



HAL
open science

The Impact-Aware Robotics Database: Supporting Research Targeting the Exploitation of Dynamic Contact Transitions

Maarten Johannes Jongeneel, Sander Dingemans, Alessandro Saccon

► **To cite this version:**

Maarten Johannes Jongeneel, Sander Dingemans, Alessandro Saccon. The Impact-Aware Robotics Database: Supporting Research Targeting the Exploitation of Dynamic Contact Transitions. 2023. hal-03900923v2

HAL Id: hal-03900923

<https://hal.science/hal-03900923v2>

Preprint submitted on 23 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Impact-Aware Robotics Database: Supporting Research Targeting the Exploitation of Dynamic Contact Transitions

M. J. Jongeneel, S. Dingemans, and A. Saccon

Abstract—We present a new database supporting the development of impact-aware robotics, an emerging field of research focused on enabling robots to exploit physical impacts with objects and environments to allow for dynamic manipulation and locomotion. The database can store a wide variety of datasets containing recorded impact experiments where robots, objects, and environments experience intentional collisions while performing dynamic robotic tasks. Examples are object tossing with robotic arms to speed up throughput or grabbing swiftly a heavy object. This open database based on FAIR principles provides access to data that supports research related to, e.g., modeling, control, parameter identification, and object tracking. We propose specific procedures for capturing, labeling, and storing the data based on a new impact-aware robotics taxonomy to facilitate the cataloging and search of specific datasets. We provide open software tools to structurally collect and store the data and allow other research groups to contribute their datasets to the database. A custom web interface allows end-users to efficiently browse the database and download datasets.

Index Terms—Robotics, Collision, Impact-Aware Robotics, Database, Data Collection, Data Storage, Experimental data

I. INTRODUCTION

Impact-aware robotics is an emerging field of research focused on developing robotic applications exploiting physical impacts with objects and environments for dynamic manipulation and locomotion. Exploiting impacts (contact at nonzero speed) in robotics allows for faster execution of existing robotic tasks, and in some cases, it enables new robotic tasks to be performed. For instance, in robot manipulation, dynamic grabbing of objects can speed up depalletizing processes, and tossing objects via robotic arms enables faster object handling and allows robots to place objects outside their kinematic reach. These illustrative examples are also shown in Figure 1, as part of the European project on Impact-Aware Manipulation (I.A.M.)¹. To foster research on impact-aware robotics, it is paramount to have experimental data to develop, test, and compare methods related to modeling, planning, learning, sensing, and control.

In the field of whole-body motion, the KIT Whole-Body Human Motion Database [1] provides an excellent example of a database that structurally stores experimental data,

M. J. Jongeneel (Corresponding author), S. Dingemans, and A. Saccon are with the Department of Mechanical Engineering, Eindhoven University of Technology (TU/e), The Netherlands ({m.j.jongeneel, s.dingemans, a.saccon}@tue.nl)

This work was partially supported by the EFRD OP Zuid project ‘Holland Robotics Logistiek’, ref. PROJ-02535.

¹This work was partially supported by the Research Project I.A.M. through the European Union H2020 program under GA 871899.

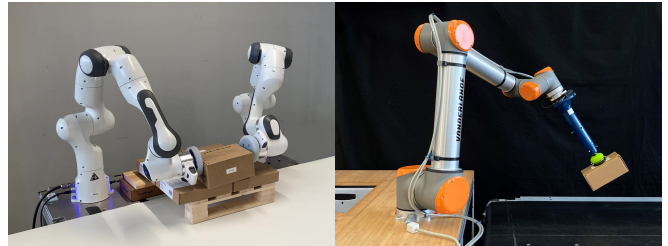


Fig. 1: Two applications of impact-aware robotics. Dynamic grabbing (left) and tossing (right) of objects.

ensuring this data is Findable, Accessible, Interoperable, and Reusable (FAIR)². The KIT database lets users access all datasets through a web interface, where they can search for specific datasets using metadata tags. Each dataset comes with a preview video, an experiment description, and detailed information about the objects used in the recording. Additionally, each file in a dataset has a description that explains the data it contains. To the best of the authors’ knowledge, there is currently no open database specifically addressing the recording and sharing of impact events in robotics, in the spirit of the excellent example of the KIT whole-body human motion database. While various datasets are available that relate to diverse robotic tasks, such as robot pushing [2], [3], [4] robot manipulation [5], [6], robot grasping [7], or collision detection [8], [9], these works do not specifically focus on the exploitation of impacts. Although some datasets related to impact-aware robotics do exist (see, for instance, [10], [11], [12]), our general observation is that most datasets are tailored to support a specific publication, with data often stored without considering its potential reuse for different purposes and by different users. In some cases, datasets also lack the support of metadata for human or machine interpretability, making it difficult to understand and use the data. Furthermore, we observe that some datasets have become unavailable unless an institution was chosen for data storage whose mission is to preserve the data for several years in the future.

