Real-time IMU-Based Learning: a Classification of Contact Materials

Carlos Magno C. O. Valle*, Alexander Kurdas*, Edmundo Pozo Fortunić,
Saeed Abdolshah and Sami Haddadin

***Abstract*— In modern highly dynamic robot manipulation, collisions between a robot and objects may be intentionally executed to improve performance. To distinguish between these deliberate contacts and accidental collisions beyond the limit of state-of-the-art human-robot interactions, new sensing approaches are required. This work seeks an easy-to-implement and real-time capable solution to detect the identity of the impacted material. We developed an inertial measurement unit (IMU) based setup that records vibration signals occurring after collisions. Furthermore, a data-set was generated in an unsupervised learning manner using the measurements of collision experiments with several materials commonly used in realistic applications. The data-set was used to train an artificial neural network to classify the type of material involved. Our results show that the neural net detects collisions and a detailed distinction between materials is achieved, even with estimating different human body parts. The unsupervised data-set generation allows for a simple integration of new classes, which provides broader applicability of our approach. As the calculations are running faster than the control cycle of the robot, the output of our classifier can be used in real-time to decide about the robots reaction behavior.**

I. INTRODUCTION

Traditionally collisions between a robot and its environment were strictly avoided to prevent damages to the environment or the robot itself. However, in recent highly dynamic robot manipulation settings, collisions between robots and objects are encouraged in order to speed-up manipulation. Therefore, new sensing modalities to evaluate the process status are needed. In this work, we are specifically interested in the vibrations occurring after an impact between a robot and an object, similarly to the well-known modal analysis in mechanics, but coping with much more disturbances due to dynamic robot motion and contact richness. So far, similar principles, based on proprioceptive sensors, are used in physical human-robot interaction (pHRI) to evaluate the human's intention, like intentionally grasping and guiding the robot, or accidentally being hit by the robot [1]. However, even with recent tactile robots, only moderate frequency components can be measured due to sensor bandwidth limitations to a sampling rate of 1 kHz.

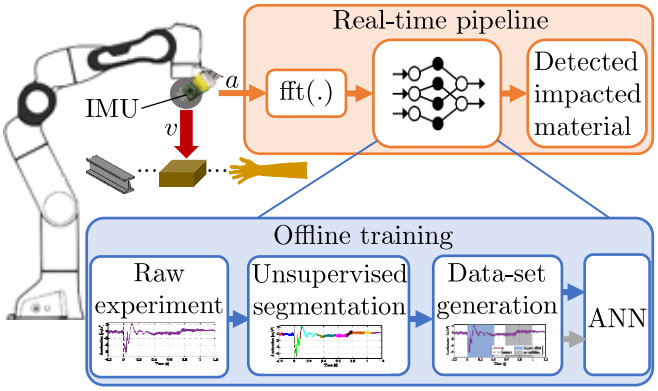

Fig. 1. Overview of the proposed concept. We impacted various types of materials, including also a human body parts surrogate with a robot and recorded the vibration measurements of an inertial measurement unit (IMU) at the end-effector. With the frequency information an artificial neural network (ANN) is trained offline (blue area) to classify the impacted material. This ANN is then used in a real-time pipeline for impact evaluation (orange area).

We aim for extending these approaches by means of the robot's intention. An illustrative example would be that the robot should put an item into its packaging box. It could happen that the robot hits the soft packaging material due to small positioning inaccuracies, which requires a re-try at a slightly different pose. It might also happen that the box is closed and hence the hard lid of the box is hit, which means it firstly needs to be opened. Or in a pHRI scenario the robot might even hit a human arm, which requires an emergency stop. All these cases may happen at the same robot pose and may in principle lead to the same absolute external force, so no distinction by state-of-the-art methods would be possible. In spite of that, all cases do differ in the vibration of the end-effector after the collision, due to the different impacted materials.

In this work, we propose a novel system to classify the impacted material by a robot in real-time; a schematic is shown in Fig. 1. Our approach consists of an inertial measurement unit (IMU) integrated into the end-effector, which measures vibrations with a high bandwidth. This generates features used by a small neural network that is trained to classify several types of materials from these vibration signals. Our proposed offline training pipeline can generate a training data-set from raw collision experiments with different materials in an unsupervised manner, allowing for a more flexible applicability of the method with minimal human labeling requirements. The material detection is achieved within the

* The authors contributed equally to this paper as first authors.

All authors are with the Technical University of Munich (TUM), Munich Institute of Robotics and Machine Intelligence (MIRMI), Chair of Robotics and System Intelligence (RSI), Germany, carlos.valle@tum.de .

