# A high-accuracy, low-budget Sensor Glove for Trajectory Model Learning

Robin Denz[1,3], Rabia Demirci[1,3], M. Ege Cansev[2], Adna Bliek[2], Philipp Beckerle[2],
Elmar Rueckert[1] and Nils Rottmann[1]

***Abstract*— Sensor gloves are gaining importance in tracking hand and finger movements in virtual reality applications as well as in scientific research. They introduce an unrestricted way of capturing motion without the dependence on direct line of sight as for visual tracking systems. With such sensor gloves, data of complex motion tasks can be recorded and used for modeling probabilistic trajectories or teleoperation of robotic arms. While a multitude of sensor glove designs relying on different functional principles exist, these approaches require either sensitive calibration and sensor fusion methods or complex manufacturing processes. In this paper, we propose a low-budget, yet accurate sensor glove system that uses flex sensors for fast and efficient motion tracking. We evaluate the performance of our sensor glove, such as accuracy and latency, and demonstrate the functionality by recording motion data for learning probabilistic movement models.**

## I. INTRODUCTION

In times of virtual reality (VR) and augmented reality (AR) applications, motion tracking of human body parts gains more importance, especially in the field of robotics, the precise tracking of hand and finger movements is of high interest. The tracked data can be used for trajectory model learning of complex tasks or for real world applications with robotic arms like human-robot collaboration.

The tracking and display of the user's field of view is a mature technique and available for the mainstream consumer. However, the availability of tracking extremities like hands is lagging far behind and is mostly exclusive for high-budget products. For the former, VR goggles use internal sensors and can therefore be used anywhere and without external setup. Meanwhile, the tracking of extremities is restricted to external optical sensor setups, e.g. Valve Index [1], Optitrack [2]. These setups rely on optical markers that are worn by the user, in combination with multiple, precisely calibrated cameras to capture the relative poses of the trackers in the setup space. Thus, those systems are not very versatile, since they require a dedicated setup of the working space as well as equipping the user with external markers. Furthermore, these setups incur a high expense, especially when a high accuracy and the tracking of smaller body parts, like fingers, is required.

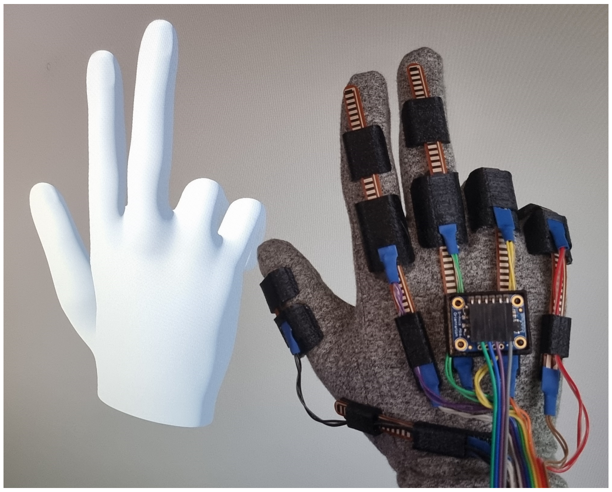


Fig. 1: Example of the here developed sensor glove together with an illustration using Unity.

Other approaches for tracking finger motions are for example realized by a camera attached to the VR goggles, which points down towards the hands of the user. Devices such as Leap Motion [3] abstract the position of the bones in each finger to display the three dimensional model of the hand including the finger joints. However, this still restricts the working area of the sensor to its line of sight.

Since all of these strategies rely on the line of sight of the used sensor and therefore heavily confine the working space, the optimal approach of a user-worn device should use gloves that are equipped with multiple sensors to track the movement of the joints in each finger. Thereby, it is important not to restrict the user in his/her movements, thus to keep the user's experience as unaffected as possible. In such settings, capturing motion data on real, complex tasks is possible without restrictions. The resulting data can then be processed to be applied on robotic arms as well as in virtual environments.

The goal of this project is to design a versatile, while low-budget sensor glove for modeling and comparing different hand movements. The measured data from the sensor glove is later used to learn probabilistic movement primitive models. These learned models can be used for finding similarities

[1] Institute for Robotics and Cognitive Systems, University of Luebeck, Ratzeburger Allee 160, 23562 Luebeck, Germany

[2] Chair of Autonomous Systems and Mechatronics, Department of Electrical Engineering, Friedrich-Alexander-University Erlangen-Nürnberg, Germany.

[3] These authors contributed equally to this work.

Correspondence to rottmann@rob.uni-luebeck.de

between the recordings.