Drawing from these lessons and our own experience and research in the field, including the need to safely store and reuse the collected data, we propose the Impact-Aware Robotics Database. We aim to provide a data repository of recorded experiments to make collected datasets available to other research institutions and companies worldwide, with the long-term goal of stimulating research and advancement in impact-aware robotics. Furthermore, we encourage other research groups to contribute to the database by uploading

²See <https://www.go-fair.org/fair-principles/>.

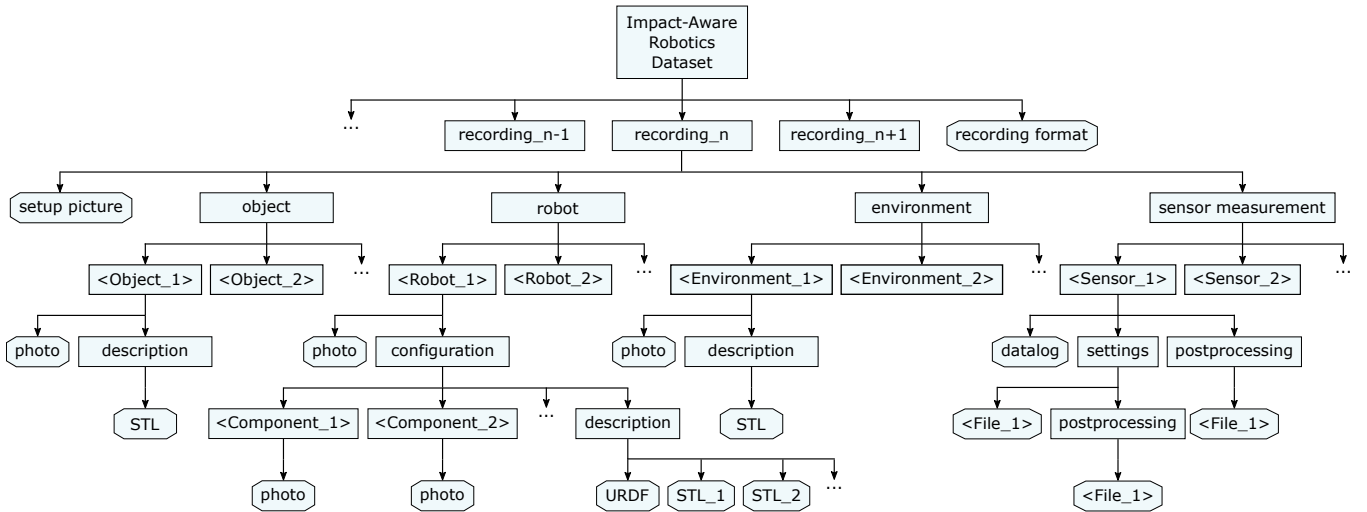


Fig. 2: Structure of a dataset. Each dataset contains a set of recordings and a description of the recording format. Each recording is subdivided into object, robot, environment, and sensor measurement subfolders and contains a picture of the setup used during for the experiments.

their datasets related to impact-aware robotics. The datasets are only required comply with a chosen impact-aware repository structure and taxonomy. This structure guarantees the reuse and searchability of the stored data, serving the purpose of encouraging the robotics community to develop, test, and compare modeling, learning, sensing, and control approaches for impact-aware robotics. Our primary contributions are:

- An open service providing systematic storage, search, and access of data for the robotics community to support the growth of impact-aware robotics-related research
- An archive structure where data is stored in institutional servers, ensuring data is Findable, Accessible, Interoperable, and Reusable (FAIR) for a long period (at least 15 years)
- Providing access to open data creation and processing tools to generate datasets complying with this structure
- A contact transition taxonomy providing specific metadata to ease and ensure the searchability of the datasets
- A web frontend enabling the domain-specific search of impact-aware datasets according to this taxonomy

A. Paper Structure

The remainder of this paper is organized as follows. In Section II, we describe the general structure of a dataset and the type of data that can be stored in the database. In Section III, we describe the data collection and storage procedures to obtain experimental data and to store it in a structured way. Section IV covers the contact transition taxonomy that provides the datasets with specific metadata tags. Section V describes the procedures for contributing to the database and accessing the data via the developed web interface. In Section VI, a dataset that is part of the database is further detailed to provide an example of the type of data that can be stored. Section VII closes with a conclusion and recommendations for future work.

II. THE IMPACT-AWARE ROBOTICS DATABASE

The Impact-Aware Robotics database consists of multiple datasets that are stored using the open HDF5 format [13].