We greatly acknowledge the financial support of Vodafone, the funding of the EU Horizon 2020 projects I.AM. (no. 871899) and Darko (no. 101017274), the Lighthouse Initiative KI.FABRIK Bayern by StMWi Bayern (KI.FABRIK Bayern Phase 1: Aufbau Infrastruktur and KI.Fabrik Bayern Forschungs- und Entwicklungsprojekt, grant no. DIK0249), the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant no. IUK-1807-0007// IUK582/001), and the LongLeif GaPa gGmbH (Project Y). Please note that S. Haddadin has a potential conflict of interest as shareholder of Franka Emika GmbH.

real-time control loop of the robot, with a simple setup, not requiring expensive specialized equipment or software.

In Sec. II we discuss related works that influenced our approach. The data-set generation process and the used methods are presented in Sec. III. In Sec. IV, we outline the hardware architecture. In Sec. V, we evaluate and discuss our performance in different experiments. Finally, Sec. VI concludes the paper.

II. RELATED WORKS

The proprioceptive joint torque sensors are often used for evaluating impacts, especially in pHRI scenarios with tactile serial kinematic robots [1]–[7]. An overview of methods, like e. g. the momentum observer for estimating the external torques, are given in [8]. Among these references many use artificial neuronal network (ANN) methods to classify the impacts, except [2] and [3], where they purely use threshold based approaches. Furthermore, the frequency content of the input signals play a major role for classification [2], [3], [7]. However, due to bandwidth limitations of the build-in sensors and estimates, it is not possible to investigate impacted material properties based on these methods. On the other hand, detection of material properties by sensing vibrations of impacts is commonly used in mechanics, where it is called modal or vibration analysis. Here, specialized, hence expensive, equipment is used and commonly, no additional disturbances, due to e. g. motors, are present. Modal analysis has already been combined with ANN for material classification in [9]. The combination of modal analysis and ANN was already transferred to robotics in [10], whereas the robot was again stand-still to reduce motor disturbances in these experiments. Additionally, the used classification approach focused on offline generated frequency characteristics of the acquired signals which prohibits analysis in real-time. Whereas, Fast Fourier Transformation (FFT) is in-general of major importance for vibration analysis, hence we use the real-time capable FFT provided in [11]. The similarity between vibration and sound can also be used to evaluate impacts on robots [12], [13], but requiring to attach microphones on the robot which is not suitable to capture impacts with soft materials that do not generate sound. Moreover, capacitive sensors can be used for contactless material detection [14]. However, this can only be achieved with a limited field of view. Additionally, with high-sensitivity haptic force sensors [15] or tactile skins [16] it is also possible to draw conclusions about touched materials, by applying similar vibration analysis and ANN methods. Instead of these highly specialized systems we may use a standard Micro-Electro-Mechanical System (MEMS) IMU. This is already often used in robotics, not only for impact detection [10], [17]–[19], but also for improving robot state measurements [20], localizing impacts on the robot surface [10], [21] or estimating impact surface orientation [22].

III. METHODOLOGY

Our approach consists of a neural network architecture trained to classify several types of materials from collision experiments. We extract input features from the acceleration readings that allow the classifier to distinguish materials in real-time. This allows the possible use of specific reactive behaviours. In this section we show the experimental protocol for generating the training data in Sec. III-A. Furthermore, we describe in the following sections the step-by-step processing of this experimental data from raw acceleration measurements to classified materials.

TABLE I
EXPERIMENTAL PARAMETERS

Material	foam, cardboard box, wood, metal, plastic box, hand (PRMS), lower arm (PRMS), upper arm (PRMS)
Location	C2, C3
Direction [°]	90, 75, 60, 45, 30
Velocity [m/s]	0.10, 0.15, 0.20, 0.25

Fig. 2. Materials and surrogate system (PRMS) configured for lower arm.

A. Experimental protocol

The experiment starts with the robot in a *static* pose. It accelerates until a *constant (moderate) velocity* is reached and moves further in a straight Cartesian line until it *collides* with a flat surface in a predefined location. The robot then *returns* to the initial static position. We performed experiments with nearly all combinations of impact parameters given in Tab. I, whereas each combination was repeated 20 times, making 6000 impacts. Only the velocity of $0.25\,\mathrm{m/s}$ was not used for wood and metal, to prevent damage of the robot. Photos of the materials are provided in Fig. 2. The experiments with human extremities were conducted with the Pilz Robot Measurement System (PRMS)[1]. This is a test device to verify safety for humans in pHRI scenarios according to ISO 15066 [23]. The PRMS can be configured with several stiffness and damping elements to behave like different human body parts in constrained contacts, following the average values given in ISO 15066 [23]. The impact locations are chosen according to ISO 9283 [24] as $\mathrm{C2} = [498, 0, 252]^\mathsf{T}\,\mathrm{mm}$ and $\mathrm{C3} = [498, 0, 452]^\mathsf{T}\,\mathrm{mm}$ for a $400\,\mathrm{mm}$ cube, see Fig. 3. The impact angle for the different directions is measured with respect to the impacted surface, whereas this was always parallel to the (x-y)-plane of the robot base.

