Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 8351–8356

Robotics Benchmark on Transfer Learning: a Human-Robot Collaboration Use Case ⋆

Asad Ali Shahid * **Marco Forgione** * **Marco Gallieri** ** **Loris Roveda** * **Dario Piga** *

* *Dalle Molle Institute for Artificial Intelligence, IDSIA USI-SUPSI, Via la Santa 1, CH-6962 Lugano-Viganello, Switzerland (e-mail: {asadali.shahid}@idsia.ch)*

** *NNAISENSE SA, Piazza Molino Nuovo 17, CH-6900 Lugano, Switzerland*

Abstract: In the context of human-robot collaboration (HRC), the model of the robots needs to be adapted to describe new tasks in new environments and under new operating conditions. A long-standing challenge in HRC is to transfer the acquired robot's skills and adapt models from a limited amount of data and/or with limited computational resources. To facilitate the research addressing this issue, this paper proposes a transfer learning benchmark in a HRC setting using data acquired from a 7-DOF Franka Emika Panda robot. The goal is to estimate a dynamical model mapping set-point pose, measured pose, and velocity of the end-effector into the external interaction wrench. This type of problem may arise in real applications to design virtual sensors for forces/torques. In the proposed benchmark, the model can be estimated based on a long dataset acquired under a nominal operating condition of the robot, along with 5 shorter trajectories (to be used for model adaptation) gathered for 5 different values of translational stiffness. Performance is measured on 5 test trajectories where the same translational stiffness values of the transfer experiments are applied. Baseline results are presented using a transfer learning approach tailored to dynamical systems recently proposed in the literature.

Keywords: System identification; Benchmark; Transfer learning; Human-robot collaboration; Robotics.

1. INTRODUCTION

Transfer learning has received increasing attention from the machine learning community in recent years. The main idea of transfer learning is to exploit the knowledge obtained on one task to learn a different but related task with less data and/or with a lower computational effort. A prominent form of transfer learning involves pre-training a model using a source dataset (*e.g.*, ImageNet in the artificial vision domain) and then fine-tuning it on the target dataset related to the new task (Chen et al., 2019). Despite the recent advances in the field, acquiring new skills from a few examples remains a formidable challenge for current automatic learning algorithms.

In the context of robotics, a wide range of operating conditions can influence the system dynamics over time (Roveda et al., 2020). For example, aging, wear and tear, joint friction, and other external effects can cause the dynamics to change in real operating conditions (Roveda et al., 2022a). Moreover, the robots are expected to collaborate with numerous users with varying characteristics (Roveda et al., 2022b). Therefore, it is crucial for the robots to transfer the acquired skills and learn to operate in new conditions from limited data. Having robots that can adapt to a wide variety of environments and effectively interact with new users is a long-standing challenge in human-robot collaboration.

Model learning in robotics has mainly been explored from the perspective of *system identification* (Nelles, 2001; Zhu et al., 2017; Evans et al., 2022). The key idea is to identify the model parameters in order to match the experimental data from a robot (also called dynamic model fitting). The model can then be used to learn a control policy. Yu et al. (2017) first trains an online model through supervised learning able to predict the environment parameters, and then uses reinforcement learning (RL) to train a control policy conditioned on these dynamic parameters. Other approaches of transferring the learned model involve *domain adaptation* (Chebotar et al., 2019), where the policies trained in the simulation are adapted to the real world or *meta-learning* that explicitly trains the model for its adaptation capability (Yu et al., 2020). A meta model is optimized across a variety of tasks in order to quickly adapt to new tasks given limited test data.

To the best of our knowledge, no public datasets are available to date to evaluate the transfer learning approaches in the context of HRC. To encourage research in this domain, a new benchmark for human-robot collaboration applications is introduced in this paper, with a main focus on transfer learning.

As a baseline, we provide the performance obtained on this benchmark by the transfer-learning methodology in-

⋆ This work was partially supported by the ROBOLUTIONARY project funded by the Hasler Foundation.