### A. Related Work

Considering a variety of sensor gloves, they can be divided into different groups, depending on the sensors used for hand motion detection. Every approach has its benefits for varying fields of applications, as well as dealing with various challenges. In the following, we will take a look at different sensor glove design concepts.

In Lin et al. [4], multiple inertial measurement units (IMU) were mounted on each finger to calculate the hand posture by looking at the difference of each bone in the finger in relation to the others. With their design, it is possible to capture multiple degrees of freedom for each joint and a very extensive representation of the whole hand. Nonetheless, the rigid and bulky sensors might constrain complex hand movements. Additionally, the calibration method and sensor fusion algorithm for the IMU's raw data can be difficult to adjust in order to achieve a high accuracy.

Other approaches rely on soft materials ranging from capacitive silicone arrays [5] to conductive liquid for soft-strain sensors embedded in a glove [6], [7], [8]. Those soft sensor gloves have one major advantage over other methods: The thin and flexible gloves fit the hand better and therefore allow complex hand movements with minimal constraints. However, the manufacturing of soft sensors requires elevated techniques and equipment for printing liquid metal into soft silicone [9]. Consequently, it is a complex process to construct the sensor glove, which is not straightforward to replicate.

In [10] and [11], linear potentiometers with flexible wires or optical linear encoders are used to measure finger movements. Both approaches lead to reasonably accurate measurements. However, the natural finger motion is restricted by the sensors on the glove. These sensor gloves only measure the flexion and not the abduction/adduction of fingers, which restricts the detection of complex hand motion. Furthermore, the sensor design is not suitable for additional sensor of this type due to the limited space on a hand. Thus, such type of sensor gloves only allows the tracking of simple motions.

Lastly, one of the most intuitive approaches is the usage of flex sensors. One commercially successful sensor glove using such flex sensors is the CyberGlove [12]. It has 18 flex sensors embedded in a fabric glove to measure flexion as well as abduction/adduction which allows the measurement of complex finger motion. However, the high-cost, non open-source solution makes it unavailable for the mainstream consumer. Nevertheless, there are multiple low-cost implementations of such a sensor glove [13], [14]. Both use flex sensors sewn into a fabric glove, whereas [14] also places flex sensors between each finger to capture abduction/adduction. They are able to carry out accurate measurements and also to successfully teleoperate a robot gripper [13]. However, Gentner et al. [14] require modifications of the flex sensors and calibration method in order to reproduce its results. This leads to more complex amplifier circuits for reading the flex sensors. In addition, both implementations do not support WiFi communication between microcontroller and computer.

For learning suitable trajectory models from recorded sensor glove data, the requirements of such models include the ability to model time-depending multi-dimensional data that is affected by sensor noise, missing sensor values and human motion variability [15], [16]. The latter property is inherent to human motion and is a result of the fact that the same task can be solved in numerous ways [17]. This is commonly discussed as Bernstein's redundancy problem, which is not further discussed in this paper.

For learning such trajectory models, there exist a number of potential candidate that fulfill the requirements [18]–[20]. Our choice was motivated by the popularity of the model in the robotics community, the availability of many code resources [21], [22], and that the model was applied already in related work for human motion modeling [13], [23].

### B. Contribution & Paper Organization

Our aim is to construct a simple, budget-friendly, yet accurate sensor glove with low latency, suited for recording hand motion data to learn trajectory models. Most sensor glove implementations require elevated techniques for construction and are not suited for easy and low-cost replication. Our contribution here is threefold: (1) an open-source realization of an intuitive sensor glove which keeps up with the state-of-the-art hand motion capture techniques, in which we payed attention to the user experience by avoiding bulkiness and movement confining parts in the glove design, since the sensor is ultimately developed for transfer learning and teleoperation, (2) a ROS (Robot Operating System [24]) interface for realizing wireless and straight forward access to the sensory data and complex visualization in Unity [25] for direct visual feedback, and (3) a comparison between our proposed sensor glove design and existing approaches together with a functionality proof for trajectory model learning.

We start with Section II by giving insights into the hardware design and software realization of the sensor glove, as well as explaining the calibration methods and the underlying processes of trajectory model learning. We proceed in Section III by evaluating the accuracy and latency of our implementation in comparison to the provided related work. Additionally, we will test the sensor glove in different experimental setups using music instruments. In Section IV we conclude our findings and give an outlook on future work regarding our sensor glove.