B. Pre-processing

We recorded 3D acceleration measurements ${}^{\mathcal{I}}\boldsymbol{a}_{\mathrm{meas.}}$ for several variations of possible collision scenarios, which are measured in the sensors own coordinate frame $\mathcal{I}$. As accelerometers measure also the static gravitational acceleration of earth, it is important for the features used in the

[1] https://www.pilz.com/de-DE/produkte/robotik/prms/prms

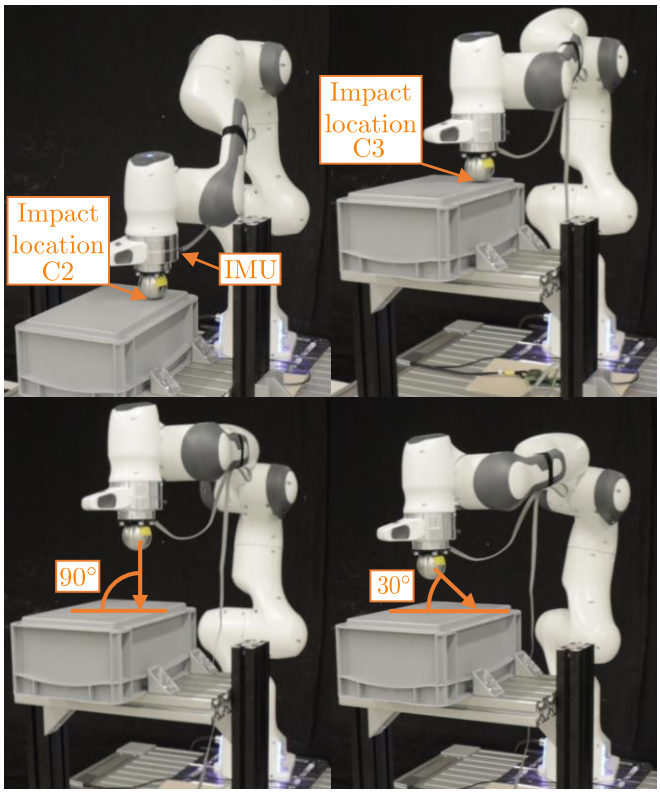

Fig. 3. The impact locations and the maximum and minimum impact directions. The IMU is attached between the flange and the metal ball as end-effector.

classifier to have this effect removed. One way of achieving this could be to apply high-pass filtering of the data to remove the static offset. However, such a filter might also influence low-frequency components of collisions, in which we are specifically interested for softer materials. Instead of using the high pass filter, we can subtract the base gravity vector ${}^{\mathcal{O}}\boldsymbol{g}$ from the measurements with

$$ {}^{\mathcal{I}}\boldsymbol{a}_{\text{comp.}} = {}^{\mathcal{I}}\boldsymbol{a}_{\text{meas.}} - ({}^{\mathcal{O}}\boldsymbol{T}_{\mathcal{E}}\ {}^{\mathcal{E}}\boldsymbol{T}_{\mathcal{I}})^{-1}\ {}^{\mathcal{O}}\boldsymbol{g}\,, \tag{1} $$

given the forward kinematics of the robot ${}^{\mathcal{O}}\boldsymbol{T}_{\mathcal{E}}$ and the location of the accelerometer relative to the end-effector of the robot ${}^{\mathcal{E}}\boldsymbol{T}_{\mathcal{I}}$, whereas ${}^{\mathcal{I}}\boldsymbol{a}_{\text{comp.}}$ denotes the compensated acceleration vector, see Fig. 4. Since the collision impact direction can interfere on which dimension of the 3D accelerometer the impact can be observed, we compute the magnitude a of the vector ${}^{\mathcal{I}}\boldsymbol{a}_{\text{comp.}}$ by

$$ a = \sqrt{a_{x,\text{comp.}}^2 + a_{y,\text{comp.}}^2 + a_{z,\text{comp.}}^2}\,, \tag{2} $$

decoupling this effect from our measurements.

The input features used by the classifier are extracted from a moving window of past measurements of the accelerometer. Since it operates at 10 kHz, we experimented with window sizes for past samples from the interval of 0.1 s to 1 s. We selected the best trade-off between performance and computation time which was 4096 samples (0.4096 s). Window sizes that are powers of 2 ($4096 = 2^{12}$) are computationally more efficient, speeding up our features calculation [11].

