Estimating Perturbations from Experience using Neural Networks and Information Transfer


Erik Berger
Technische Universität Bergakademie Freiberg


David Vogt
Technische Universität Bergakademie Freiberg


Heni Ben Amor
Arizona State University

Estimating Perturbations from Experience using Neural Networks and Information Transfer

Erik Berger[1], David Vogt[1], Steve Grehl[1], Bernhard Jung[1], Heni Ben Amor[2]

***Abstract*— In order to ensure safe operation, robots must be able to reliably detect behavior perturbations that result from unexpected physical interactions with their environment and human co-workers. While some robots provide firmware force sensors that generate rough force estimates, more accurate force measurements are usually achieved with dedicated force-torque sensors. However, such sensors are often heavy, expensive and require an additional power supply. In the case of lightweight manipulators, the already limited payload capabilities may be reduced in a significant way. This paper presents an experience-based approach for accurately estimating external forces being applied to a robot without the need for a force-torque sensor. Using Information Transfer, a subset of sensors relevant to the executed behavior are identified from a larger set of internal sensors. Models mapping robot sensor data to force-torque measurements are learned using a neural network. These models can be used to predict the magnitude and direction of perturbations from affordable, proprioceptive sensors only. Experiments with a UR5 robot show that our method yields force estimates with accuracy comparable to a dedicated force-torque sensor. Moreover, our method yields a substantial improvement in accuracy over force-torque values provided by the robot firmware.**

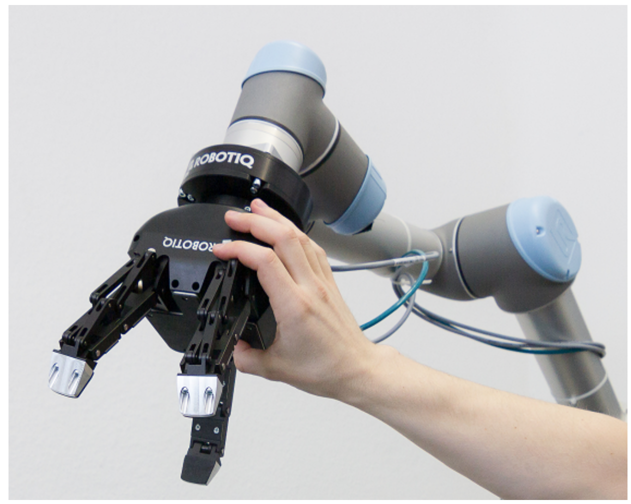

Fig. 1. For safe behavior execution, robots must be able to reliably detect perturbations resulting from unexpected physical interactions with their environment and human co-workers.

I. Introduction

The ability to sense the environment is a vital requirement for intelligent and safe robotics. Modern sensors, such as force-torque sensors (FT) can be used to measure external influences on a robot and, in turn, generate adequate responses. However, it is often difficult to distinguish between natural, behavior-related fluctuations in the sensor readings and external perturbations that are caused by forces or collisions applied by the outside world. Dynamic tasks in particular can cause significant variation in sensor readings that could potentially be mistaken for external influence.

In recent years, various methods have been proposed for behavior-specific estimation of external perturbations [1][2]. In prior work, we have introduced a new methodology for *estimating forces from experience* [3]. We have shown that nonlinear state prediction and machine learning can be used to generate accurate estimates of external perturbations, even in the absence of force measuring sensors. First, a model of the expected sensory feedback during a physical task is learned. During runtime, expected sensations are compared to measured sensor values and the difference is transformed into a perturbation estimate. Our results showed that feature extraction is a key component to the above methodology.

[1]Institute of Computer Science, Technical University Bergakademie Freiberg, Bernhard-von-Cotta-Str. 2, 09599 Freiberg, Germany

[2]School of Computing, Informatics and, Decision Systems Engineering, Arizona State University, 699 S Mill Ave, Tempe, AZ 85281, USA

Recent advances in deep learning research have produced powerful neural network models for feature extraction and nonlinear regression [4]. In this paper, we investigate such deep learning techniques as the core component of our methodology. To this end, we will contrast feature extraction using autoencoders to our previous approach using Transfer Entropy [5]. For modeling the evolution of sensor values in time, we will employ different neural network architectures, including feedforward and recurrent neural networks. One potential advantage of neural networks over previous approaches, is their scalability to large numbers of input dimensions and large numbers of data points, as well as their ability to represent *both* feature extraction and regression within the same framework.