10.1016/j.ifacol.2023.10.1026

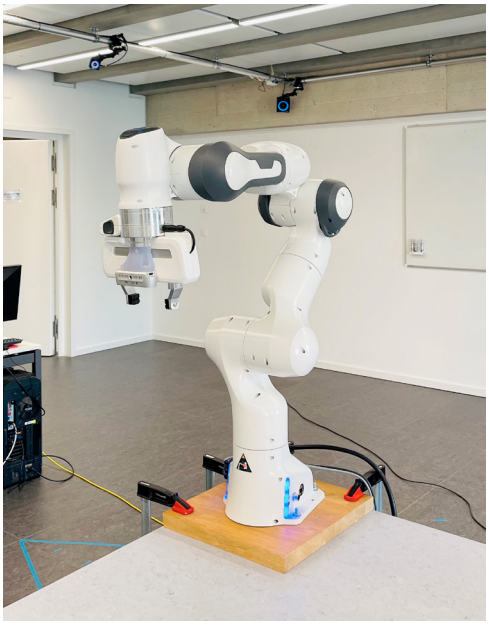

Fig. 1. The FRANKA Emika Panda manipulator used to generate data.

troduced in Forgione et al. (2022), where the nominal model (estimated on the training dataset) is augmented with a linear-in-the-parameter adaptation term learned on the transfer dataset. While the methodology is shown to be effective, we believe that there is room for further improvement; users are thus encouraged to implement their own approaches and report the achieved results.

2. ROBOT DESCRIPTION

The Franka Emika Panda robot shown in Figure 1 is used in this work. It is a 7-DOF collaborative arm. The weight of the arm is approximately 18 kg with a payload capacity of 3 kg. The torque sensors in each joint along with adjustable stiffness and compliance allow the Panda to be a versatile and powerful robot for the research community, especially considering its interaction with the surrounding environment.

Direct low-level control of the robot can be performed through the Franka Control Interface (FCI) implemented in the C++ library `libfranka`. The `franka_ros` package exposes `libfranka` to the ROS ecosystem for controlling the robot via ROS. The `franka_ros` provides many useful packages including `franka_example_controllers` to send real-time control values to the robot using different interfaces. Furthermore, FCI provides access to raw data on joints or links position, velocity, torque and estimation of externally applied wrenches.

3. DATA COLLECTION

3.1 Design of Experiments

The experiments are designed as multiple trajectories of different time duration, each collected using a specific translational stiffness for the `cartesian impedance example controller` available in ROS. During the data acquisition, the robot is moved in a collaborative hand-guiding mode for a certain duration. This kind of data collection is common in the usual application settings for human-robot collaboration, where the human might teach a new task to the robot by guiding it along a certain trajectory. The robot starts at the home position as shown in the Fig. 1 at the beginning of each trajectory. In total, 11 trajectories are collected: 1 of duration 100 s, 5 of duration 60 s, and 5 of duration 20 s. The controller and the data acquisition system run at a frequency of 1 kHz.

The recorded quantities at each time step are:

- Set-point position of the end effector expressed in base frame of the robot, $\mathbf{x_{ref}} \in \mathbb{R}^3$ (m);
- Set-point orientation of the end effector expressed in base frame of the robot, $\mathbf{q_{ref}} \in \mathbb{R}^4$ (quaternion)
- Measured position of the end-effector expressed in base frame of the robot, $\mathbf{x_{meas}} \in \mathbb{R}^3$ (m),
- Measured orientation of the end-effector expressed in base frame of the robot, $\mathbf{q_{meas}} \in \mathbb{R}^4$ (quaternion);
- Measured velocity vector of the end effector, $\mathbf{v_{meas}} \in \mathbb{R}^6$ defined as:
$$\mathbf{v_{meas}} = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^\top,$$
with v_x, v_y, v_z representing linear velocities (m/s) and ω_x, ω_y, ω_z angular velocities (rad/s).
- External wrench (force, torque) acting at the end effector expressed relative to the base frame, $\mathbf{F_{ext}} \in \mathbb{R}^6$, defined as:
$$\mathbf{F_{ext}} = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^\top,$$
with F_x, F_y, F_z representing forces (N) and τ_x, τ_y, τ_z torques (N $\cdot$ m).
- Coriolis force vector, $\mathbf{C} \in \mathbb{R}^7$ (N);
- Body Jacobian matrix expressed in the base frame, $\mathbf{J} \in \mathbb{R}^{6\times7}$;
- Control torque, $\tau_{\mathbf{ctrl}} \in \mathbb{R}^7$ (N $\cdot$ m);
- Null-space torque, $\tau_{\mathbf{null}} \in \mathbb{R}^7$ (N $\cdot$ m);
- Impedance torque, $\tau_{\mathbf{imp}} \in \mathbb{R}^7$(N $\cdot$ m);

The orientation quaternions are represented with order x, y, z, w.