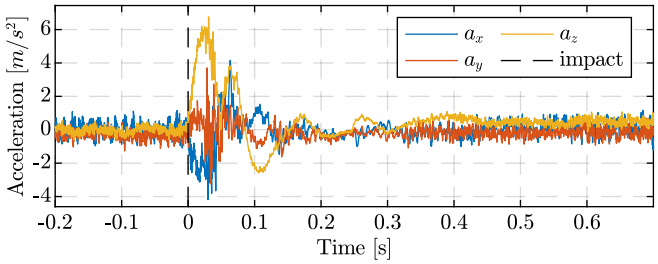

Fig. 4. Exemplary gravity compensated accelerometer data for the x-, y-, and z-axis. The same experiment is also shown in Figs. 5 and 9 for comparison.

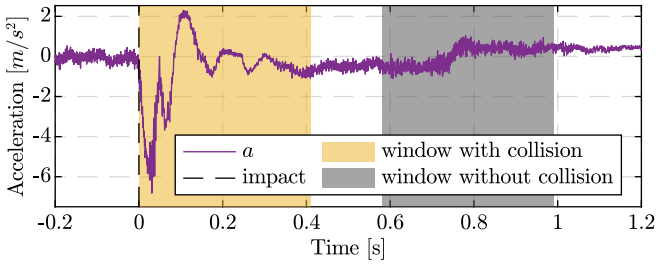

Fig. 5. Input accelerometer data for the ANN, where the yellow area denotes a time-window classified as with impact, and the gray window is a "no collision" sample.

Furthermore, typical vibrations after impacts show also a similar time window, see the yellow area in Fig. 5. We compute our features at every 10 steps, which means our classifier operates at 1 kHz. This is the same frequency that the robot control loop operates in, allowing the system to react as soon as possible for a given collision scenario. Taking into consideration that each experiment lasts for around 3 s, the average total number of windows samples is 2600. The total number of sampled windows for all tested experiment variations is roughly 15.6 million windows to be classified.

At every sampled window we compute the FFT, with 10 kHz sampling rate, and extract the frequency amplitude spectrum of it. We chose to use the amplitudes until 512 Hz as input for our classifier, as [9] reported a similar range. Similarly, we also observe that frequencies above this threshold are consistently smaller in the samples used for training. As considering also higher frequencies would increase the amount of computations required by the algorithm, we choose to limit the regarded frequencies to 512 Hz, which can also be seen as low-pass filtering.

1) Data-set generation: We opt to use supervised learning to train the classifier, therefore requiring corresponding output labels for every input sample. For every experiment recorded for training it was known a priori the corresponding material that was collided, we therefore needed to automatically distinguish samples corresponding to the collision period from the ones that did not. For this task, we applied the K-means clustering method [25], which aggregates samples by similarity in clusters. We chose the number of cluster based on the number of possible motion states described in

Sec. III-A (static, acceleration, constant velocity, collision, retraction and deceleration), more precisely 4 times that number (24 clusters). This multiplier is necessary because the transition between each state of the experiment is not instantaneous, therefore additional clusters to separate transitions samples are necessary. The exact multiplier for the clusters number needs to be bigger than 3 (to account for the state and the transition period from and into another state) and also low enough to avoid single sample clusters, which would denote over-fit. To distinguish samples from the collision period we simply accounted the average acceleration magnitude of each cluster and the one with the highest was used to draw samples for the collision and labeled accordingly to the material involved in the experiment and an equivalent number of samples of the other cluster were chosen randomly as examples of non collision periods, see also Fig. 5.

C. Classification

The selection of the classification method took into consideration scalability of the method to more samples and more classes (for future expansion of the number of classes, for example) and computation efficiency. We opted to use a fully connected feed-forward neural network since it is fast to compute (compared to more complex architectures like using LSTMs [26]) and, potentially, scales well for more classes (unlike SVM-based approaches [27]). We used the data-set to train a fully connected neural network with a single hidden layer with 25 neurons with Softplus activation function and bias. The output layer used a Softmax activation function with bias and outputs a one-hot representation of the corresponding class. The depth and number of neurons was empirically selected for minimizing the number of calculations required without reducing performance and might be subject to change depending on different classification applications. It is important to highlight that more complex neural network architectures could be used but in early experiments we empirically did not observe accuracy improvements to justify the additional computational cost. As an example, our fully connected architecture already consists of approximately 13000 trainable weights (512 inputs, 25 hidden neurons and 9 classses), which denotes clearly that any growth on complexity can vastly increase the number of calculation needed for a real-time inference, therefore, limiting the hardware capable of handling it.

1) Training: As each experiment combination was repeated 20 times, 14 would be used for training the classifier, 3 for validation and 3 for testing. We used gradient descent on the training data and used the validation data as early stop metric. The network was trained via stochastic gradient descent using Adam optimizer [28] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and initial learning rate of 10^{-4}. The early stop was 200 steps. Training was done in Python 3.8.12 using the Pytorch[2] [29] library version 1.10.1 (under BSD license). Training was optimized for GPU and conducted in a Nvidia RTX 2080ti.