In the remainder of this paper, we will introduce our methodology for perturbation estimation and show how neural networks can be used to model the evolution of sensor values in time. We will present a number of experiments to identify performance of different neural network architectures in this application domain and evaluate the accuracy of these results.

II. Related Work

Robots that engage in physical interactions with humans and objects need to regulate the forces exchanged with their environment. This requires methods for estimating the occurring intrinsic and external forces to allow for appropriate robot responses. Recent developments in compliant

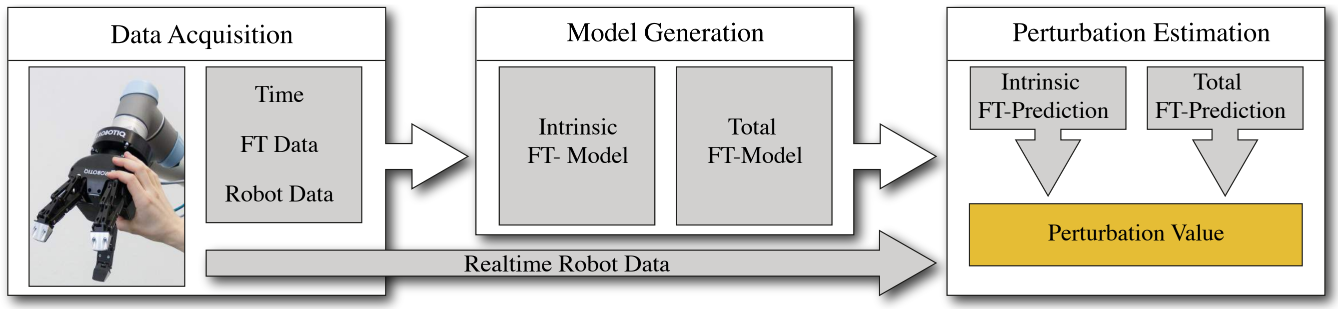

Fig. 2. An overview of the presented machine learning approach. During an offline training phase, the robot sensor data together with information about the time, force, and torque are recorded (Section III-A). The recorded data is analyzed for features and used to learn two models (Section III-B). In order to estimate external influences, the robot's realtime sensor data is used to predict intrinsic (Section III-C) and total force-torque measurements(Section III-D). The difference between both is used as estimation of the actual external perturbation.

control have lead to the emergence of robots with joint torque sensing and feedback control [6]. For measuring external forces and perturbations, however, typically additional force-torque (FT) sensors are used. Such sensors are often expensive, add weight to the robot, and require additional power supply. Hence, various authors have suggested using algorithmic approaches for inferring applied forces. In [7], a depth camera is used in order to estimate applied forces. Using a depth camera allows for generic contact locations on the robot. In [1] machine learning methods are used to extract an inverse dynamics model for a cable-driven robot manipulator. Measuring the difference between predicted controls from the inverse dynamics model and the executed controls provides an estimate for external forces applied on the robot. Using machine learning for force-based robot control has also been suggested a number of other publications. In [8] a robot learns to adapt its motion by anticipating human intentions from force measurements only. In [9], a method for learning force-based manipulation skills from demonstrations was presented. The approach generated variable-impedance control strategies thereby producing the necessary compliance for handling deformable objects.

The work presented in the remainder of our paper focuses on different aspect: how can a robot learn to estimate forces from experience? In contrast to earlier discussed papers, we learn behavior-specific models for intrinsic and total force estimation. To increase the accuracy, an additional FT-sensor is used during training these models. During runtime the FT-sensor is not available anymore and therefore is predicted by the learned models from the robot's sensor data only. Finally, the difference between intrinsic and total forces is used to predict the magnitude and direction of external perturbations.

III. Approach

The goal of the presented approach is to learn a behavior-specific perturbation model which is able to predict the readings of a FT-sensor from previous experience. Specifically, a FT-sensor with six degrees of freedom (Robotiq FT150) is mounted close to the *tool center point* (TCP) of a robotic arm (Universal Robots UR5). Figure 1 shows the principal setup including a gripper for pick-and-place operations.