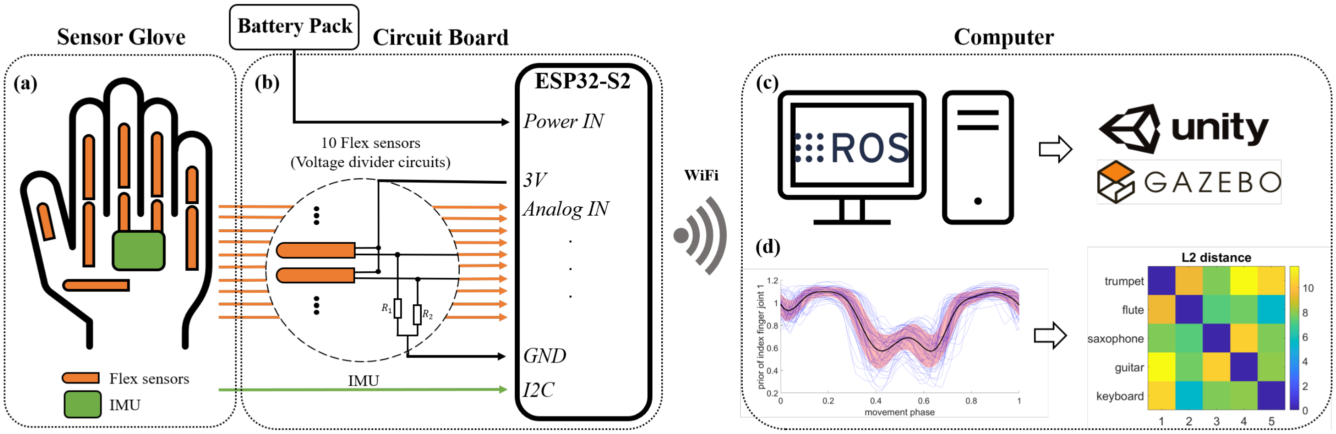


Fig. 2: Simplified schematic diagram of the system architecture for our sensor glove design: (a) Glove layout with sensor placements, the orange fields denote the flex sensors, while the IMU is marked as a green rectangle, (b) Circuit board which is wired with the sensor glove, has 10 voltage dividers for reading each flex sensor connected to ADC pins of the microcontoller ESP32-S2 and the IMU is connected to I2C pins, (c) The ESP32-S2 sends the raw data via WiFi as ROS messages to the computer, which allows a real-time visualization in Unity or Gazebo, (d) Post-processing of the recorded data, e.g. learning probabilistic movement models and searching for similarities

## II. METHODS

In this section, we present the methods we used to design and build our sensor glove as well as the trajectory learning algorithms for generating probabilistic motion models.

### A. *Sensor glove design*

For easy donning and doffing, the sensors and further components are mounted on a fabric glove. To achieve a highly versatile glove that fits different users and does not disturb the users by being too tight or by having excess material especially on the finger tips, we chose to use special soft fabric gloves made for osteoarthritis patients. These gloves are highly stretchable and therefore adjust to many hand types and sizes, while being very pleasant to wear.

We used 10 flex sensors in total [26], [27], where two of them are smaller to fit the thumb and pinkie finger (see Figure 3). The reading of each sensor requires a voltage divider circuit, illustrated in Figure 2(b). Since we use two different types of flex sensors, two different resistors are required. We chose 47k$\Omega$ resistors for the long flex sensors and 10k$\Omega$ for the smaller ones to ensure the voltage change lies between the range of the input pins. As shown in Figure 3, our sensor glove measures the flexion of the proximal interphalangeal (PIP) and metacarpophalangeal (MCP) joints for each finger in addition to the interphalangeal (IP) and carpometacarpal (CMC) joints of the thumb. To mount the sensors on the glove, we designed and printed custom 3D pieces shown in Figure 4, out of slightly flexible filament. First, the sensors were fixed on these pieces and afterwards glued onto the glove. The flex sensors were secured in place through this construction, while allowing the fingers to move without constraints. This enables a consistent bending of the sensors to maintain the precision of our measurements.

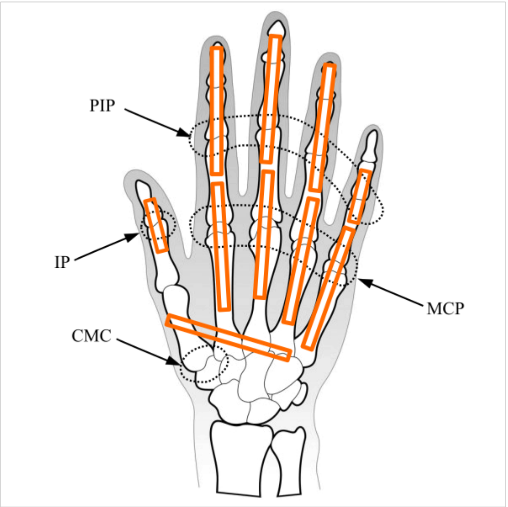


Fig. 3: Placement of the flex sensors to measure human hand articulations: Proximal interphalangeal (PIP), metacarpophalangeal (MCP), interphalangeal (IP) and carpometacarpal (CMC) joints.