2) Usage: The trained neural network weights, biases and activation functions were then transcribed in C++ for real-time experiments. For each sampled window we have to calculate the FFT, its frequency amplitude spectrum, and multiply the corresponding frequencies for the neural network weights, biases and activation functions. This results in an estimate of the impacted material, with which a proper robot reaction strategy can be started if needed. However, due to not including the transition phases into the neural net, we observe a high number of wrong classifications at the beginning and end of a detection phase. As the class is certain at the end of a classification phase, we introduced to prolong the output of the class for $50\,\mathrm{ms}$, to avoid further misclassifications.

IV. EXPERIMENTAL SETUP

All experiments were done with a Franka Emika Robot with a metal sphere of $50\,\mathrm{mm}$ diameter attached to the end-effector. The robot was commanded Cartesian velocity inputs via the Franka Control Interface (FCI). All robot settings were set to default values, except the collision detection thresholds which are set to $[20, 20, 18, 18, 16, 14, 12]\mathrm{Nm}$ for the joints and $[20\mathrm{N}, 20\mathrm{N}, 20\mathrm{N}, 25\mathrm{Nm}, 25\mathrm{Nm}, 25\mathrm{Nm}]$ for the Cartesian impact settings.

The acceleration data was acquired using an on-board TDK ICM-42688-P MEMS IMU sensor, which was rigidly attached close to the flange. The full-scale range and the Output Data Rate (ODR) frequency of the sensor were set to $\pm 16\,\mathrm{g}$ and $32\,\mathrm{kHz}$, respectively. All internal filters of the IMU were deactivated. In order to extract the sensor data, a real-time Data Acquisition System (DAQ) was implemented using a Microchip EVB-LAN9252-SAM D51 board and a custom-made Printed Circuit Board (PCB) shield for connectivity. The on-board SAMD51 micro controller handles the online configuration of sensor parameters via a bidirectional SPI bus and acquires the data of up to four sensors and six channels (three for the accelerometer and three for the gyroscope) per sensor at $10\,\mathrm{kS/s/ch}$. The data is reformatted from 16-bit signed integers to single-precision floating point SI values and synchronously re-transmitted to the on-board LAN9252 EtherCAT slave controller via the QSPI bus at the same sample rate. The data is then send on request of the EtherCAT Master. Simple Open EtherCAT Master (SOEM) software[3] was used to manage the EtherCAT bus, which was set to a bus frequency of $10\,\mathrm{kHz}$. To further process the data mathematically the Eigen C++ library [30] was used and for real-time FFT we used FFTw [11] with default settings. The PC was a generic office PC, with an Intel i5-9400 processor, $8\,\mathrm{GB}$ memory, and a Realtek RTL8168 network chip. A sketch of the setup is depicted in Fig. 6.

V. RESULTS AND DISCUSSION

The final precision of the neural net approach is evaluated offline based on parts of the prerecorded data that was not used in training (test data set). The evaluation over these

[2] https://pytorch.org/

[3] https://rt-labs.com/product/soem/

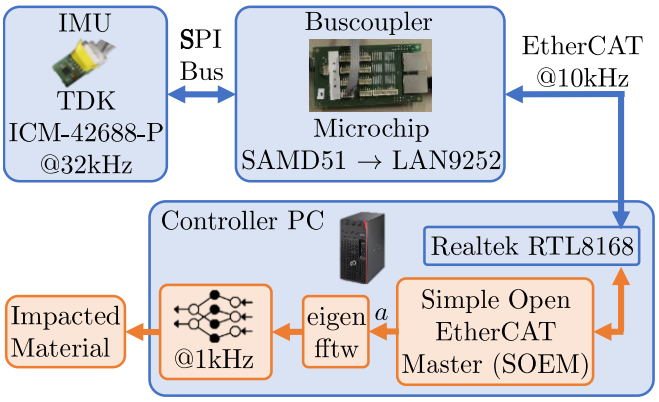

Fig. 6. Hardware (blue) and software (orange) setup.

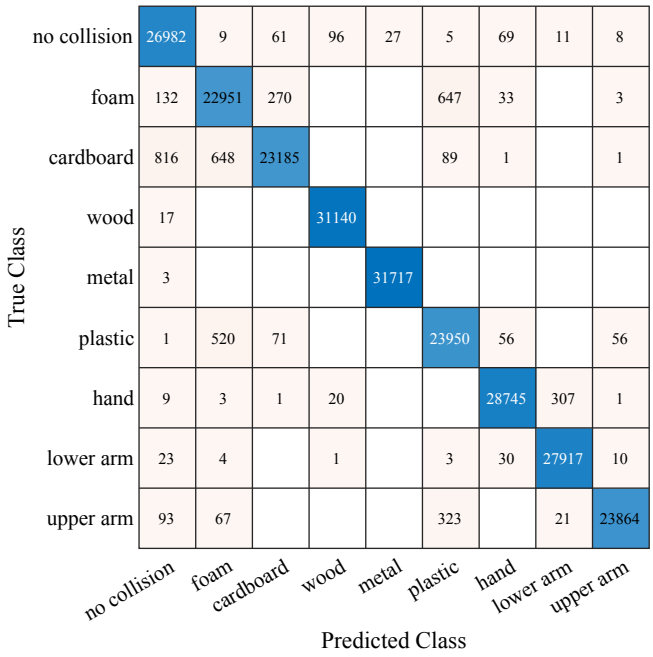

Fig. 7. Test data confusion matrix. The classes are presented in the order of Tab. I, whereas the "no collision" class is added on top.