The depicted robot provides 104 different measurements from a multitude of internal sensors, e.g., the applied current or the joint encoder state. In our approach, sensor readings generated from these sensors of the UR5 are used to learn a model that predicts FT-values measured by the FT150. The force-torque sensor is used in this context to provide ground truth data about all measured forces acting on the robot.

The basic rationale underlying our approach is that accurate estimates of forces applied on the robot can be generated by fusing information and evidence from a large number of low-cost sensors. Note that these sensors do not have to be related to force estimation. A crucial component in our methodology is the identification of relevant features that are used for learning the predictive models. Figure 2 show an overview of the approach.

The first step of the presented approach is to record training data representing the evolution of sensor values for the particular behavior. In contrast to our previous work [3], [2] no labeled data is recorded. Instead, the values of the FT150 are used as ground truth data for the actual force and torque. Next, the recorded data is analyzed for relevant features. As presented in [10], classical dimensionality reduction methods such as *Principal Component Analysis* fail to identify behavior-specific features. Instead, *Transfer Entropy* [5] (TE) is utilized to extract the most relevant features. In turn, these features are used to train two neural network models. The first model is used to predict the natural, behavior-related dynamics and is therefore called intrinsic FT-model. The second network is trained to predict total FT-values acting on the robot (intrinsic dynamics + external perturbations) and is called total FT-model.

The difference between total and intrinsic prediction is used to estimate the magnitude and direction of external perturbations applied by a human or through a collision. Since both models require only robot sensor data to predict the actual FT-values, no expensive and heavy FT-sensor is required during runtime. In the following, each step of the presented approach will be explained in more detail.

A. Data Acquisition

The realtime interface of the UR5 provides 104 different sensors, containing a variety of different information sources, e.g., the mainboard voltage, control, target and actual joint values. Some of these sources are relevant to the FT prediction task while others are redundant or non-informative. Hence we first identify a smaller set of relevant, informative sensors. To this end, the robot sensor values $\mathbf{s} = (s_1, \ldots, s_{104})$ and the FT-values $\mathbf{f} = (f_1, \ldots, f_6)$ are recorded during the execution of a behavior. Since the FT150 data rate is less than the UR5's $125Hz$, FT-values need to be interpolated to generate intermediate values. We use *Dynamic Mode Decomposition* (DMD) [11], a nonlinear interpolation method as described in [2]. Additionally, the time index r is recorded. During behavior execution the time index increases and is reset after each repetition. This time index will be relevant for the later feature selection step of the intrinsic FT-model.

For all following experiments a pick and place behavior was executed on the robot for 60 seconds. During this process, the behavior was repeated about 16 times. The recorded data $\mathbf{R} = ((\mathbf{s}, \mathbf{f}, r)_1; \ldots; (\mathbf{s}, \mathbf{f}, r)_n)$ represents training data consisting of $n = 7500$ equidistant samples with no external perturbations. In addition, two other data sets $\mathbf{T1}$ and $\mathbf{T2}$ with a length of 60 seconds were recorded. In contrast to the training data, $\mathbf{T1}$ was perturbed by a human during the last half of its execution, while $\mathbf{T2}$ was continuously perturbed.

B. Model Learning

In the following section, different neural network architectures for dynamical systems modeling are used to model the recorded robot dynamics. More specifically, a *feed-forward* (FF), a *time-delay* (TD), a *recurrent* (REC), and a *nonlinear autoregressive network with exogenous inputs* (NARX) are employed. For the sake of reproducibility, each network was generated with one hidden layer containing six neurons. Training was performed with the Levenberg-Marquardt method. In order to allow for a dynamic response, all but the FF network require setting a temporal delay parameter d. The TD network has a delay on the input weights while the REC network layer has a delayed recurrent connection to itself. In addition to delayed input weights, the NARX network makes use of previous predictions. For a more detailed description of the different neural network architectures, the reader is referred to [12]. All network architectures have been trained to map input data $\mathbf{s}$ to output data $\mathbf{f}$.