Name	Duration (s)	Number of trajectories	Transnational Stiffness (N/m)
Train	100	1	2
Transfer	20	5	15, 50, 80, 130, 175
Test	60	5	15, 50, 80, 130, 175

Table 1. Trajectories in the dataset.

3.2 Dataset description

The original raw dataset contains training, transfer and test trajectories. The training trajectory is collected for a duration of 100 s at a translational stiffness value of 2 N/m, while the transfer and test trajectories are collected at 5 different values of translational stiffness, as specified in Table 1. The duration of the transfer and test trajectories are 20 s and 60 s, respectively.

The raw data recorded from the robot contains high-frequency signals at 1 kHz. Input and output trajectories are post-processed by resampling at 100 Hz.

3.3 Folder structure

The folder structure and file format of the data, both raw and post-processed, are detailed in Table 2. The `data_raw`

Folder name	Sub-folder	Content
`data_raw`	`train`	9 `.npy` files representing the recorded quantities for the training trajectory
	`transfer`	45 `.npy` files representing the recorded quantities for 5 transfer trajectories
	`test`	45 `.npy` files representing the recorded quantities for 5 test trajectories
`data_resampled`	`train`	2 `.npy` files representing the input/output of a training trajectory
	`transfer`	10 `.npy` files representing the input/output of 5 transfer trajectories
	`test`	10 `.npy` files representing the input/output of 5 test trajectories

Table 2. Structure and format of the dataset.

folder contains all the recorded quantities at the original sampling frequency of 1 kHz. The `data_resampled` folder contains the quantities $\mathbf{x_{ref}}$, $\mathbf{q_{ref}}$, $\mathbf{x_{meas}}$, $\mathbf{q_{meas}}$ $\mathbf{v_{meas}}$, $\mathbf{F_{ext}}$ resampled at 100 Hz.

The user of the benchmark is normally expected to work with these resampled quantities only, and otherwise should state explicitly which of the raw measurements are used to allow a fair comparison. The complete datasets are available in the GitHub repository Shahid (2020).

4. BENCHMARK DESCRIPTION

The main challenge proposed in this benchmark is to perform transfer learning from a trained nominal model to a different trajectory using the least possible transfer data. The proposed problem is motivated by the fact that it is not possible to model all of the environment variations for HRC applications.

4.1 Identification problem

The identification problem considered in this benchmark consists in constructing a dynamical model from an input $\mathbf{u}$ of dimension $n_u = 20$ to an output $\mathbf{y}$ of dimension $n_y = 6$. The input $\mathbf{u}$ consists of the set-point $\mathbf{x_{ref}}$, $\mathbf{q_{ref}}$, the measured pose of the end-effector $\mathbf{x_{meas}}$, $\mathbf{q_{meas}}$ and the velocity vector of the end-effector $\mathbf{v_{meas}}$, while the output $\mathbf{y}$ is the external wrench (force, torque) acting at the end-effector $\mathbf{F_{ext}}$, i.e., $\mathbf{y} = \mathbf{F_{ext}}$. Table 3 provides a summary of the considered input-output signals.

This mapping from a set-point pose, measured pose and velocity of end-effector to the external interaction wrench would allow the modelling of the external interaction wrench based on the robot state to be used without force/torque sensors in a real application (such as an interaction task in the industrial workplace).

The main aim is to reconstruct the complete output trajectory $\mathbf{y}$, given the first output sample $\mathbf{y}(0)$ and the input sequence $\mathbf{u}$, *i.e.*, the model should simulate the complete trajectory instead of doing one-step ahead prediction. Only the train and the transfer trajectories can be used for model learning.

The obtained performance has to be reported on the test trajectories, which shall not be accessed for any other purpose.