900 impacts results in the lowest precision of 93.71 % for cardboard, see Fig. 8. We further also report the precision in training, so with already known data, and on the validation data set. The lowest class precision observed in the training data was 99.1 % for the "no collision" class and 96.31 % for foam in the validation set. Furthermore, we provide in Fig. 7 the confusion matrix of the test data set. The classes are presented in the order of Tab. I, whereas the "no collision" class is added on top. Even though, these evaluations were done on offline data, they show still the good capabilities of the approach, as the input data is recorded on the real setup. In the following we further elaborate on certain aspects of the application of the approach in real world scenarios.

In Fig. 9, we see the performance of the classifier in a real-time experiment using one of the trained variations of the experiment. It can be observed that the classification is mostly correctly done. It is also possible to observe a

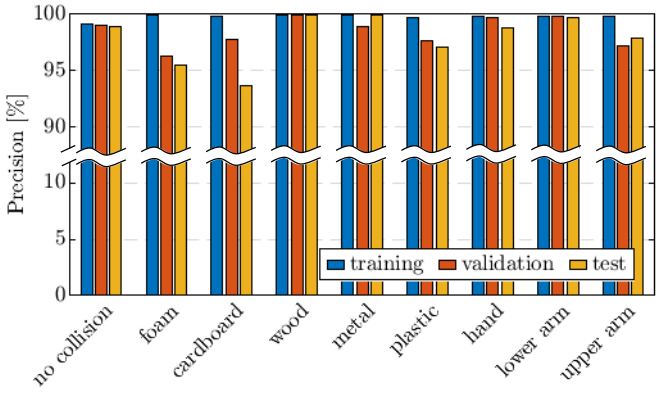

Fig. 8. The precision of the approach on different data sets. The lowest precision on the unknown test data set defines the final result, which is 93.71 % for cardboard. We also show for comparison the results of training (known data set) and validation.

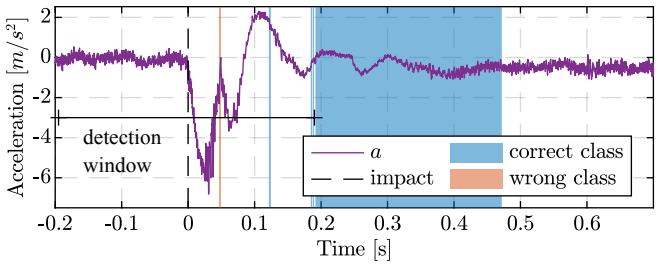

Fig. 9. Class prediction in real-time experiment with trained parameter variation. The blue section corresponds to correct material detected, red denotes wrong classification, whereas for the beginning of the confidence phase the marked detection window is used. Here, the rather soft plastic box was impacted with 0.2 m/s and 75°.

delay from the impact start until detection. This is because a certain number of samples inside of the sampling window is required. This minimum number of samples is inversely proportional to the resonance frequency of the material the robot is colliding with. Therefore, it is expected that a collision with metal (higher stiffness, hence higher resonance frequency) requires less samples to be detected than the one with plastic. Hence, in an example collision with metal, which can be seen in Fig. 10, the approach shows much faster recognition with higher vibration frequency. Furthermore, especially in Fig. 9 it can be seen that misclassifications happen mainly at the start of a classification period. This behavior is explained by the fact that the transition phases, were excluded for training the ANN to keep its structure simple.

We conducted further experiments with non-trained variations of the setup. Figure 11 shows the performance of the classifier with an impact in the ${}^{\mathcal{O}}y$-axis, instead of the ${}^{\mathcal{O}}z$-axis used in training. We can observe that the performance remains good. This shows that the magnitude of the 3D acceleration vector as input to the algorithm improves the applicability, as it makes the method direction independent, without requiring further training data. Also tests with velocities outside the training data range were conducted, see Fig. 12. However, for lower velocities the vibrations vanish,

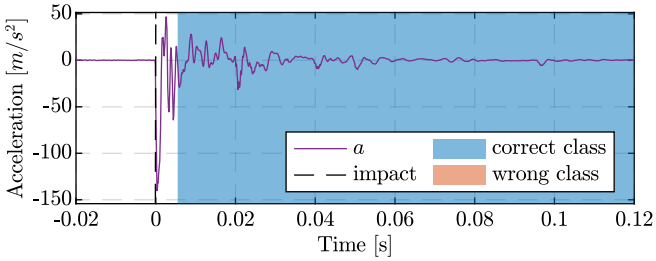

Fig. 10. The higher the frequency a material resonates at the less samples are required to detect its collision. The whole collision process is also a lot faster (see timescale). This graph shows a rather rigid metal impact at $0.2\,\mathrm{m/s}$ and $75°$.