As for the controller of our sensor glove, we chose ESP32-S2 [28] microcontroller which provides a built-in WiFi module for connectivity, a 10-channel 13-bit analog to digital converter (ADC) for the connection of the 10 bend sensors and an I2C-bus for the IMU. This controller holds just the right amount of analog input channels for our sensor layout while providing a higher resolution than most competitive microcontrollers. It is also programmable

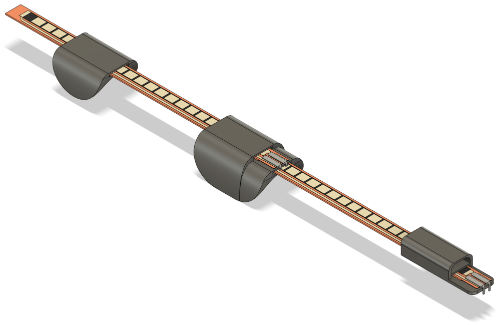


Fig. 4: Two flex sensors mounted onto the 3D printed guiding shapes. This represents the arrangement on one finger.

through the widely-used Arduino IDE [29] that allows the use of many pre-build libraries.

For tracking rotation and hand movement, we attached an Adafruit BNO055 [30] Inertial Measurement Unit (IMU) to the center of the back of the glove, which allows for reasonably good orientation tracking of the hand posture. This breakout board provides a 9-degree-of-freedom IMU coupled with a controller to calculate the absolute orientation, then making it accessible through the I2C-bus that is hooked up to our microcontroller.

Furthermore, a WiFi connection offers more flexibility by being able to broadcast information to every other device in one network, while a Bluetooth connection is restricted to peer-to-peer communication with one other device. The resulting availability of the sensor data is furthermore supported by our usage of a ROS interface. In summary, we benefit from the intuitive, inexpensive and accurate way of realizing a sensor glove by using flex sensors. Additionaly, we are elevating our system by incorporating a WiFi connection, offering a ROS interface and providing a real-time visualization.

### B. Software implementation

As mentioned before, the microcontroller used in our setup is programmable through the widely-used Arduino IDE [29]. We therefore can make use of many prebuild libraries that allow the implementation of a ROS interface on the glove, publish the sampled sensor data in ROS messages and make it available to any other machine running ROS in the WiFi network. We furthermore implemented a complex simulation environment in Unity, shown in Figure 1, that can display the published data in real time, allowing the user a good understanding of the raw sensor data generated by the glove. Since this is only available on Windows platform, we also implemented a simpler simulation in Gazebo. For better visualization, we also simulated the flexion of the interphalangeal joints of each finger, that has no flex sensor attached to it, by presuming it to be half as flexed as the proximal interphalangeal joints of that finger.

Moreover, as proposed in [31], we also used hyper sampling for collecting smooth sensor data, which means for each flex sensor 50 measurements are taken, 10% of the highest and lowest values were discarded. Afterwards, the average of the remaining 80% of measurements are taken. This minimizes jittering in the sensor data and drastically smoothes the output.

### C. Calibration

The calibration of the sensors before data acquisition is essential for precise and accurate output. This process should be executed always before usage of the sensor glove. The IMU already includes internal algorithms to calibrate the gyroscope, accelerometer and magnetometer inside the device. Therefore, the calibration method is started and carried through until the calibration status reaches its maximum.

The flex sensors are calibrated using two different approaches, a fast, but less accurate and a complex but accurate one. For the first calibration approach, different hand poses are performed. The sensor values are measured in straight position (0°) by laying the stretched hand on a flat surface. Afterwards, the calibration pose of the hand is a fist with the thumb laying outside. In this position, the sensor values are measured for a bending of approximately 90° except the two sensors for the thumb. Lastly, the thumb is bent inwards for its calibration. It is important to point out, that this calibration procedure is the simplest and quickest, but has some major drawbacks in the resulting sensor accuracy. Since one can not be sure of the exact bend of the joints when performing a fist, the resulting accuracy of the sensors may suffer from an offset, which can not be ignored.

Therefore, the second approach is attempted to reach a more accurate calibration. Since the joint positions for these different hand poses may not exactly match the assumed positions of 0° and 90°, we will use 3D-printed guiding shapes to ensure a precise flexion. These guiding shapes match the previously printed pieces for the flex sensors and allow a seamless placement of the fingers within the guiding shapes. Subsequently, every sensor has to be calibrated individually with placing the guiding shapes, which leads to a longer calibration phase of the whole hand.

