2nd Workshop Toward Robot Avatars, IEEE International Conference on Robotics and Automation (ICRA), London, UK, 2023.

6D Dynamic Tool Compensation using Deep Neural Networks to improve Bilateral Telemanipulation

Stephan Andreas Schwarz[1], Carl Gaebert[1] and Ulrike Thomas[1]

***Abstract*— Force feedback is a crucial component to improve the accuracy and transparency in telemanipulation. Unfortunately, attached tools distort the measured forces of the force sensor. Thus, a compensation of the static and dynamic forces and torques is desired to estimate the robot's actual interactions with the environment. Due to the inaccuracy of model-based approaches, this paper presents a model-free approach to estimate the 6D forces and torques resulting from an attached tool to compensate the measurement of the force-torque sensor. We use a deep neural network to achieve this and compare multiple combinations of neuron numbers and inputs with an already existing approach. Experiments on a real telemanipulation setup show that the proposed algorithm has a higher accuracy with mean Euclidean errors of only $[0.7307 \pm 0.4974]\,\mathrm{N}$ in force and $[0.031 \pm 0.02]\,\mathrm{Nm}$ in torque. The low computation time of $0.12\,\mathrm{ms}$ makes it suitable for real-time applications such as telemanipulation.**

I. INTRODUCTION

Bilateral telemanipulation received a lot of attention within the last decades [1]. Usually, a force sensor is mounted between the end effector and the used tool to measure the interaction force [2]. This increase of transparency is especially used in telemedicine to improve the safety and performance of the operation [3], [4]. Unfortunately, the sensor does not directly measure the interaction force but the sum of the interaction forces and the force resulting from the attached tool. Thus, it is necessary to compensate the tool force to guarantee accurate force feedback.

In this paper, we present a deep neural network (DNN) to estimate forces and torques induced by the tool. This enables to extract the interaction force from the measurement of the sensor. Further, we tested this approach on a real telemanipulation setup and compared it to the approach presented in [5] that we extended so that it also contains the torques. We also compared it to a model-based approach. The experiments show that our approach results in more accurate estimations while having a lower computation time.

II. RELATED WORK

Conventionally, the exerted forces from the tool are compensated using dynamic tool identification approaches such as presented in [6] and [7]. Unfortunately, the accuracy of these widely used model-based techniques is limited due to the lack of specific dynamic models [2]. Thus, model-free approaches using machine learning techniques gained more attention during the last years. The authors of [2] and [8] present a multi-layer neural network to estimate the gravitational forces of the tool depending on the end effector orientation. They expand this approach in [5] by considering also the Cartesian velocities and accelerations and by using a deep convolutional neural network (DCNN). A dropout layer is used to increase noise robustness, reduce computation time and enhance stability.

Our approach differs from the present ones since we expand the algorithm from [5] to estimate not only the three-dimensional forces but also the torques generated by the tool. Further, we present a DNN based on real training data and compare the results of both architectures. We first use the the inputs presented in [5] but also reduce the input vector to the minimum needed values. Moreover, we define a suitable choice of the parameters to achieve a satisfying result and finally compare these results with a model-based approach.

III. METHODOLOGY

We shortly describe the applied deep convolutional neural network introduced in [5] for the sake of clarity and present our extensions on this approach. Furthermore, our DNN architecture is introduced to compensate the tool forces.

A. Deep Convolutional Neural Network

Su et al. present a deep neural network to compute the three-dimensional force resulting from the used tool that is measured by the mounted force sensor and distorts the measurement [5]. They use four convolutional modules, each with a 2D convolutional layer, a Rectified Linear Unit function and a batch normalization layer. A dropout layer is used to reduce overfitting as well as the computation time. Finally, a fully connection layer computes the three-dimensional force. The input $\boldsymbol{X} = [\boldsymbol{\theta}, \boldsymbol{V}_P, \boldsymbol{V}_\theta, \boldsymbol{A}_P, \boldsymbol{A}_\theta]$ consists of the euler angles $\boldsymbol{\theta}$ of the end effector, the six-dimensional velocity $\boldsymbol{V}_P$, $\boldsymbol{V}_\theta$ and acceleration $\boldsymbol{A}_P$, $\boldsymbol{A}_\theta$.

They use the Homogeneous matrix to further improve the results of the convolutional network. The final input matrix is given with

$$\boldsymbol{X}^* = \left[\boldsymbol{X}, \boldsymbol{X} - \bar{\boldsymbol{X}}, \tilde{\boldsymbol{X}}, |\boldsymbol{X}|, \boldsymbol{X}^2\right] \tag{1}$$

and $\tilde{\boldsymbol{X}} = \frac{\boldsymbol{X} - \bar{\boldsymbol{X}}}{\sigma(\boldsymbol{X})}$ with $\bar{\boldsymbol{X}}$ and $\sigma(\boldsymbol{X})$ being the average and variance of $\boldsymbol{X}$, respectively [5].