C. Intrinsic FT-Prediction

In the following section, all neural networks are trained with $\mathbf{R}$ and a delay of $d = 2$. The goal is to predict the intrinsic state for the semi-perturbed data set $\mathbf{T1}$. First, the neural networks is trained using all sensors. The resulting predictions for f_2 can be seen in Figure 3. As long as no perturbation occurs, all networks are approximately predicting the correct value. However, during the second half of $\mathbf{T1}$ all networks fail to predict the correct intrinsic state. This is due to the fact that the 104 input sensors contain non-relevant information that may obfuscate important features. Hence,

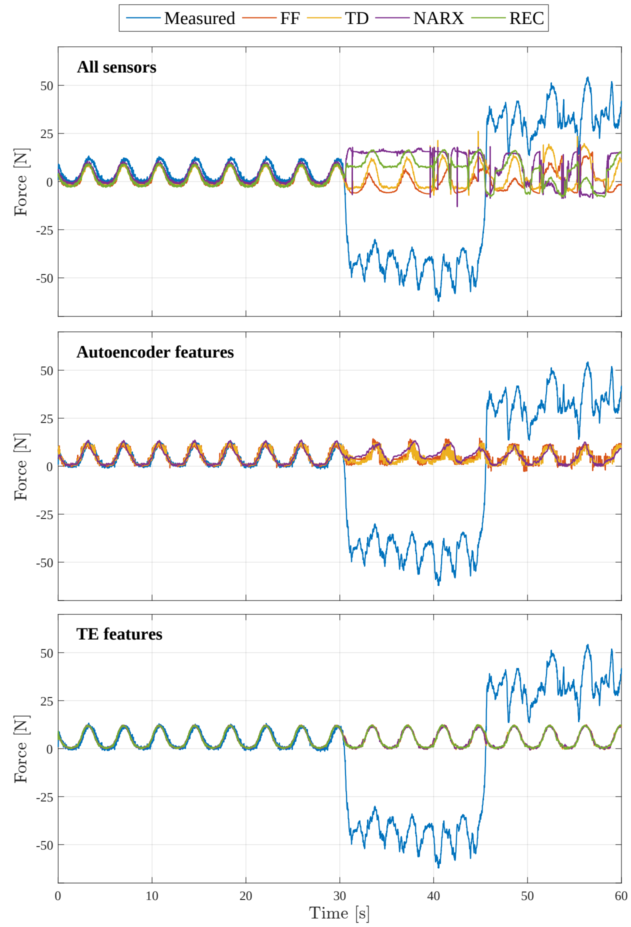

Fig. 3. The different predictions of $f_2 \in \mathbf{T1}$ for different neural network architectures trained with the data set $\mathbf{R}$. Using all sensors (top), autoencoders (middle), or TE feature selection (bottom) influences the accuracy of the predictions. Only the sensors selected by utilizing TE was able to suppress the external perturbations in the second half of the data set.

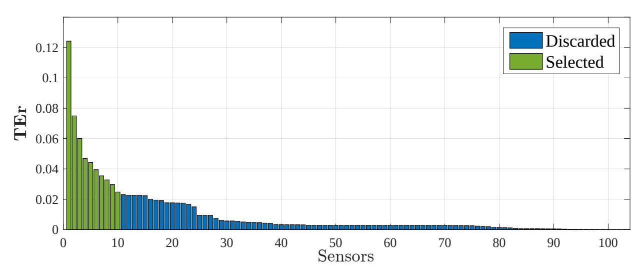

Fig. 4. 10 sensors (highlighted green) with 50% of the overall TE are selected as features for training. The remaining 94 sensors (highlighted blue) are classified as less important for predicting the actual phase and therefore being discarded.

feature extraction methods are needed to identify relevant features.

A feature extraction approach that is gaining popularity, is the use of autoencoders [4]. An autoencoder consists of an encoder and a decoder component, both of which are neural networks.

Broadly speaking, the encoder maps the input data to a smaller set of hidden neurons while the decoder tries to reconstruct the original input. This process can be stacked to reduce the dimensionality of the input data through a stepwise layering. Using autoencoders, we reduce 104 sensors to 50 values using the first encoder, and then to 10 values by stacking a second encoder. For the encoding process, a logistic sigmoid function was used while a linear transfer function was utilized for the decoders. As a result, the 104 dimensional input data **s** is reduced to a 10 dimensional feature data set. Next, this feature data was used to train the neural networks. The resulting predictions for $f_2 \in \mathbf{T1}$ can be seen in Figure 3 middle. Especially during the perturbation phase, the overall accuracy increases since irrelevant information is discarded. However, a problem of autoencoders is that the *temporal influence and correlation of variables* is not taken into account.