### D. Trajectory Learning with Movement Primitives

We assume that measurements are given as trajectory, i.e., a sequence of multi-dimensional sensor values denoted by the matrix $\boldsymbol{\tau} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T]$. Here, $\boldsymbol{\tau}$ is modeled through a basis function approximation using the parameter vector $\boldsymbol{w}$,

$$p(\boldsymbol{\tau}|\boldsymbol{w}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{\Phi}_t\boldsymbol{w}, \boldsymbol{\Sigma}_y) = \mathcal{N}(\boldsymbol{y}_{1:T}|\boldsymbol{\Phi}_{1:T}\boldsymbol{w}, \boldsymbol{\Sigma}_y),$$

where $\boldsymbol{\Phi}_t$ denote the basis function matrix defined in [19], [20]. $\boldsymbol{\Sigma}_y$ denotes the measurement noise which is often specified manually, i.e., we used $\boldsymbol{\Sigma}_y = 10^{-12}\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix with convenient dimension.

TABLE I: Comparison of different sensor gloves

| Reference | Sensing method | Raw data | Wireless | Measurement accuracy | Sampling rate |
|---|---|---|---|---|---|
| Nassour et al. [6] | Electrolyte solution KI-Gly | 14 (Bend, Abd.) | Yes (Bluetooth) | 0.885° | 100 Hz |
| W. Park et al. [7] | Conductive liquid metal | 15 (Bend, Abd.) | Not mentioned | 1.39° | Not provided |
| Chossat et al. [8] | Ionic liquid and liquid metal | 11 (Bend) | No | Not provided | 200 Hz |
| Glauser et al [5] | Capacitive sendors | 44 (Bend, Abd.) | No | 7.6° | 60 Hz |
| Lin et al. [4] | IMU arrays | 17 (9-axes IMU's) | Yes (Bluetooth) | 3° | 50 Hz |
| Li et al. [11] | Optical linear encoders | 10 (Bend) | Yes (Bluetooth) | 1° | 150 Hz |
| Y. Park et al. [10] | Linear potentiometers | 10 (Bend) | Not mentioned | 0.65° | Not provided |
| Cyberglove [12] | HyperSensor™ | 18 (Bend, Abd.) | Yes(WiFi) | 1° | 120 Hz |
| Gentner et al. [14] | Flex sensors | 14 (Bend, Abd.) | No | 0.1° | 50 Hz |
| Weber et al. [13] | Flex sensors | 10 (Bend) | Yes (Bluetooth) | Not Provided | 25 Hz |
| **Our proposed glove** | **Flex sensors** | **10 (Bend)** | **Yes (WiFi)** | **$\leq 2°$** | **20 Hz** |

To compute the movement model $p(\boldsymbol{\tau})$, the parameter vector $\boldsymbol{w}$ is integrated out, i.e.,

$$\begin{aligned} p(\boldsymbol{\tau}) &= \int p(\boldsymbol{\tau}|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w} \\ &= \int \mathcal{N}(\boldsymbol{y}_{1:T}|\boldsymbol{\Phi}_{1:T}\boldsymbol{w}, \boldsymbol{\Sigma}_y)\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)d\boldsymbol{w} \\ &= \mathcal{N}(\boldsymbol{y}_{1:T}|\boldsymbol{\Phi}_{1:T}\boldsymbol{w}, \boldsymbol{\Phi}_{1:T}\boldsymbol{\Sigma}_w\boldsymbol{\Phi}_{1:T}^T + \boldsymbol{\Sigma}_y). \end{aligned}$$

This marginalization process can be computed in closed form and results in a single Gaussian distribution. The core of the movement model is the Gaussian prior $\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$, which can be computed from the sensor measurements (i.e., the training data) through maximum likelihood [32] or in the simplest case through ridge regression,

$$\boldsymbol{w}^{[i]} = (\boldsymbol{\Phi}_{1:T}^{\intercal}\boldsymbol{\Phi}_{1:T} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}_{1:T}^{\intercal}\boldsymbol{\tau}^{[i]}.$$

The regularization term $\lambda$ is set to $\lambda = 10^{-6}$ in our experiments.

Note that the correlation of the multi-dimensional data is captured through the covariance matrix $\boldsymbol{\Sigma}_w$. This covariance can be used to predict individual dimensions (exemplary shown in Figure 2(d)). For example in [23], the right wrist motion is predicted from a few measurements of optical markers placed on the left wrist.