We extend the output of this approach to also contain the torques produced by the tool which enables a 6D dynamic tool compensation. The resulting output is $\boldsymbol{F} = [f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]$. We use a final linear layer with 32 neurons and removed the batch normalization and dropout layers since they did not perform well with our data.

[1]The authors are with the Robotics and Human-Machine-Interaction Lab at Chemnitz University of Technology, 09126 SN Chemnitz, Germany. Emails: {stephan-andreas.schwarz, carl.gaebert, ulrike.thomas}@etit.tu-chemnitz.de

TABLE I

EXPERIMENTAL RESULTS OF THE TEST MOTION AND THE COMPUTATION TIMES OVER 4000 RANDOM SAMPLES.

Network Architectur			Forces			Torques			Computation Time	
Name	Input	n	$\bar{\Delta F}$ [N]	$\sigma(\Delta F)$ [N]	$\Delta F_{\max}$ [N]	$\bar{\Delta\tau}$ [Nm]	$\sigma(\Delta\tau)$ [Nm]	$\Delta\tau_{\max}$ [Nm]	$\bar{t}_c$ [ms]	$\sigma(t_c)$ [ms]
DCNN	$\boldsymbol{X}^*$	-	1.2834	0.8978	6.0388	0.0383	0.0291	0.2510	0.4295	0.2179
DNN	$\tilde{\boldsymbol{X}}$	16	0.6964	0.5040	3.5715	0.0332	0.0215	0.1319	0.1241	0.0547
		32	0.8140	0.5484	4.1347	0.0295	0.0197	0.1486	0.1210	0.0543
	$\tilde{\boldsymbol{X}}'$	8	0.7307	0.4974	3.7051	0.0310	0.0200	0.1587	0.1200	0.0562
		16	0.6853	0.5026	3.7069	0.0343	0.0239	0.1640	0.1244	0.0658
		32	0.7825	0.5312	4.2771	0.0342	0.0240	0.2386	0.1250	0.0660

B. Deep Neural Network

Our DNN consists of three fully connected layers with the output $\boldsymbol{F}$ for the 6D tool compensation. As the input, we compare the full input $\boldsymbol{X}$ with the reduced input $\boldsymbol{X}' = [\boldsymbol{\theta}, \boldsymbol{V}_\theta, \boldsymbol{A}_P, \boldsymbol{A}_\theta]$.

We decided to ignore the translatoric velocities since they have no direct impact on the force of the tool. Note that this assumption is only valid while neglecting the air resistance. The final input is the mean adjusted representation $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{X}}'$. Further, we vary the amount of used neurons to find an ideal balance between accuracy and computation time.

IV. EXPERIMENTS

We use a real telemanipulation setup to generate our training data and to conduct our experiments. It consists of a Haption Virtuose 6D input device as the leader and a F6D100-50 force-torque sensor attached on a Franka Emika Panda robot as the follower. We implemented our algorithms in C++ using Torch for the neural network and ROS2 for the telemanipulation system. All processes run on a Linux real-time kernel using a Intel(R) Core(TM) i7-7700 with 3.60 GHz and 16 GB RAM.

We used 46417 samples for the training (10 % for the validation set) and a test motion that consists of 4670 samples. The sampling frequency was 10 ms and all data were unfiltered. Further, we trained our DNN with different amount of neurons. We used 8, 16 and 32 neurons for the reduced input $\tilde{\boldsymbol{X}}'$ and 16 and 32 with $\tilde{\boldsymbol{X}}$ for comparison. Finally, we evaluated the computation time of each network over 4000 random samples using the stated hardware configuration.

V. RESULTS

Fig. 1 shows the orientation of the tool and the resulting velocities during the test motion. This motion contains a variety of orientations with many different acceleration patterns in translation and rotation. Table I shows the results of each neural net with the errors during the test motion as well as the computation time of the 4000 random samples. The errors ΔF and $\Delta\tau$ describe the Euclidean error of the three-dimensional force and torque, respectively. n describes the amount of neurons for each layer within the net.