To predict the time-dependent intrinsic values of the FT-sensor the actual phase of the behavior is taken into account. TE is used as a measure of predictability and information flow between the robots sensor values and the relative time by $\mathbf{TEr} = TE(\mathbf{s}, r)$. The TE from $\mathbf{J}$ to $\mathbf{I}$ is defined as

$$TE(\mathbf{I},\mathbf{J}) = \sum_{i\in\mathbf{I},j\in\mathbf{J}} p(i+1,i,j)log_2\frac{p(i+1|i,j)}{p(i+1|i)},$$

where $(i_1,\ldots,i_q) \in \mathbf{I}$ and $(j_1,\ldots,j_q) \in \mathbf{J}$ are the possible states of quantized time-series data and the function $p(\cdot|\cdot)$ describes the conditional probability. For a more detailed derivation the interested reader is referred to [5]. The resulting $\mathbf{TEr}$ describes how strong each sensor of the robot is influenced by the relative time. Sensors with a high TE are assumed to be beneficial for predicting the relative time and therefore the actual phase of the behavior. Figure 4 shows the normalized and ordered TE values of $\mathbf{TEr}$. As can be seen, only 10 sensors with at least 50% overall TE are selected for training the neural networks. These sensors are, in particular, target and control values of the robot's joint states (e.g. position and velocity), which are independent of external influences and therefore are good predictors for the actual phase of the behavior. The resulting predictions for this subset of sensors can be seen in Figure 3 bottom. In contrast to the previous results, the selected sensors are able to accurately predict the intrinsic fluctuations. Also during external perturbations the predicted intrinsic forces are not influenced. The difference between the actual measured total FT-values and the intrinsic ones can be used to estimate the magnitude and direction of an external perturbation.

D. Total FT-Prediction

In contrast to the prediction of the intrinsic FT-values explained in the previous section the total FT-values are not exclusively dependent on the state of the robot. Consequently, it is important that the neural networks are additionally trained with information about how external perturbations influence the sensors of the robot. To this end, the neural networks are trained with $\mathbf{R}$ and the perturbed data set $\mathbf{T2}$. Furthermore, feature selection is not calculate with respect to relative time anymore. Instead, the TE is calculated between the robots sensor values and each FT-value by $\mathbf{TEf} = |TE(\mathbf{s}, \mathbf{f})|$.

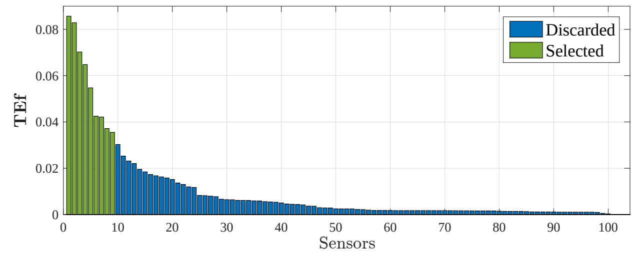

Fig. 5. 9 sensors (highlighted green) with 50% of the overall TE are selected as features for training. The remaining 95 sensors (highlighted blue) are classified as less important for predicting the actual total FT-values and therefore being discarded.

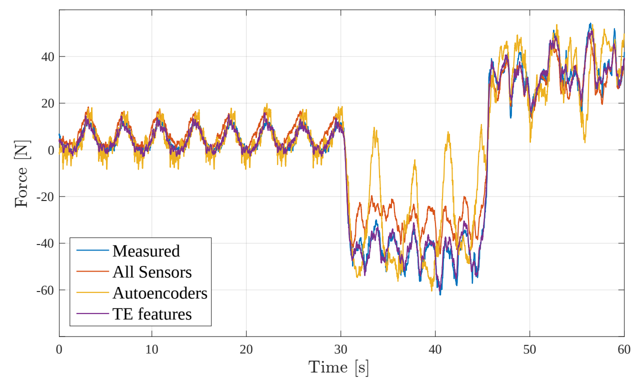

Fig. 6. The different predictions of $f_2 \in \mathbf{T1}$ for a recurrent neural network which was trained with the data sets $\mathbf{R}$ and $\mathbf{T2}$. Using all sensors (red), autoencoders (yellow), or TE feature selection (purple) influences the accuracy of the predictions.