In the following, we summarize how to compute model similarities. For further probabilistic operations like the *computation of predictions* or *likelihoods*, we refer to [19], [23]. In a training phase, measurements of elementary movement primitives like the playing of a guitar were collected. This training data is used, as discussed above, to compute model priors, like $\mathcal{N}(\boldsymbol{w}_{\text{guitar}}|\boldsymbol{\mu}_{\text{guitar}}, \boldsymbol{\Sigma}_{\text{guitar}})$, $\mathcal{N}(\boldsymbol{w}_{\text{sax}}|\boldsymbol{\mu}_{\text{sax}}, \boldsymbol{\Sigma}_{\text{sax}})$, etc. To evaluate the similarities of different movement primitives, we computed the Kullback-Leibler divergence of the two Gaussian prior distributions,

$$\begin{aligned} \text{KL}(\mathcal{N}_1||\mathcal{N}_2) = \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} - n + \text{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + \\ (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^{\intercal}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1), \end{aligned}$$

where $\mathcal{N}_1$ denotes a Gaussian distribution with the n-dimensional mean $\boldsymbol{\mu}_1$ and the covariance $\boldsymbol{\Sigma}_1$. The symbol tr denotes the matrix trace. This probabilistic measure in the parameter space is also compared to a deterministic measure computing Euclidean distances in the trajectory space (exemplary shown in Figure 2(d)). The evaluation results are shown in Subsection III-B in the experiments.

## III. RESULTS

In this section, we start by evaluating the performance of our sensor glove and give insights into our experiments to record the movement primitives used for trajectory learning. In the later part, we demonstrate motion model learning based on recorded movement data.

### A. Performance

The performance of our sensor glove is evaluated by three criteria, namely, the *accuracy* of the flex sensors in our design, the *sampling rate*, in which the sensor data is collected and also the *latency* of the whole system connected to the simulation.

The measurement of the accuracy is, in our terms, described with the repeatability of the sensors when performing the same finger motion. By performing the same bending positions for different finger joints with the assistance of 3D-printed guiding shapes to secure the finger, we evaluated the accuracy of our flex sensors and therefore of our sensor glove to be below 2°. In comparison to the different approaches presented in Subsection I-A, that are also displayed in Table I, we can rank our sensor glove in the midfield of the comparable approaches. However, it must be considered that a direct comparison regarding results of related work is not always possible, since every approach differs in the method used for measuring the accuracy.

The second important aspect is the sampling rate of the whole sensor glove. This is highly dependent on the used microcontroller, or precisely the analog-to-digital converter to read the sensor values. While the overall sampling rate of the integrated ADC is sufficient to reach high amounts of measurements, the switching between the several channels is the main cause for longer sampling duration. Therefore, the amount of samples per sensor used for the hyper sampling and smoothing of the data is not very influential on the whole sampling rate. Due to these properties of the ADC, we reached a sampling rate of 20 Hz which ranks our glove at the bottom of the other designs shown in Table I. To achieve a higher sampling rate, one should include external ADC chips with a high sample rate for each flex sensor and then hook them up to a serial peripheral interface (SPI) bus. We did not take this approach, as it would raise the costs of our design significantly while also complicating the building and replicating process. Furthermore, our achieved sampling rate of 20 Hz is sufficient for our purpose of transfer learning and teleoperation.

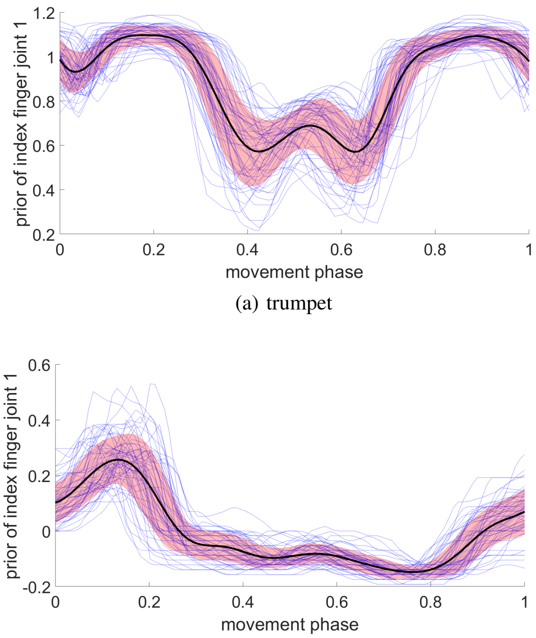
Especially for the teleoperation of robotic arms, it is important to have a real-time system with very low latency. In our setup with a normal WiFi router, we observed none to little latency of 1-10 milliseconds. This is not a noticeable delay for the human eye, so online teleoperation can be done with fast reactions to external influences. However, it should be considered, that this can vary and is highly dependent on the available network, but since we are working with very small data packages, the latency should always stay in an acceptable range for real-time applications.