4.2 Performance Metrics

The performance of the adapted model is assessed on the test trajectories through the $\mathbf{R^2}$ coefficient, which is defined for i-th output channel as:

Input $\mathbf{u}$	Set-point pose $\mathbf{x_{ref}}$, $\mathbf{q_{ref}}$
	End-effector pose $\mathbf{x_{meas}}$, $\mathbf{q_{meas}}$
	End-effector velocity $\mathbf{v_{meas}}$
Output $\mathbf{y}$	External wrench $\mathbf{F_{ext}}$

Table 3. Identification problem: considered input and output variables.

$$\mathbf{R_i^2}(\mathbf{y_i}, \hat{\mathbf{y}}_\mathbf{i}) = 1 - \frac{||\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}||^\mathbf{2}}{||\mathbf{y_i} - \bar{\mathbf{y}}_\mathbf{i}||^\mathbf{2}}, \tag{1}$$

where $\hat{\mathbf{y}}_\mathbf{i}$ denotes the i-th component of the predicted external wrench, and $\bar{\mathbf{y}}_\mathbf{i}$ is the time average of the measured external wrench $\mathbf{y_i}$. As an aggregate metric, the mean $\bar{\mathbf{R}}^\mathbf{2}$ metric over all the dimensions of external wrench $\mathbf{y}$ is used:

$$\bar{\mathbf{R}}^\mathbf{2} = \frac{1}{n_y} \sum_{i=1}^{n_y} \mathbf{R_i^2}. \tag{2}$$

5. BASELINE EXAMPLE

In this section, baseline results for the proposed benchmark are presented. For the model adaptation, the approach proposed by Forgione et al. (2022) is employed.

5.1 Nominal model training

The nominal model is a state-space neural network Forgione and Piga (2020) with recursive structure:

$$\mathbf{h}(k+1) = \mathcal{N}(\mathbf{h}(k), \mathbf{u}(k); \theta_{\text{nl}}) \tag{3a}$$

$$\hat{\mathbf{y}}(k) = \mathbf{h}(k), \tag{3b}$$

where $\mathbf{h} \in \mathbb{R}^6$ is the state of the network and equal to the (estimate of the) output of interest. In the state-space representation (3), $\mathcal{N}$ is a feed-forward neural network with unknown parameters θ_{nl}, two hidden layers; 25 units per layer; and *tanh* activation functions. The parameters θ_{nl} of the nominal model are obtained by minimizing the *mean square error* loss on the training dataset over sequences of length 500, where the initial state $\mathbf{h}(0)$ for each sequence is set equal to its corresponding measurements. For gradient-based optimization, the Adam algorithm is used with learning rate set to 10^{-3}.

The performance of the nominal model is then evaluated on a validation dataset (not used to learn the model parameters θ_{nl}) representing a 40-% portion of the training set. However, since this benchmark is intended for transfer learning, users are free to use the complete training dataset as they wish. For prediction, only the initial state and the input sequence is passed to the network in order to simulate the external wrench $\mathbf{F_{ext}}$ for a complete trajectory as mentioned in Section 4. Thus, at each step,