The HDF5 format is chosen as it is a versatile data model that can represent complex, heterogeneous data objects and a wide variety of metadata through an unlimited variety of data types, it is self-describing, and it is an architecturally independent software library that runs on a wide range of computational platforms and programming languages. Each dataset is constituted by several recorded experiments, focusing on a specific type of release/impact motion and scenario, involving known **objects**, **robots**, and **environments**. It can contain, especially, time series data related to

- objects impacting the environment,
- objects impacting robots,
- objects held by robots impacting the environment,
- objects impacted by robots (being in contact with the environment),

and corresponding release motions related to

- objects detaching from the environment,
- objects separating from a robot,
- objects separating from the environment while being held by robots,
- objects separating from a robot while being in contact with the environment.

Each dataset is organized in a hierarchical structure (folder/file type structure), containing both sensor measurement data and metadata (attributes, both at folder and file level). The proposed structure for each dataset is straightforward, containing a root folder and several subfolders, each corresponding to a specific recording. Figure 2 shows this dataset structure and highlights the structure of the specific recording number n . Under each recording, the **object**, **robot**, and **environment** folder contain metadata that allows for both human and machine interpretability and search. Examples are the properties of the object (mass, inertia, dimensions), the exact robot configuration and description (e.g., URDF with STL files), and properties of the environment (e.g, surface material and speed of a conveyor belt). Furthermore, each recording contains generic metadata about the experiment, such as date, location, and author.

The **sensor measurement** folder contains all time series data collected using external and robot sensors. A representative list of sensors by which time series data can be collected is the following:

- robot joint displacement sensors (encoders/resolvers),
- robot joint torque sensors,
- robot joint current,
- force/torque sensors,
- standard cameras/depth cameras,
- motion capture systems,
- high-speed cameras/event cameras,
- accelerometers/IMU sensors,
- pressure sensors,
- robot input commands,
- any other sensors attached to objects or environments.

We opt to store a large variety of sensor data corresponding to various robotics scenarios. This allows the database to be used for various research purposes, but requires a proper storage structure of the time series data based on their primary recording purpose. For example, we consider time series data relevant for online learning, sensing, estimation, and feedback control, and time series data relevant for offline modeling and learning. We also store time series data that enhances human understanding of the recordings, for example, a camera recording of impact/release experiments synchronized with the time series data obtained by other sensors. Based on the primary recording purpose of a sensor, we store post-processed data together with the raw sensor data. A concrete example of raw data can be the output of a motion capture system in the form of a table, containing many columns corresponding to different properties of tracked objects (e.g., position data of a motion-capture marker corresponding to a predefined rigid body). The corresponding post-processed data can then contain the rigid-body transformation matrices, expressing the position and orientation of the object with respect to a reference frame. When we provide post-processed data, we include all software scripts and metadata to make the conversion from raw sensor-data to post-processed data.

Because the number of datasets within the Impact-Aware Robotics database will grow, it is essential to provide each dataset with metadata that are capable of identifying the purpose and type of impact/release motions, besides the metadata related to time, location, authors, type of object(s), robot(s), environment sensor(s), and associated time-series sensor data. These metadata have the purpose to identify similar datasets and, within those, identify specific recordings that share similar features (e.g., the impact of a box with a conveyor belt or the grabbing of a box via a dual-arm robotic system). For each dataset, metadata are stored both at root and recording level, allowing every single recording within the dataset to be searchable within the entire repository. Specific metadata tags have been created to describe the different types of impact/release motions, and we will detail these in Section IV.

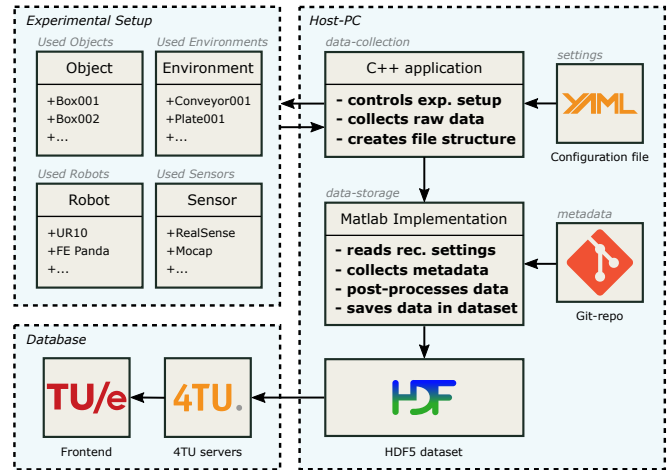


Fig. 3: Overview of the software architecture. The host-PC contains the data-collection application that takes the recording settings from a configuration YAML file (top-right corner) to control the experimental setup, collects the data from the recording, and stores the data in a session/recording file structure. The data-storage application takes this raw data and stores it with the associated metadata imported from an external Git repository, and to an HDF5 dataset. This dataset is then uploaded to the 4TU servers, and a web application crawls this information to display it on the frontend.