In summary, the performance of our sensor glove has some minor drawbacks in the sampling rate when comparing it to the other approaches, while keeping up in measurement accuracy and also providing a real-time latency. But it fully satisfies the constraints of being low-budget (below 200€), highly accurate and very versatile, which makes it an appropriate fit for transfer learning of movement trajectories and also for teleoperating robot arms.

### B. Movement Evaluation

To analyze the performance of our sensor glove with respect to motion model learning scenarios, we recorded the hand motion while playing music instruments: a trumpet, a flute, a saxophone, a guitar and a keyboard. For every instrument, a short melody is repeatedly performed. This results in a data-set of 60 trajectories of each joint for each instrument respectively. In our experiment, we only captured the right hand finger motion. This is sufficient for our proof-of-concept. However, for future work, the additional data from the left hand may increase the detection of differences between the hand motions while playing these instruments.

We were able to generate movement primitives which allow a good representation of the given motion of each joint. Two exemplary generated trajectories are shown in Figure 5. Furthermore, we calculated the learned model similarities between each instrument (see Figure 6). It shows that our developed sensor glove is able to capture accurate hand motion to learn a motion model and also to show differences between them.
Since our main focus here were on the design and development of the sensor glove, this experiment only gives an insight into future work and is a proof-of-concept.



(a) trumpet

(b) keyboard

Fig. 5: Exemplary movement primitives derived from trajectories of the index finger MCP joint for different music instruments: (a) trumpet and (b) keyboard. The data-set of trajectories is illustrated as the thin blue lines, while the mean of the given trajectories is shown as a solid black line. The shaded area illustrates the 1-sigma confidence level.

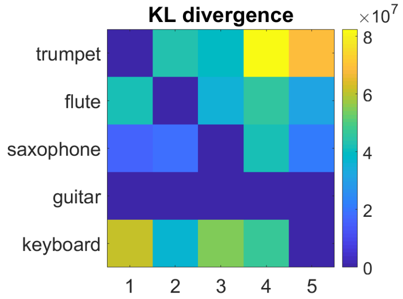


Fig. 6: Kullback-Leibler divergence of model similarities between each instrument. A low distance value denotes a high similarity and vice versa.

## IV. CONCLUSION

In in this paper, we proposed a low-cost sensor glove design that satisfies the prerequisites for trajectory learning with dynamic movement primitives. Our sensor glove reaches a high accuracy and repeatability as well as very low latency, which renders it suitable for other real-time application, for example teleoperation of robotic manipulators. Furthermore, our approach has a high flexibility due to the integrated WiFi connection and the implemented integration to ROS in combination with a real time simulation in Unity. It is also easily replicable, due to its low-cost design, i.e., only requires some 3D-printed parts. Our whole implementation is open source and available on

GitHub (`https://github.com/ai-lab-science/SensorGloves`). The main drawback of this simple approach is in the low sampling rate of the whole setup, which might be counteracted by extending the design with external analog-to-digital converters to get faster measurements of the multiple flex sensors. Nevertheless, we were still able to use our sensor glove to record different movement trajectories and successfully applied machine learning algorithms to analyze them. We also illustrated an initial result by fitting probabilistic movement primitives on our sensor recordings for playing different instrument, verifying our design for future experiments.

To further extend our sensor glove, we will enhance our design by the addition of coin vibration motors on the fingertips to be able to give force feedback to the user during teleoperation tasks. Furthermore, we will use the generated probabilistic model to generate new data for robot hand control. Additionally, we will enhance our experimental setup with a real robot arm to test the model and also the real time teleoperation of it using the sensor glove.

## V. ACKNOWLEDGEMENT

This project has received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) No #430054590 (TRAIN).

## REFERENCES

[1] V. Corporation. Valve index vr system. [Online]. Available: https://www.valvesoftware.com/de/index

[2] Optitrack. Optitrack - motion capture systems. [Online]. Available: https://optitrack.com/

[3] Ultraleap. Leap motion controller. [Online]. Available: https://www.ultraleap.com/product/leap-motion-controller/

[4] B.-S. Lin, I. Lee, S.-Y. Yang, Y.-C. Lo, J. Lee, J.-L. Chen *et al.*, "Design of an inertial-sensor-based data glove for hand function evaluation," *Sensors*, vol. 18, no. 5, p. 1545, 2018.

[5] O. Glauser, S. Wu, D. Panozzo, O. Hilliges, and O. Sorkine-Hornung, "Interactive hand pose estimation using a stretch-sensing soft glove," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–15, 2019.