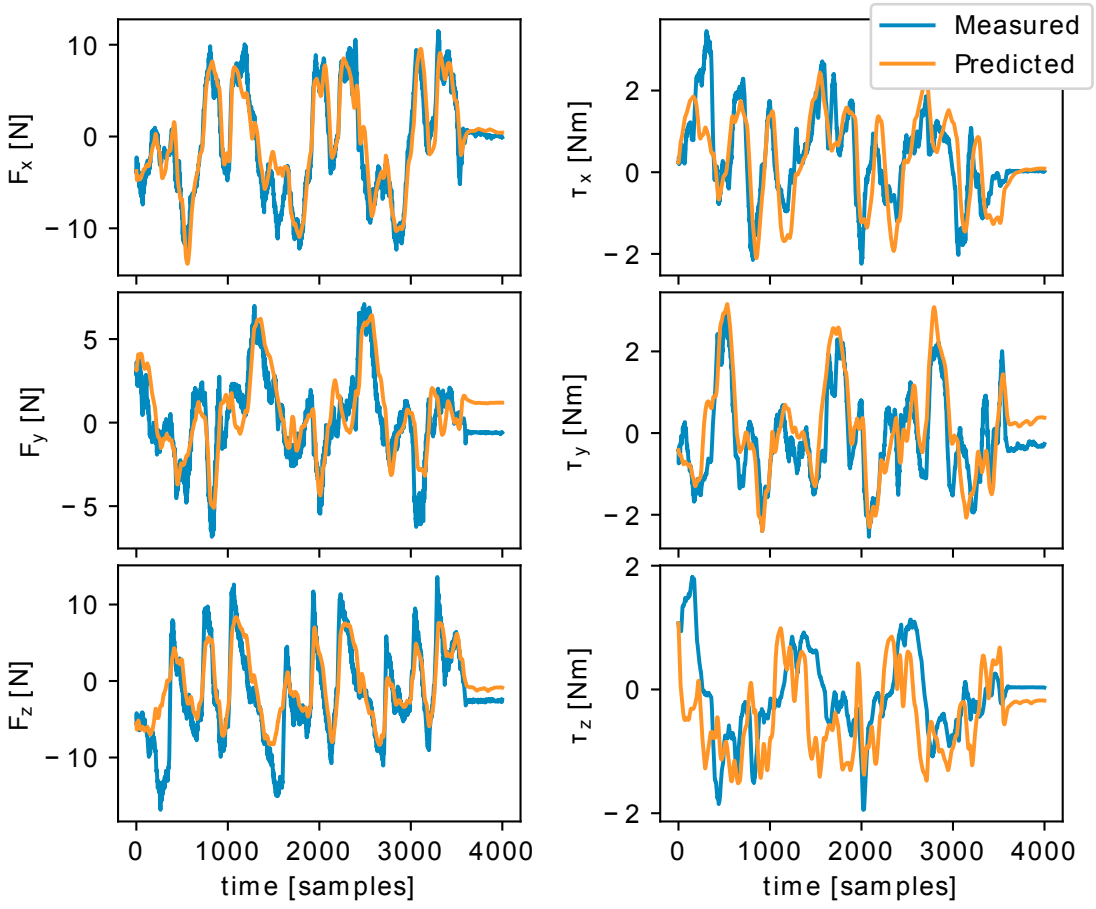

Fig. 2. Actual outputs *vs* outputs of the trained nominal model on a validation data extracted from the training dataset.

the model uses predictions (and not measurements) of a previous step. Results are shown in Figure 2.

5.2 Nominal model adaptation

The performance of the trained nominal model drops to a lower level on the test trajectories, which are collected using different values of translational stiffness. For instance, Figure 3 shows the nominal model's performance on a test trajectory collected at a translational stiffness value of 175 N/m. The corresponding $\mathbf{R^2}$ metrics are reported in Table 4 (fifth column).

Following the methodology in Forgione et al. (2022), we define the adapted model as:

$$\mathcal{M}_{\text{lin}}(\mathbf{u};\theta_{\text{nl}},\theta_{\text{lin}}) = \mathcal{M}(\mathbf{u};\theta_{\text{nl}}) + \mathbf{J}(\mathbf{u},\theta_{\text{nl}})\theta_{\text{lin}}, \quad (4)$$

where θ_{lin} is a new parameter vector to be determined and $\mathbf{J}(\mathbf{u},\theta_{\text{nl}})$ is the Jacobian of the nominal model's output with respect to the parameter vector θ:

$$\mathbf{J}(\mathbf{u},\theta_{\text{nl}}) = \left.\frac{\partial \mathcal{M}(\mathbf{u};\theta)}{\partial \theta}\right|_{\theta=\theta_{\text{nl}}}. \quad (5)$$

The adapted system dynamics (4) may be interpreted as a first-order Taylor expansion of the nominal model with respect to its nominal parameters. Assuming that the perturbed dynamics can be captured by the given model structure with a different parameter θ'_{nl}, we have:

$$\mathcal{M}(\mathbf{u};\theta'_{\text{nl}}) \approx \mathcal{M}(\mathbf{u};\theta_{\text{nl}}) + \mathbf{J}(\mathbf{u},\theta_{\text{nl}})(\theta'_{\text{nl}} - \theta_{\text{nl}}). \quad (6)$$

The right-hand-side of (6) corresponds indeed to (4) setting $\theta_{\text{lin}} = \theta'_{\text{nl}} - \theta_{\text{nl}}$.