III. DATA COLLECTION AND STORAGE PROCEDURES

In this section, we will describe the software architecture we have developed for collecting and storing experimental data according to the structure discussed in Section II and shown in Figure 2. Within this software architecture, we distinguish between *data collection* (e.g., controlling the experimental setup and collecting raw sensor data) and *data storage* (e.g., post-processing raw sensor data to enhance human and machine interpretability, providing metadata, and storing the data in a structured way in an open format file). In the following subsections, we will detail our approach to data collection and storage. We share the software online³, which provides an example and possibility for other researchers that want to contribute to the Impact-Aware Robotics Database. Further details on our approach on data collection and data storage are given in [14, Section 2.3].

A. System overview

Figure 3 provides an overview of the developed software architecture. A host PC controls the experimental setup and logs the incoming data from the sensors via the *data collection* application, which also stores the data of each experiment in separate recording folders. Next, via the *data storage* application, the raw sensor data is processed and stored together with the associated metadata in datasets. These datasets are uploaded to 4TUResearchData [15] OpenDAP servers as part of the Impact-Aware Robotics Archive Collection [16]. 4TUResearchData is an international data repository, funded by the 4TU⁴, and offers dataset curation, sharing, long-term access, and preservation services. A web application that we have developed extracts information from

³The code used for collecting and storing the data is made publicly available via <https://gitlab.tue.nl/robotics-lab-public/impact-aware-robotics-database>.

⁴4TU is a collaboration between the four technical universities in the Netherlands. See also https://www.4tu.nl/en/about_4tu/.

the 4TU servers and displays this information on the front-end. Further details about the structure and functionalities of the web application are given in Section V.

B. Data Collection

The purpose of the data collection application is to record raw sensor data and store this data in a structured way on the host-PC as preparation for the data post-processing and storage. The application contains various functions that each control and record data of a specific sensor. The structure of this application is modular to ensure that new functions corresponding to new sensors can easily be added to the application without interfering with the existing structure. Next to these functions, the application contains a configuration YAML file (top right of Figure 3) containing the recording settings (e.g., it defines what sensors are used to record the experiments) and initializes and configures a recording of a specific experiment. Furthermore, this configuration YAML file contains metadata specifying what objects, robots, and environments are used in the recordings and contains information about the setup, the author(s), and a short description of the experiment. The configuration YAML file is stored with the raw sensor data in the folder for that specific experiment. This ensures that all information about one experiment is stored together in one dedicated folder.

C. Data Storage

The purpose of the data storage application is to post-process and store the recorded (raw) data files with the associated metadata in a single dataset according to the structure described in Section 2. All objects, robots, and environments have associated metadata (e.g., pictures, STL files, robot configuration files) stored on an external public Git repository (middle right of Figure 3). This metadata is copied and stored in the dataset with the raw and post-processed data to ensure that the datasets are stand-alone and the data can be interpreted by both humans and machines.

IV. CONTACT TRANSITION TAXONOMY

This section provides further details on how the time series of sensor recordings related to a contact transition (impact or release) can be uniquely classified using a limited number of metadata entries. This classification is essential to be able to retrieve specific datasets (e.g., just those containing recordings of impact data between an object and a surface or those containing recordings of specific impacts between a robot and an object, while in a specific contact state with the environment) out of the database.

As an illustration, the classification of impact events related to grabbing a box sliding on a table via a dual-arm robot can be represented as in Figure 4, which also illustrates the main structure of the taxonomy tree. In the following, the rationale and details of this classification (taxonomy) are provided, together with how such information will be stored as metadata in every dataset.

We propose a qualitative metric based on the type of contact transitions involving objects, robots, and environment

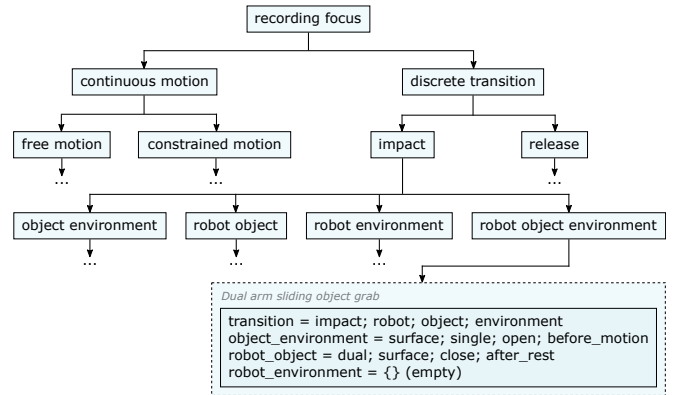


Fig. 4: Illustration of the taxonomy. The drawing illustrates a special leaf in the taxonomy tree associated to a dual-arm grabbing of a box, sliding on a surface before being picked up.