[6] J. Nassour, H. G. Amirabadi, S. Weheabby, A. Al Ali, H. Lang, and F. Hamker, "A robust data-driven soft sensory glove for human hand motions identification and replication," *IEEE Sensors Journal*, vol. 20, no. 21, pp. 12 972–12 979, 2020.

[7] W. Park, K. Ro, S. Kim, and J. Bae, "A soft sensor-based three-dimensional (3-d) finger motion measurement system," *Sensors*, vol. 17, no. 2, p. 420, 2017.

[8] J.-B. Chossat, Y. Tao, V. Duchaine, and Y.-L. Park, "Wearable soft artificial skin for hand motion detection with embedded microfluidic strain sensing," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2568–2573.

[9] J. T. Muth, D. M. Vogt, R. L. Truby, Y. Mengüç, D. B. Kolesky, R. J. Wood, and J. A. Lewis, "Embedded 3d printing of strain sensors within highly stretchable elastomers," *Advanced materials*, vol. 26, no. 36, pp. 6307–6312, 2014.

[10] Y. Park, J. Lee, and J. Bae, "Development of a wearable sensing glove for measuring the motion of fingers using linear potentiometers and flexible wires," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 198–206, 2014.

[11] K. Li, I.-M. Chen, and S. H. Yeo, "Design and validation of a multi-finger sensing device based on optical linear encoder," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3629–3634.

[12] G. D. Kessler, L. F. Hodges, and N. Walker, "Evaluation of the cyberglove as a whole-hand input device," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 2, no. 4, pp. 263–283, 1995.

[13] P. Weber, E. Rueckert, R. Calandra, J. Peters, and P. Beckerle, "A low-cost sensor glove with vibrotactile feedback and multiple finger joint and hand motion sensing for human-robot interaction," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2016, pp. 99–104.

[14] R. Gentner and J. Classen, "Development and evaluation of a low-cost sensor glove for assessment of human finger movements in neurophysiological settings," *Journal of neuroscience methods*, vol. 178, no. 1, pp. 138–147, 2009.

[15] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nature neuroscience*, vol. 3, no. 11, pp. 1212–1217, 2000.

[16] J. Diedrichsen, R. Shadmehr, and R. B. Ivry, "The coordination of movement: optimal feedback control and beyond," *Trends in cognitive sciences*, vol. 14, no. 1, pp. 31–39, 2010.

[17] N. Bernstein, *The Co-ordination and Regulation of Movements*. Pergamon Press, 1967. [Online]. Available: https://books.google.de/books?id=F9dqAAAAMAAJ

[18] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[19] A. Paraschos, C. Daniel, J. Peters, G. Neumann *et al.*, "Probabilistic movement primitives," *Advances in neural information processing systems*, 2013.

[20] A. Paraschos, E. Rueckert, J. Peters, and G. Neumann, "Probabilistic movement primitives under unknown system dynamics," *Advanced Robotics*, vol. 32, no. 6, pp. 297–310, 2018.

[21] E. Rueckert, "Probabilistic trajectory model toolbox." [Online]. Available: https://ai-lab.science/wp/resources/code/MATLAB_ProbabilisticTrajectoryModel_2016Rueckert.zip

[22] A. Paraschos, "Movement primitives toolbox." [Online]. Available: https://www.ias.informatik.tu-darmstadt.de/uploads/Alumni/AlexandrosParaschos/ProMP_toolbox.zip

[23] E. Rueckert, J. Čamernik, J. Peters, and J. Babič, "Probabilistic movement models show that postural control precedes and predicts volitional motor control," *Scientific reports*, vol. 6, no. 1, pp. 1–12, 2016.

[24] O. Robotics. Ros. [Online]. Available: https://www.ros.org/

[25] U. Technologies. Unity. [Online]. Available: https://unity.com/

[26] . F. S. Systems. Flexpoint bend sensor. [Online]. Available: https://www.flexpoint.com

[27] S. Electronics. Sparkfun bend sensor. [Online]. Available: https://www.sparkfun.com/

[28] E. Systems. Esp32s2. [Online]. Available: https://www.espressif.com/en/products/socs/esp32-s2

[29] Arduino. Arduino ide. [Online]. Available: https://www.arduino.cc/en/software

[30] Adafruit. Adafruit bno055. [Online]. Available: https://www.adafruit.com/product/2472

[31] O. Nisar, M. A. Imtiaz, S. Hussain, and O. Saleem, "Performance optimization of a flex sensor based glove for hand gestures recognition and translation," *International Journal of Engineering Research & Technology*, vol. 3, no. 5, pp. 1565–1570, 2014.

[32] E. Rueckert, J. Mundo, A. Paraschos, J. Peters, and G. Neumann, "Extracting low-dimensional control variables for movement primitives," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1511–1518.