The parameters θ_{lin} of the linear adaptation term in (4) are estimated on the transfer dataset through regularized linear least-squares (Ridge regression) as:

$$\theta_{\text{lin}} = (\mathbf{J}_{\mathbf{xf}}^{\top}\mathbf{J}_{\mathbf{xf}} + \sigma^{\mathbf{2}}\mathbf{I})^{-1}\mathbf{J}_{\mathbf{xf}}^{\top}\mathbf{\Delta y_{xf}}, \quad (7)$$

where $\mathbf{J_{xf}} = \mathbf{J}(\mathbf{u_{xf}},\theta_{\text{nl}})$, $\mathbf{\Delta y_{xf}} = \mathbf{y_{xf}} - \mathcal{M}(\mathbf{u_{xf}};\theta_{\text{nl}})$, and the subscript $\mathbf{xf}$ refers to the transfer dataset. More precisely, the initial 10-second segment of the transfer dataset is used to solve the least-squares problem (7), while the trailing 10-second segment is used as a validation set to tune the regularization hyper-parameter σ. Problem (7) is solved multiple times for different values of σ, and the one giving the best performance in the validation segment is eventually selected.

The predicted output in the test dataset is finally obtained as:

$$\hat{\mathbf{y}}_{\mathbf{tst}} = \mathcal{M}(\mathbf{u_{tst}};\theta_{\text{nl}}) + \mathbf{J}(\mathbf{u_{tst}},\theta_{\text{nl}})\theta_{\text{lin}}, \quad (8)$$

where the subscript $\mathbf{tst}$ refers to the test dataset.

The output trajectories simulated with the adapted model with translational stiffness of 175 N/m in the test dataset are shown in Figure 4 and the corresponding $\mathbf{R}^2$ metrics are reported in Table 4 (last column). The performance of adapted models on the other trajectories in the test dataset is also reported in Table 4. We observe that the