entities. Contact transitions are classified from simpler to more complex, based on four levels, depending if the transition involves (i) just the object and the environment; (ii) just the robot and the object; (iii) just the robot and the environment, and (iv) the robot, object, and the environment. Furthermore, the dynamic contact transitions are classified either as impacts (e.g., a box hitting a table) or release (e.g., a box falling off a table). We consider impacts to be at a higher level of complexity than releases, since impacts cause jumps in the velocity of the object, while releases only cause jumps in the acceleration of the object.

Correspondingly, each recording of dynamic contact transition in a dataset can be classified using the metadata entry *transition*, which is a list of the following entries, comprising both the type of transition and the entities involved:

- **impact**
- **release**
- **object**
- **robot**
- **environment**

Only eight accepted combinations for the transition entries above are possible:

- **release; object; environment**
- **impact; object; environment**
- **release; robot; object**
- **impact; robot; object**
- **release; robot; environment**
- **impact; robot; environment**
- **release; robot; object; environment**
- **impact; robot; object; environment**

It is essential to understand that impact and release refers to the main purpose for the action: a box grabbed from a table will show both an impact between the gripper and the box and a release between the table and the box. Nonetheless, this is classified simply as impact. Although the database targets to mainly store impact and release motions, the database also allows to store datasets with no contact transitions (e.g., for storing data to be used to identify the flexible model of a suction cup). To this end, two additional entries for the metadata transition are provided. These two additional entries are **free** and **constrained**, and additional eight combinations

are possible (not listed here for sake of brevity). Besides the metadata *transition*, three other metadata entries are defined to further detail the content of the dataset. These metadata will allow for a finer search within the repository (e.g., to get from all recordings of impact between an object and the environment, only those showing bouncing on a single point). These three additional metadata are

- *object_environment*
- *robot_object*
- *robot_environment*

For all of these metadata, we introduce the entries **point**, **edge**, and **surface** to specify the contact geometry. Furthermore, we introduce the entries **single**, **dual**, and **multi** to specify the number of separate contact locations. As an example, for an impacting grasp of a two-finger end-effector with a box, the *robot_object* metadata is specified as **robot_object = dual; point**.

To allow for a more detailed level of classification that ease the search of specific archives in the database, for each object-environment, robot-object, and robot-environment contact pair, one can use the following position-level entries, where, between brackets, we describe what happens before, during, or after the impact event:

- **bounce**: [open before & after, contact during]
- **open**: [contact before, open after]
- **partly_open**: [contact before, less contact area after]
- **stay_closed**: [contact before and after, same area]
- **further_closed**: [contact before, more contact area after]
- **closed**: [open before, contact after]
- **shift**: [contact before & after, different area]
- **partly_shift**: [contact before & after, overlap area]

These eight position-level entries are the enumeration of all possible pre- and post-event combinations of the contact interface state between two entities. Finally, the following pre- and post-event velocity-level entries can be used to describe if relative motion is present at the contact interface before and after contact is established:

- **before_motion**: [relative motion, before event]
- **before_rest**: [no relative motion, before event]
- **after_motion**: [relative motion, after event]
- **after_rest**: [no relative motion, after event]

The entries **before_motion** and **before_rest** cannot be used when the entries **closed** and **bounce** are used, as there was no contact before the event. Similarly, the entries **after_motion** and **after_rest** cannot be used for the entries **open** and **bounce**, as there is no contact after the event. Although further detailing the contact motion state is possible, the current level of description is deemed sufficient for current and planned datasets stored in the database.

As a first illustrative example, an impact corresponding to a dual-arm robot grabbing a box that was sliding, isolated, on a surface can be described with the following metadata, also displayed in Figure 4:

transition = impact; robot; object; environment
object_environment = surface; single; open; before_motion