Figure 5 shows the normalized and ordered TE values of $\mathbf{TEf}$. As can be seen in the figure, a small set of 9 sensors contains more than 50% of the overall TE information. These are especially measured values (e.g. current values close to the TCP) and no more target and control values. This is due to the fact that external perturbations does not perturb target/control values but the measured actual ones instead. The resulting mean squared errors for the different network predictions of $\mathbf{f} \in \mathbf{T1}$ are shown in Table I. As can be seen, utilizing TE for feature selection outperforms the usage of autoencoders or all sensors. Furthermore, for NARX and REC networks using autoencoders further deteriorates the results. A possible explanation for this effect is that the objective function of autoencoders only focuses on the amount of information retained by using the generated features. This may be detrimental in various physical tasks in which some

TABLE I
THE MEAN SQUARED ERRORS RESULTING FROM DIFFERENT INPUTS AND NEURAL NETWORK ARCHITECTURES.

	All Sensors	Autoencoder	TE Features
FF	155.24	72.62	25.14
TD	95.48	81.45	14.82
NARX	22.43	62.34	7.33
REC	13.74	42.48	**3.41**

sensors have limited variability but strong influence on the task. The best result is obtained by using TE features and the REC network architecture. Predictions of $f_2 \in \mathbf{T1}$ generated by the REC network are compared to using all sensors and autoencoders in Figure 6. Given these results, the following experiment utilizes the proposed TE feature selection method combined with recurrent neural networks.

IV. Experiments

Different experiments have been conducted to validate the proposed method. To this end, the data set **T1** introduced in Section III-A was used.

A. Accuracy of Estimates

First, the RECo model is evaluated by investigating the mean absolute error (MAE) in comparison to ground truth data of the FT150 and an intrinsic FT-sensor included in the firmware of the UR5. This sensor makes use of joint torques and a kinematic model of the robot to predict the FT-values at the TCP. In order to get comparable results, the firmware was calibrated to the mass and size of the FT150. For the FT-sensor, the manufacturer specifies a force accuracy of $25\,\mathrm{N}$ at the TCP and a detection time of $250\,\mathrm{ms}$. The different force estimates for each dimension can be seen in Figure 7. The estimates provided by the firmware sensor follow the general trend but exhibit significant noise. By contrast, the estimates of the RECo model are close to the ground truth data of the FT150. The accuracy of the firmware sensor resulted in a MAE of $26.7401\,\mathrm{N}$ which is slightly below the $25\,\mathrm{N}$ specified by the manufacturer. In addition, considering the mentioned detection time delay of $250\,\mathrm{ms}$ did not decrease the MAE score. In comparison, the MAE of the RECo model is $3.8336\,\mathrm{N}$. Furthermore, the detection time is less than $10\,\mathrm{ms}$ (on a dual core with $3.2\,\mathrm{GHz}$) and scales with the performance of the computer system. Increasing the network delay from $d = 2$ to $d = 125$ further reduces the MAE to $1.9417\,\mathrm{N}$. Consequently, the model requires one second of continuous sensor data to start the prediction while the detection time slightly increases to $15\,\mathrm{ms}$. However, in order to keep the robot reactive from the beginning, a minimal delay of $d = 2$ was used for all experiments. Additionally, the RECo model also provides better torque estimates $(1.0761\,\mathrm{N\,m})$ when compared to the firmware sensor $(5.6735\,\mathrm{N\,m})$.