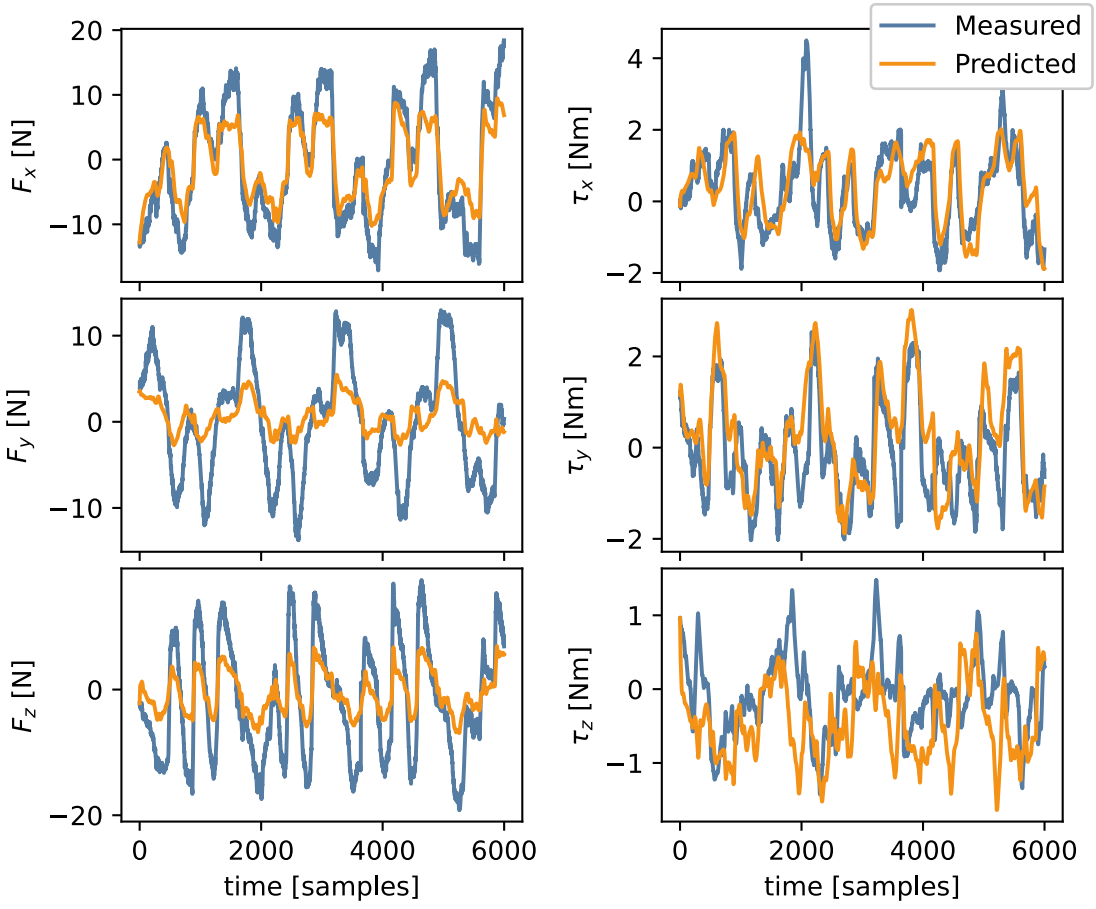

Fig. 3. Actual outputs *vs* outputs of the trained nominal model on a test trajectory with translational stiffness of 175 N/m.

	Nominal					Adapted				
k_{trans} (N/m)	15	50	80	130	175	15	50	80	130	175
F_x	0.63	0.51	0.79	0.76	0.79	0.85	0.63	0.83	0.72	0.84
F_y	0.50	0.47	0.53	0.43	0.41	0.52	0.75	0.70	0.62	0.73
F_z	0.44	0.38	0.61	0.51	0.55	0.64	0.41	0.75	0.57	0.78
τ_x	0.39	0.30	0.42	0.35	0.56	0.43	0.56	0.38	0.31	0.54
τ_y	0.08	0.15	0.22	0.43	0.32	0.18	0.41	0.26	0.32	0.56
τ_z	0.02	0.21	0.17	0.23	0.27	0.19	0.35	0.36	0.45	0.42
Overall	0.34	0.33	0.46	0.45	0.48	0.46	0.52	0.54	0.50	0.65

Table 4. R^2 performance metric of nominal and adapted model on test trajectories.

performance of the adapted model is consistently superior to the nominal one. The achieved accuracy level of wrench estimation would be sufficient for real applications such as force-based control or collision avoidance. Note that for the adapted models, the corresponding Jacobian $\mathbf{J_{xf}}$ is computed for each individual transfer trajectory as mentioned in Section 5.2, *i.e.* transfer and test is done independently for each value of translational stiffness.

6. CONCLUSIONS

In this paper, a transfer learning benchmark is presented in the HRC context with the main aim of adapting learned models to new operating conditions. The proposed challenge addresses the problem of estimating the external wrench based on the state of the system, thus eliminating the need for external force/torque sensors. Although some cobots allow the computing of external wrenches based on joint torque sensors, the cost of such sensors is usually very high. The model is first trained on a dataset acquired at a nominal value of translational stiffness and later adapted to work under new stiffness values using a baseline methodology proposed recently in the literature. We present the results obtained by applying the baseline methodology to the 5 pairs of transfer/test trajectories available in our benchmark. Even though the achieved results are promising, there is definitely room for improvement to enable the applications of transfer learning in real-