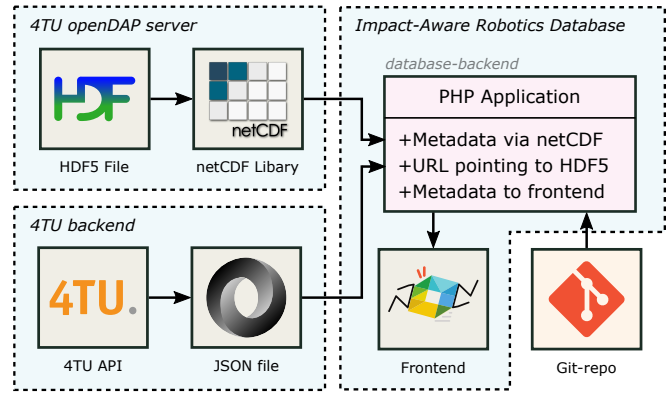


Fig. 5: Overview of the database web application. A PHP application crawls information from the 4TU backend about the datasets stored under the Impact-Aware Robotics archive collection and retrieves metadata from these datasets stored on the 4TU openDAP servers via a netCDF library. This information is then displayed on the database frontend, together with metadata on the objects, robots, and environments retrieved from an external Git-repository.

robot_object = surface; dual; closed; after_rest
robot_environment = (empty)

Further details about the taxonomy can be found in [17, Section 3] and on the Impact-Aware Robotics Database website [18].

V. ACCESSING THE DATABASE

All datasets are published in the Impact-Aware Robotics Archives collection [16] and can be viewed and downloaded via the web interface of 4TUResearchData. However, the 4TU web interface does not provide the option to search for specific metadata tags (related to objects, robots, environments, sensors, and the contact transition taxonomy) that are stored within the dataset. This makes it difficult for any user to understand the content of a dataset without having to download the dataset first.

To this end, we developed a web interface [18] that provides a platform to search for and download datasets from the database. On this web interface, users can search for datasets via specific keywords such as keywords related to the objects, robots, environments, and sensors used in the dataset, as well as the contact transition taxonomy and recording purpose. The web interface contains all metadata of all objects, robots, and environments used in the datasets that are published in the database. Dedicated pages are created for objects, robots, and environments. On each of these pages, users can search for specific items (e.g., Box005 for object, or Conveyor002 for environment), view the detailed item information, and view all datasets where this item was used after which they can be downloaded.

Figure 5 shows an overview of the database web application. A PHP application queries information about the datasets stored under the Impact-Aware Robotics archive collection via the 4TU web API, that returns this information in a JSON file. This JSON file contains the URL pointing to the HDF5 files stored on the OpenDap [19] servers of 4TU. A request for metadata or content of these HDF5 files can then be made by issuing an HTTP request to the OpenDAP

server via the netCDF [20] library. It then displays this information, together with all metadata of the objects, robots, and environments retrieved from an external Git-repository (lower right corner of Figure 5), on the web interface.

A. Contributing to the database

The database contains, at the moment of writing, 24 datasets with a total storage of 37GB recorded on various setups and for different purposes. Figure 1 shows two different examples of setups used in the recordings. Most of the datasets currently stored in the database are collected during the research project I.A.M. However, we intend the database to grow beyond this project and to provide a platform for data storage for any experiment related to impact-aware robotics.

To this end, we encourage researchers in the field of impact-aware robotics to contribute to the database by providing the option to upload datasets to the database. In Section III, we discussed the open software we have created that can be used to collect and store experimental data in a predefined structure used for the Impact-Aware Robotics Database. Datasets complying with this structure can be uploaded to the database. At this stage, the procedure to do so is as follows, but the reader is referred to the database website [18] for the latest information.

To upload a dataset, the end-user must create an account on 4TUResearchData [15]. This can be done either through an institutional login (if the institution the end-user is connected to is a partner of 4TU) or via an eduID⁵, which can be created using an email address and name. End-users that are not connected to an institution that is partnered with 4TU can get up to 5GB of free storage. End-users that are connected to a partner institution can upload more data, depending on the partner status (either gold, silver, or bronze member). Datasets can then be uploaded under different licenses⁶ and linked to the Impact-Aware Robotics Archive Collection [16] on 4TUResearchData by adding a metadata tag pointing to this collection. The best practice is to contact a TU/e contact mentioned on the website [18] for advice or support on data conversion and upload. After the dataset is uploaded, a data curator from the TU/e evaluates the dataset before it is added to the collection. Once the dataset is part of the collection on 4TU, it is automatically displayed on the website of the Impact-Aware Robotics Database.

VI. ILLUSTRATIVE EXAMPLE DATASET

In this section, we will further detail the content of a dataset that is part of the database to provide an illustrative example of the content that can be stored. We consider a dataset that contains 150 recordings of a UR10 manipulator tossing a carton box on a conveyor [21]. We encourage the

⁵An eduID is an account that allows to login into several services connected to Dutch educational and research institutions, meant for users that are not connected to any of those institutions. More info can be found at <https://eduid.nl/en/>.

⁶All datasets currently stored in the database are publicly available under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC-BY-NC-SA 4.0) license. We recommend using this license, but authors are free to choose differently.