B. Perturbation Estimation

The intrinsic forces need to be predicted in order to dissect external perturbations from the predicted total forces. For this task, the RECi model is used. Figure 8 shows the resulting intrinsic force predictions. As can be seen in Figure 8, the intrinsic forces are not affected by perturbations in the second half of the recording. Finally, the perturbation value shown in Figure 9 is defined as the difference between total and intrinsic FT-values. The length and direction of the perturbation value is used to estimate the magnitude and direction of external perturbations. Generated estimates only represent the external perturbations applied by humans,

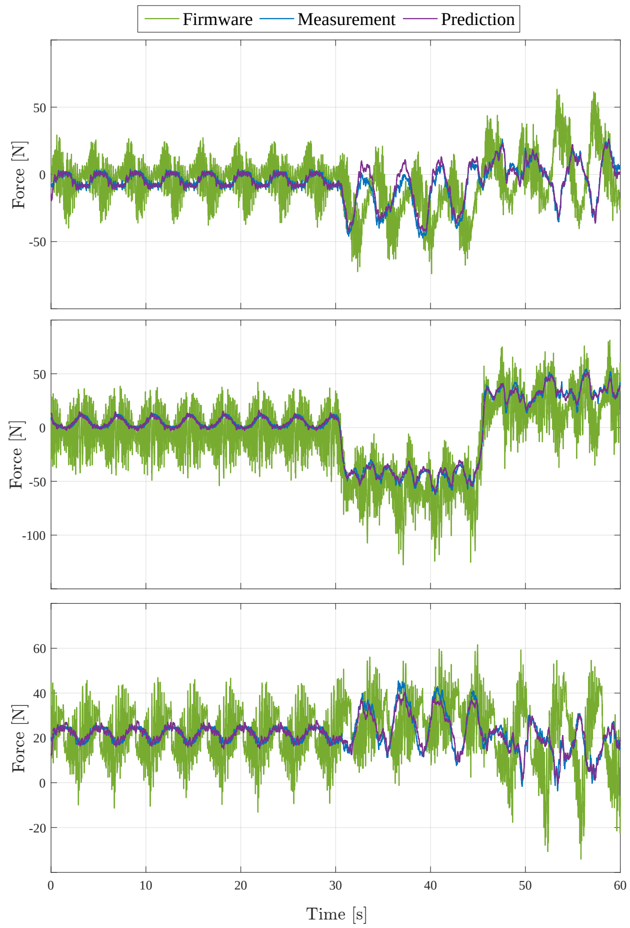

Fig. 7. The x (top), y (middle) and z (bottom) force values of the firmware FT-sensor (green), the FT150 sensor (blue), and the prediction of the learned model (purple). The predicted values provide a tighter approximation of the measured ground truth data.

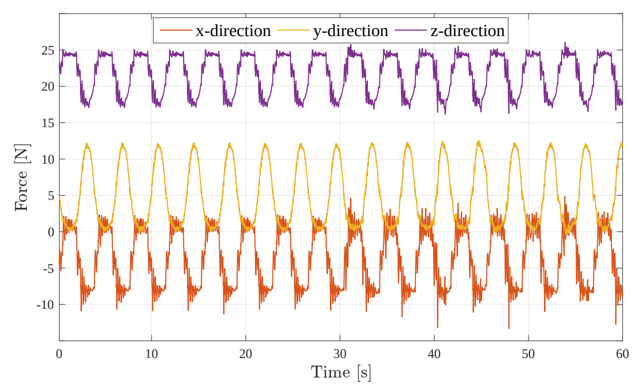

Fig. 8. The intrinsic force predictions are not influenced by the external perturbations.

collisions or other external factors. As a result, the proposed method can be applied during runtime without making use of a FT-sensor in order to estimate total, intrinsic and in consequence external FT-values from previous experience. A video of some further experiments can be found here [1].

[1]`https://youtu.be/60ue0X25S6k`

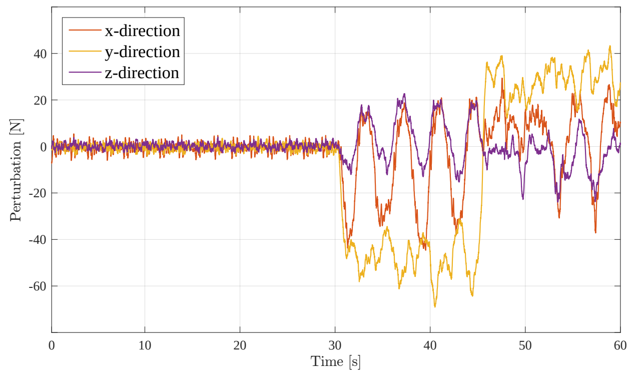

Fig. 9. The difference between the total and intrinsic forces represent the perturbation value which is the FT-value applied to the robot from its environment.