In general, accuracy and computation time of the DNN are better compared to the DCNN. The accuracy and computation time within the different versions of the DNN are quite similar. A proper choice of the network depends on the requirements of the system. A combintation of good force

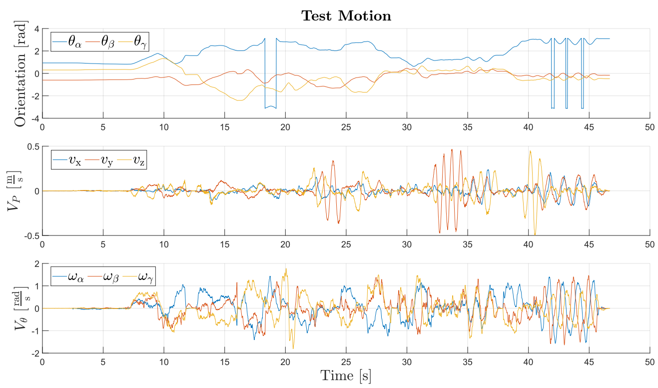

Fig. 1. Orientation and translatoric and angular velocity of the test motion to validate the trained networks.

and torque compensation together with very low computation times is achieved by the DNN without the translatoric velocity and 8 used neurons for each layer.

Finally, we compute the inertial parameters of the tool applying the approach presented in [9]. With these parameters, we receive mean Euclidean errors of $[1.4049 \pm 0.75]$ N and $[0.0362 \pm 0.255]$ Nm in force and torque for the test motion, which show a higher force error compared to our DNN.

VI. CONCLUSIONS

This work provides a model-free approach to compensate the force and torque distortion generated by a tool, mounted on a moving force-torque sensor. Thus, we expand an already proposed approach to include the measured torques. Further, we implement a simple deep neural network to achieve the same goal. We tested multiple input combinations and compared our results. The best results are received using a DNN with three layers, 8 neurons each and an input containing the tool orientation, the angular velocity and the 6D Cartesian acceleration. The norm of the errors are $[0.7307 \pm 0.4974]$ N and $[0.031 \pm 0.02]$ Nm and the maximum errors are 3.7051 N and 0.1587 Nm, respectively. The computation time of this approach was 0.12 ms on average which makes it very suitable for real-time applications such as telemanipulation. Further work aims at developing a similar approach for a flexible tool such as a robotic hand or a three-finger gripper.

ACKNOWLEDGMENT

This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German ResearchFoundation) – Project-ID 416228727 – SFB 1410.

REFERENCES

[1] S. N. F. Nahri, S. Du and B. J. Van Wyk, "A Review on Haptic Bilateral Teleoperation Systems," in Journal of Intelligent & Robotic Systems, vol. 104, no. 13, 2022.

[2] H. Su, W. Qi, Y. Hu, J. Sandoval, L. Zhang, Y. Schmirander, G. Chen, A. Aliverti, A. Knoll, G. Ferrigno, and E. De Momi, "Towards Model-Free Tool Dynamic Identification and Calibration Using Multi-Layer Neural Network," in Sensors, vol. 19, no. 17, 2019.

[3] G. Tholey, J. P. Desai and A. E. Castellanos, "Force feedback plays a significant role in minimally invasive surgery: results and analysis," in Annals of surgery, vol. 241, January 2005.

[4] A. Naceri, J. Elsner, M. Tröbinger, H. Sadeghian, L. Johannsmeier, F. Voigt, X. Chen, D. Macari, C. Jähne, M. Berlet, J. Fuchtmann, L. Figueredo, H. Feußner, D. Wilhelm and S. Haddadin, "Tactile Robotic Telemedicine for Safe Remote Diagnostics in Times of Corona: System Design, Feasibility and Usability Study," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 10296–10303, October 2022.

[5] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno and E. D. Momi, "Deep Neural Network Approach in Robot Tool Dynamics Identification for Bilateral Teleoperation," in IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2943–2949, April 2020.

[6] C. M. Oddo, P. Valdastri, L. Beccai, S. Roccella, M. C. Carrozza and P. Dario, "Investigation on calibration methods for multi-axis, linear and redundant force sensors," in Measurement Science and Technology, vol. 18, no. 3, January 2007.

[7] J. Wu, J. Wang and Z. You, "An overview of dynamic parameter identification of robots," in Robotics and Computer-Integrated Manufacturing, vol. 26, no. 5, pp. 414–419, 2010.

[8] H. Su, C. Yang, H. Mdeihly, A. Rizzo, G. Ferrigno and E. De Momi, "Neural Network Enhanced Robot Tool Identification and Calibration for Bilateral Teleoperation," in IEEE Access, vol. 7, pp. 122041–122051, August 2019.

[9] W. Khalil, M. Gautier and P. Lemoine, "Identification of the payload inertial parameters of industrial manipulators," in Proceedings 2007 IEEE International Conference on Robotics and Automation, pp. 4943–4948, April 2007.