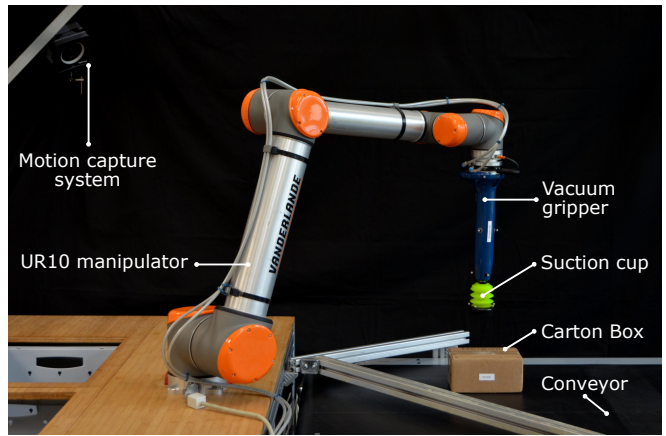


Fig. 6: Picture of the experimental setup used for data collection of the example dataset.

TABLE I: Technical details about the used equipment

Item	Specification	Value
Camera Sensor	Type	RealSenseD415
	Resolution	640 × 480
	Frame rate	30 FPS
	Output	bag file converted to mp4
Pressure Sensor	Type	Schmalz VS VP8 SA M8-4
	Range	[−1, . . . , 8] bar
	Input/output	Analog 0-10V
	Repeatability	±3% of range
Motion Capture	Output	CSV file
	Type	OptiTrack
	Cameras	4 × PrimeX17, 2 × PrimeX22
	Frame rate	360 FPS
UR10 sensors	Accuracy	±0.15 mm
	Additional files	Obj. marker pos., assets
	Output	CSV file, assets file
	Type	Encoder data
Host PC	Accuracy	0.1 degree
	Resolution	20 bit
	Output	CSV file
	Type	HP EliteDesk 800 G5 TWR
Object	Processor	Intel Core i7 9th Gen
	Storage	2TB SSD
	Type	Box006
	Mass	0.365 kg
Robot	Dimensions	[0.2052, 0.1554, 0.1006] m
	Inner Material	Sculpture Block SB30
	Outer Material	Carton
	Additional files	Inertia, Picture, STL
Environment	Type	UR10
	Tool Changer	FIPA SR90-A-SL
	Tool Arm	Smart Robotics GS02-v05
	End Effector	Piab piGRIP
Environment	Additional files	Picture, STL, URDF
	Class	Conveyor
	Type	Custom Vanderlande
	Additional files	Picture, STL

reader to download and inspect the content of the dataset for better understanding of its content and structure⁷. The purpose of these experiments is to obtain the friction and restitution parameters related to the box-conveyor impacts, which are used in a modeling framework to predict the trajectory of the box after it is tossed by the robot arm. Figure 6 shows an overview of the experimental setup with the different components indicated.

⁷To view and inspect a HDF5 file, one can use the open HDF View software [22].

In each experiment, the UR10 robot arm picks up the box from a known location on the conveyor belt. The robot then moves forwards and tosses the box such that it impacts the conveyor with a single corner. The conveyor belt then transports the box back to its initial position in the corner of the metal frame shown in Figure 6. As the main focus of these experiments is related to the box-conveyor impacts, we can classify these impacts under the *object environment* class as part of the discrete transition/impact branch of the taxonomy tree of Figure 4. Hence, the **transition** metadata tag for this dataset is

- **transition = impact; object; environment.**

The three additional metadata tags are

- **object_environment:** point; single; bounce
- **robot_environment:** (empty)
- **robot_object:** (empty)

since the impact is mostly happening at a single corner of the box when landing on the conveyor and the contact is open before and after the impact.

The box, the suction cup, and the vacuum gripper are equipped with motion capture markers, which allow us to track the pose (position and orientation) of these parts with sub-millimeter accuracy at a rate of 360 frames per second. A RealSenseD415 RGB camera is used to record each of the experiments and is oriented towards the experimental setup to obtain a view as shown in Figure 6. Furthermore, inside the vacuum gripper, a pressure sensor is mounted that allows us to record the pressure inside the gripper affected by the vacuum generated during the pick-up and holding of the box. From the UR10 manipulator, we retrieve the joint-encoder sensor data and the trigger signals sent to release the object. We use the host PC to start and stop the recording of each sensor, and these signals are also recorded. Table I shows the technical details of the equipment used in these experiments, which are all stored in the dataset created from this set of experiments. Furthermore, Table I shows the additional files that are stored in the dataset, such as the URDF of the robot or pictures of each of the components. All this data is stored in a single HDF5 dataset, using the structure shown in Figure 2, and the dataset can be viewed and downloaded via [21].

