Difference	1.88 ± 4.83	0.03 ± 7.13

IV. Conclusion

This work introduces the integration of a ubiquitous robot control system by leveraging smart devices and employing differentiable filters. The experimental results demonstrate that the proposed framework effectively addresses the estimation challenges related to human arm pose estimation, especially in scenarios involving diverse human hip orientations. Additionally, the results from the human-robot handover task showcase that the proposed system achieves comparable error metrics, highlighting its effectiveness. With no additional user instrumentation, the proposed framework offers new and intriguing possibilities for low-cost robot control and human-robot collaboration applications.

REFERENCES

[1] 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.

[2] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3d human pose estimation in rgbd images for robotic task learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1986–1992.

[3] 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.

[4] 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.

[5] "Movella xsens motion capture," https://www.xsens.com/motion-capture/, accessed: 2023-08-24.

[6] 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.

[7] "Sony mocopi 3d motion capture system," https://www.sony.jp/mocopi/, accessed: 2023-08-24.

[8] 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.

[9] 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.

[10] 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," *arXiv preprint arXiv:2306.13192*, 2023.

[11] L. Wang, G. Wang, S. Jia, A. Turner, and S. Ratchev, "Imitation learning for coordinated human–robot collaboration based on hidden state-space models," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102310, 2022.

[12] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on industrial electronics*, vol. 59, no. 11, pp. 4409–4420, 2011.

[13] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a cassie bipedal robot," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2589–2595.

[14] 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," *arXiv preprint arXiv:2308.09870*, 2023.

[15] 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.

[16] X. Liu, S. Ikemoto, Y. Yoshimitsu, and H. B. Amor, "Learning soft robot dynamics using differentiable kalman filters and spatio-temporal embeddings," *arXiv preprint arXiv:2308.09868*, 2023.

[17] 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.

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

[19] 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.

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

[21] 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.

[22] 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.