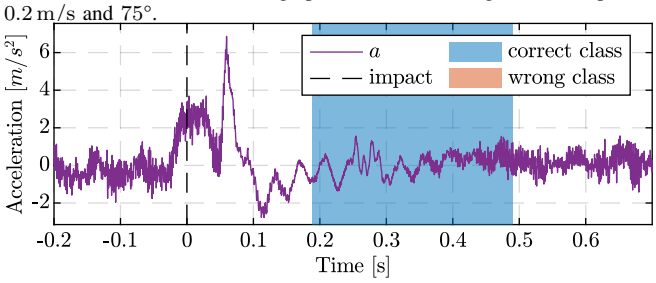

Fig. 11. Direction extrapolation performance. This collision with foam happened while the robot was moving in y-direction, whereas also the collision location was shifted $80\,\mathrm{mm}$ in y-direction of the base. The velocity was set to $0.2\,\mathrm{m/s}$.

which is expected from contact mechanics, but still this constitutes a limitation for our algorithm to velocities above at least $0.1\,\mathrm{m/s}$. Nevertheless, at higher velocity collision scenarios the classifier is able to successfully detect the collisions. Only at almost twice the maximum training collision speed the performance starts to decrease.

An evaluation of the calculation time over 64260 cycles for the FFT and the ANN in C++, showed a mean cycle time of $57.4 \pm 7.4\,\mu\mathrm{s}$, with a maximum of $439.9\,\mu\mathrm{s}$. As all these values are below the cycle time for the robot controller ($1\,\mathrm{ms}$) it proves that our algorithm may run in a real-time loop, without necessitating extensive hardware. The experiments denote that the method is robust enough to work with real sensory measures and can correctly detect trained classes in real-time. Furthermore, the approach shows large potential due to its simplicity, low-cost, speed, and application flexibility which motivates further investigations on the topic.

VI. CONCLUSIONS

We presented a real-time collision detection and classification method that is capable of generating a training data-set in an unsupervised manner from generally labelled experimental collision data. The data is then used to train a small neural network that can predict the set of classes predefined in the experiments. The approach is robust and material agnostic and has the potential extension to many different use cases. Our experiments showed that the method successfully works in real-world real-time experiments and is also capable of extrapolation outside the training set of experiments. Future work intends to investigate on techniques to improve classification robustness, specially for extrapolation scenarios and early collision detection refinements.

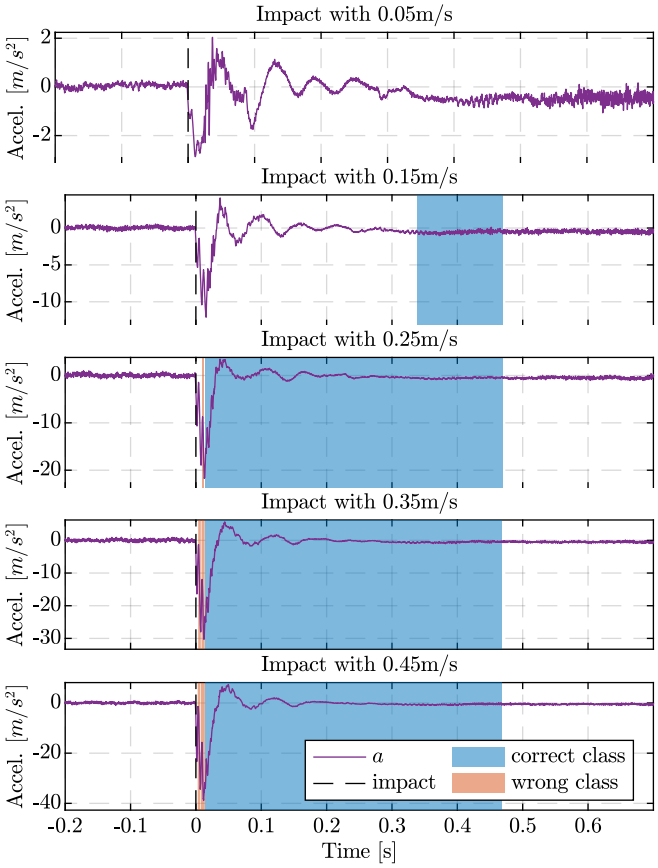

Fig. 12. Velocity extrapolation performance. We recorded impacts with the PRMS configured as hand and impacted with $0.05\,\mathrm{m/s}$ to $0.45\,\mathrm{m/s}$ with $90°$. For lower velocities the performance decreases drastically, whereas for $0.05\,\mathrm{m/s}$ no classification is possible, as the collision is not even recognized. For higher velocities the classification shows good results.