VII. CONCLUSION

We have presented a new database dedicated to the development of impact-aware robotics. At the time of writing, the database contains 24 different datasets related to manipulation, grasping, pushing, and tossing in robotics. We provide open software to collect and store experimental data in a structured way based on an impact-aware contact transition taxonomy and FAIR principles, allowing researchers to contribute to the database with their datasets. Furthermore, we provide a dedicated web interface to access, view, and download the datasets. As future work, we plan to expand the database with new datasets, and we will continue the development of the web interface and the software used for data collection and storage. We are open to contributions of datasets and future improvements and developments.

REFERENCES

- [1] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "The kit whole-body human motion database," in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 329–336.
- [2] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised Learning for Physical Interaction through Video Prediction," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 64–72.
- [3] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters, "Learning inverse dynamics models in O(n) time with LSTM networks," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 811–816.
- [4] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez, "More than a million ways to be pushed. A high-fidelity experimental dataset of planar pushing," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 30–37.
- [5] A. S. Polydoros and L. Nalpantidis, "A reservoir computing approach for learning forward dynamics of industrial manipulators," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 612–618.
- [6] Y. Huang and Y. Sun, "A dataset of daily interactive manipulation," *The International Journal of Robotics Research*, vol. 38, no. 8, pp. 879–886, 2019. [Online]. Available: <https://doi.org/10.1177/0278364919849091>
- [7] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018. [Online]. Available: <https://doi.org/10.1177/0278364917710318>
- [8] K. Katsampiris Salgado and K. Haninger, "High payload collaborative robot - joint states/motor current/tcp force-torque," Oct 2022. [Online]. Available: [10.5281/zenodo.7240050](https://zenodo.org/record/7240050)
- [9] Z. Zhang, "The robot joint torque measurements for accidental collisions and intentional contacts," Apr 2022. [Online]. Available: [10.5281/zenodo.6461868](https://zenodo.org/record/6461868)
- [10] S. Pfrommer, M. Halm, and M. Posa, "ContactNets: Learning Discontinuous Contact Dynamics with Smooth, Implicit Representations," in *4th Conference on Robot Learning (CoRL)*, ser. Proceedings of Machine Learning Research, vol. 155, Cambridge, MA, USA, Nov. 2020, pp. 2279–2291.
- [11] B. Acosta, W. Yang, and M. Posa, "Validating Robotics Simulators on Real-World Impacts," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6471–6478, 2022.
- [12] N. Fazeli, A. Ajay, and A. Rodriguez, "Long-Horizon Prediction and Uncertainty Propagation with Residual Point Contact Learners," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 7898–7904.
- [13] The HDF Group, "Hierarchical data format version 5," 2000-2022. [Online]. Available: <https://www.hdfgroup.org/solutions/hdf5/>
- [14] M. J. Jongeneel, A. Saccon, and J. den Ouden, "Deliverable D1.4: Publication of I.A.M. Dataset," Eindhoven University of Technology, Eindhoven, The Netherlands, Tech. Rep. D1.4, 2022.
- [15] The 4TU.Federation, "4TU.ResearchData data and software repository," 2008-2022. [Online]. Available: <https://data.4tu.nl/info/en/>
- [16] A. Saccon and M. J. Jongeneel, "Impact-Aware Robotics Archives Collection," *4TU.ResearchData*, Apr. 2021, Dataset. [Online]. Available: <https://doi.org/10.4121/c.5405187>
- [17] M. J. Jongeneel, J. D. Ouden, and A. Saccon, "D1.1, Publication of I.A.M. Dataset," 2020.
- [18] A. Saccon and M. J. Jongeneel, "TU/e Impact-Aware Robotics Database," Eindhoven, The Netherlands, 2022. [Online]. Available: <https://impact-aware-robotics-database.tue.nl/>
- [19] J. Gallagher, N. Potter, T. Sgouros, S. Hankin, and G. Flierl, "The data access protocol - dap," 2005-2022.
- [20] R. Rew and G. Davis, "Netcdf: an interface for scientific data access," *IEEE Computer Graphics and Applications*, vol. 10, no. 4, pp. 76–82, 1990.
- [21] M. J. Jongeneel and S. Dingemans, "Impact Aware Manipulation (I.A.M.) archive containing box-drop experiments for Parameter Identification of Box006," *4TU.ResearchData*, Sep. 2022. [Online]. Available: <https://doi.org/10.4121/21024007>
- [22] The HFD Group, "HDF View," 2022. [Online]. Available: <https://www.hdfgroup.org/downloads/hdfview/>