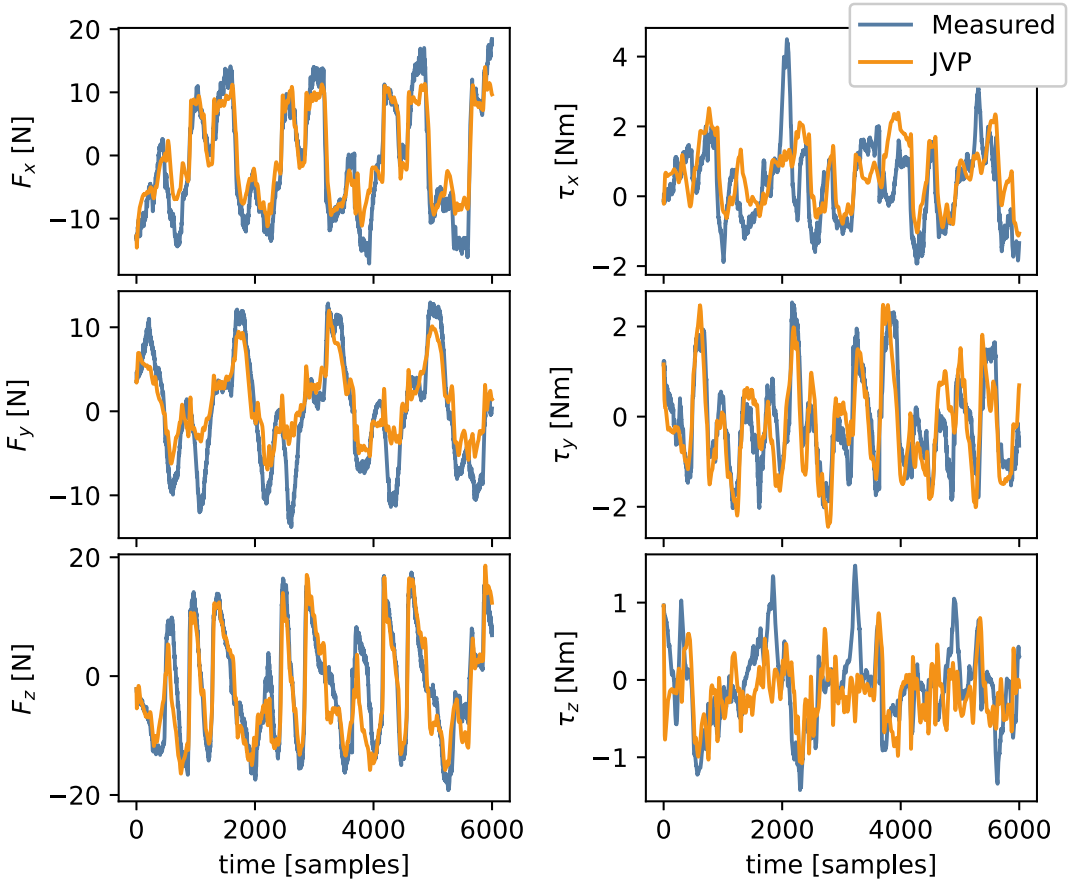

Fig. 4. Actual outputs *vs* outputs of the adapted model on a test trajectory with translational stiffness of 175 N/m.

life HRC tasks. This benchmark aims to encourage the research community to develop novel approaches in order to facilitate research in this domain.

REFERENCES

Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N., Fox, D., 2019. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, pp. 8973–8979.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., Huang, J.-B., 2019. A closer look at few-shot classification. In: International Conference on Learning Representations. URL `https://openreview.net/forum?id=HkxLXnAcFQ`

Evans, B., Thankaraj, A., Pinto, L., 2022. Context is everything: Implicit identification for dynamics adaptation. arXiv preprint arXiv:2203.05549.

Forgione, M., Muni, A., Piga, D., Gallieri, M., 2022. On the adaptation of recurrent neural networks for system identification. arXiv preprint arXiv:2201.08660.

Forgione, M., Piga, D., 2020. Model structures and fitting criteria for system identification with neural networks. In: 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT). pp. 1–6.

Nelles, O., 2001. Nonlinear dynamic system identification. In: Nonlinear System Identification. Springer, pp. 547–577.

Roveda, L., Bussolan, A., Braghin, F., Piga, D., 2022a. Robot joint friction compensation learning enhanced by 6d virtual sensor. International Journal of Robust and Nonlinear Control.

Roveda, L., Forgione, M., Piga, D., 2020. Robot control parameters auto-tuning in trajectory tracking applications. Control Engineering Practice 101, 104488.

Roveda, L., Testa, A., Shahid, A. A., Braghin, F., Piga, D., 2022b. Q-learning-based model predictive variable impedance control for physical human-robot collaboration. Artificial Intelligence 312, 103771.

Shahid, A., 2020. Transfer learning benchmark for human-robot collaboration. `https://github.com/Asad-Shahid/TransferLearning-Benchmark-HRC`.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S., 2020. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Conference on robot learning. PMLR, pp. 1094–1100.

Yu, W., Tan, J., Liu, C. K., Turk, G., 2017. Preparing for the unknown: Learning a universal policy with online system identification. arXiv preprint arXiv:1702.02453.

Zhu, S., Kimmel, A., Bekris, K. E., Boularias, A., 2017. Fast model identification via physics engines for data-efficient policy search. arXiv preprint arXiv:1710.08893.