REFERENCES

[1] S. Golz, C. Osendorfer, and S. Haddadin, "Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3788–3794.

[2] A. Kouris, F. Dimeas, and N. Aspragathos, "Contact distinction in human-robot cooperation with admittance control," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 1951–1956.

[3] ——, "A Frequency Domain Approach for Contact Type Distinction in Human–Robot Collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 720–727, 2018.

[4] D. Popov, A. Klimchik, and N. Mavridis, "Collision detection, localization & classification for industrial robots with joint torque sensors," in *Human-robot collaboration and human assistance for an improved quality of life*, 2017, pp. 838–843.

[5] A.-N. Sharkawy, P. Koustoumpardis, and N. Aspragathos, "Human-robot collisions detection for safe human-robot interaction using one multi-input-output neural network," *Soft Computing*, vol. 24, p. 66876719, 05 2020.

[6] M. Lippi, G. Gillini, A. Marino, and F. Arrichiello, "A data-driven approach for contact detection, classification and reaction in physical human-robot collaboration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3597–3603.

[7] Z. Zhang, K. Qian, B. W. Schuller, and D. Wollherr, "An online robot collision detection and identification scheme by supervised learning and bayesian decision theory," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1144–1156, 2021.

[8] S. Haddadin, A. de Luca, and A. Albu-Schäffer, "Robot Collisions: A Survey on Detection, Isolation, and Identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[9] M. H. M. A. Tan, F. Mat, I. M. A. Rahim, N. L. T. Lile, and S. Yaacob, "Classification of materials by modal analysis and neural network," in *ICIMU: Proceedings of the 5th international Conference on Information Technology Multimedia*, 2011, pp. 1–5.

[10] P. Wisanuvej, J. Liu, C.-M. Chen, and G.-Z. Yang, "Blind collision detection and obstacle characterisation using a compliant robotic arm," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2249–2254.

[11] M. Frigo and S. G. Johnson, "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, special issue on "Program Generation, Optimization, and Platform Adaptation".

[12] M. Neumann, K. Nottensteiner, I. Kossyk, and Z.-C. Marton, "Material classification through knocking and grasping by learning of structure-borne sound under changing acoustic conditions," in *IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 1269–1275.

[13] X. Fan, D. Lee, Y. Chen, C. Prepscius, V. Isler, L. Jackel, H. S. Seung, and D. Lee, "Acoustic collision detection and localization for robot manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9529–9536.

[14] Y. Ding, H. Kisner, T. Kong, and U. Thomas, "Using machine learning for material detection with capacitive proximity sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 424–10 429.

[15] M. Kerzel, M. Ali, H. G. Ng, and S. Wermter, "Haptic material classification with a multi-channel neural network," in *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 439–446.

[16] S. S. Baishya and B. Buml, "Robust material classification with a tactile skin using deep learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 8–15.

[17] S. A. B. Birjandi and S. Haddadin, "Model-Adaptive High-Speed Collision Detection for Serial-Chain Robot Manipulators," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6544–6551, 2020.

[18] S. A. B. Birjandi, J. Kühn, and S. Haddadin, "Observer-Extended Direct Method for Collision Monitoring in Robot Manipulators Using Proprioception and IMU Sensing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 954–961, 2020.

[19] F. Min, G. Wang, and N. Liu, "Collision Detection and Identification on Robot Manipulators Based on Vibration Analysis," *Sensors (Basel, Switzerland)*, vol. 19, no. 5, 2019.

[20] S. A. B. Birjandi, J. Kühn, and S. Haddadin, "Joint Velocity and Acceleration Estimation in Serial Chain Rigid Body and Flexible Joint Manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7503–7509.

[21] W. McMahan, J. M. Romano, and K. J. Kuchenbecker, "Using Accelerometers to Localize Tactile Contact Events on a Robot Arm," 2012.

[22] L. Kaul, S. Ottenhaus, P. Weiner, and T. Asfour, "The sense of surface orientation — A new sensor modality for humanoid robots," in *Humanoids*, 2016, pp. 820–825.

[23] DIN ISO/TS 15066:2016-02, Robots and robotic devices - Collaborative robots (ISO/TS 15066:2016).

[24] ISO 9283:1998-04, Manipulating industrial robots - Performance criteria and related test methods (ISO 9283:1998).

[25] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019.

[30] G. Guennebaud, B. Jacob, and others, "Eigen v3," http://eigen.tuxfamily.org, 2010.