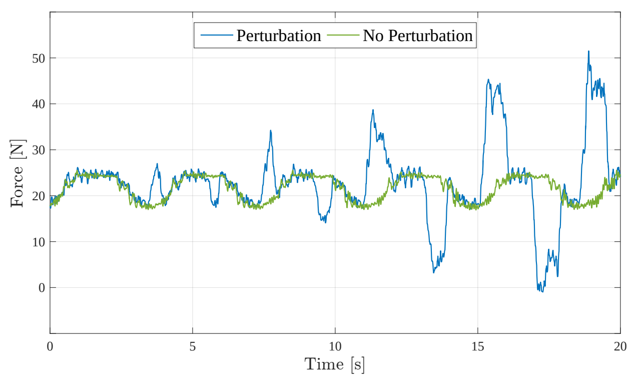

Fig. 10. Non-perturbed and perturbed execution of a pick and place behavior.

C. Discussion

As can be seen in Figure 7 (bottom), the force applied on the robot is estimated to be about 23 N. This is due to the weight of the gripper mounted on top of the FT-sensor – the manufacturer specifies a weight of 2.3 kg. A strength of the presented approach is that, no prior knowledge of the robots kinematics, dynamics, sensor instrumentation, or parameters is required. As a result, our approach can quickly be applied to any kind of robot platform. In addition, no hard thresholds for detecting external perturbations need to be set. Figure 10 illustrates this point. The green trajectory shows the force readings during a normal execution of a pick and place behavior. As can be seen, forces of 22 ± 3N are generated. In contrast to that, the blue trajectory shows an execution with different degrees of perturbations.

V. Conclusion

In this paper, we have described a methodology for *estimating forces from experience* and showed how to use neural network models in order to generate accurate estimates of external perturbations. The main advantage of the presented approach is that the learned models are able to map low-cost robot sensor data to accurate FT measurements. Hence, no FT-sensor is required during runtime and consequently the robot's weight is reduced. This is particularly beneficial for light-weight manipulators with a limited payload.A further strength of the presented approach is that no prior knowledge of the robot kinematics, dynamics, or sensor characteristics is required. As a result, the approach generalizes to arbitrary robot platforms. We have shown that, without further adjustment, usual neural network architectures produce adequate estimates of the intrinsic, external, and total FT-values. Adapting the network properties, for instance by increasing the network delay, could further increase the estimation accuracy. A limitation of the approach is that the robot needs to perform the same behavior during training and runtime estimation. However, early results on the generalization capability of this approach show that it generalizes to mild variations of the behavior. A more in-depth evaluation of the generalization ability will be conducted in future work.

References

[1] A. Colome, D. Pardo, G. Alenya, and C. Torras, "External force estimation during compliant robot manipulation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3535–3540.

[2] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. B. Amor, "Estimation of perturbations in robotic behavior using dynamic mode decomposition," *Advanced Robotics*, vol. 29, no. 5, pp. 331–343, 2015.

[3] E. Berger, S. Grehl, D. Vogt, B. Jung, and H. Ben Amor, "Experience-based torque estimation for an industrial robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[4] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[5] T. Schreiber, "Measuring information transfer," *Physical Review Letters*, vol. 85, no. 2, pp. 461–464, 2000.

[6] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The KUKA-DLR lightweight robot arm - a new reference platform for robotics research and manufacturing," in *ISR/ROBOTIK 2010, Proceedings for the joint conference of ISR*. VDE Verlag, 2010, pp. 1–8.

[7] E. Magrini, F. Flacco, and A. De Luca, "Control of generalized contact motion and force in physical human-robot interaction," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 2298–2304.

[8] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans." in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 4326–4332.

[9] A. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[10] E. Berger, D. Müller, D. Vogt, B. Jung, and H. Ben Amor, "Transfer entropy for feature extraction in physical human-robot interaction: Detecting perturbations from low-cost sensors," in *Humanoids'14*, 2014.

[11] M. R. Jovanovi, P. J. Schmid, and J. W. Nichols, "Sparsity-promoting dynamic mode decomposition," *Physics of Fluids (1994-present)*, vol. 26, no. 2, 2014.

[12] M. Nrgaard, O. E. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2000.