



**HAL**  
open science

## The Impact-Aware Robotics Database

Maarten Johannes Jongeneel, Sander Dingemans, Alessandro Saccon

► **To cite this version:**

Maarten Johannes Jongeneel, Sander Dingemans, Alessandro Saccon. The Impact-Aware Robotics Database. 2022. hal-03900923v1

**HAL Id: hal-03900923**

**<https://hal.science/hal-03900923v1>**

Preprint submitted on 15 Dec 2022 (v1), last revised 23 Jun 2023 (v2)

**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.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

# The Impact-Aware Robotics Database

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. This database allows getting access to data that can be used to answer various research questions related to, e.g., motion planning, parameter identification, and object tracking. The database can store a wide variety of datasets obtained from recordings of impact experiments. In these experiments, robots, objects, and environments are used in the context of physical impacts to perform various robotic tasks, such as object tossing with robotic arms to speed up throughput or drones impacting walls for inspection. To facilitate the search of specific datasets, we propose specific procedures for capturing, labeling, and storing the data based on an impact-aware robotics taxonomy. We provide software tools to structurally collect and store the data and allow other research groups to contribute their datasets to the database. We developed a web interface that allows end-users to efficiently search 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 in robotics allows for faster execution of existing robotic tasks, and in some cases, it enables new robotic tasks to be performed. For example, in robot manipulation, dynamic grabbing of objects can speed up depalmitizing processes and tossing objects via robotic arms can result in faster object handling and allows robots to place objects outside their kinematic reach. These two illustrative examples are also shown in Figure 1. In this context, it is paramount to have experimental data to develop, test, and compare methods related to modeling, planning, learning, sensing, and control for impact-aware robotics systems. However, we are unaware of any open database specifically addressing the recording and sharing of impact and release events in robotics. Although various datasets related to impact-aware robotics exist, our general observation is that most of these datasets support a specific publication, and data is often stored without considering the reuse of the data for different purposes. In many cases, datasets lack the support of metadata for human or machine interpretability, making it difficult to understand and use the

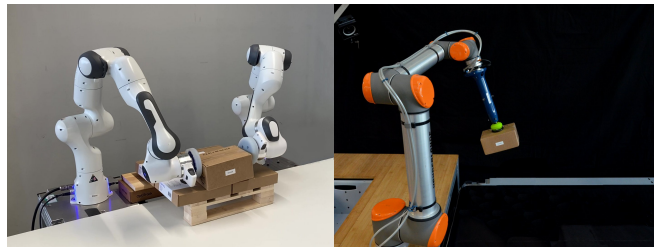


Fig. 1: Examples of two different setups used to collect data that is stored in the Impact-Aware Robotics Database.

data. Furthermore, on ample occasion we observe that dataset have become unavailable, unless an institution is chosen whose mission is to preserve the data.

Therefore, 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 stimulate other researchers to contribute to the database by allowing them to upload their datasets related to impact-aware robotics to the database. The datasets will be requested to comply with a repository structure and taxonomy. This structure will guarantee the reuse and searchability of the stored data, ensuring particularly that the stored data will have value for the whole robotics community to develop, test, and compare methods related to modeling, learning, sensing, and control of impact-aware robotics systems. Our primary contributions are:

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

### A. Related Work

To the best of author’s knowledge, there does not exist a database on impact-aware robotics or robots exploiting impacts in general. However, various databases and datasets exists of related topics, such as grasping, pushing, manipulation, and picking and placing and in Table I, we provide

All authors are with the Department of Mechanical Engineering, Eindhoven University of Technology (TU/e), The Netherlands (`{m.j.jongeneel, s.dingemans, a.saccon}@tue.nl`)

\*Corresponding author

TABLE I: Overview of related datasets and databases, including our database.

| Type              | Purpose                                  | Content  | # items                                | Status              |
|-------------------|--|--|--|---------------------|
| Dataset [1]       | Collision detection                      | Robot joint positions, motor currents, and force torque sensor values.   | 80 recordings                          | Available at [1]    |
| Dataset [2]       | Collision detection                      | Joint torque measurements.   | 21 Recordings                          | Available at [2]    |
| Database [3]      | Grasping                                 | Pre-grasp and final grasp poses, providing necessary data to recreate each grasp, like joint angles, hand position, and contact points between hand and object.  | 235737 grasps                          | No longer available |
| Dataset [4]       | Grasping                                 | Grasps (joint positions and velocities) and RGB images.  | 650000 grasps                          | No longer available |
| Dataset [5]       | Pick & Place                             | Robot joint positions, velocities & torques, target joint positions & velocities.  | 20 trajectories                        | Available at [6]    |
| Dataset [7]       | Pushing                                  | 59000 robot pushing motions, consisting of 1.5 million frames and corresponding actions. RGB images and gripper poses were collected.  | 59000 motions                          | No longer available |
| Dataset [8]       | Pushing                                  | KUKA arm pushes flasks filled with 200, 300 and 400 milliliters of water. The recorded datasets consists of joint angles, velocities, accelerations, and torques in a teleoperation experiment using a mirroring of commands approach in a bimanual setting. | 100000 samples                         | No longer available |
| Dataset [9]       | Pushing                                  | Each data file corresponds to one straight push and contains the interaction forces and the positions of the object and the pusher. The dataset contains recorded robot tip poses, object poses, and the wrench exerted on the pusher.                       | 6145 recordings                        | Available at [10]   |
| Dataset [11]      | Manipulation                             | Contains daily performed fine motion in which an object is manipulated to interact with another object. The dataset contains recorded position and orientation, force/torque sensor data, and vision data.   | 3354 robot motions                     | Available at [12]   |
| Our Database [13] | Manipulation, grasping, pushing, tossing | Datasets contain recordings of robotic tossing, dual-arm grasping, impacting and pushing, and manipulation. Datasets contain a wide variety of sensor data, including joint positions, velocities, and torques, RGB videos, and motion capture data.         | 16 datasets, 2174 recordings (growing) | Available at [13]   |

an overview of some of these. A survey on recent datasets on object manipulation can be found in [14].

A dataset on collision detection of a robotic arm can be found in [1]. The dataset contains data that was used for the design and development of a collision detection scheme, and examples of the stored data are the robot joint positions, the motor currents, and the force torque sensor values. A related dataset can be found in [2], containing joint torque measurements of a robot manipulator making intentional impacts, used in a research study on robot collision detection, classification, diagnosis, and prediction. Although both datasets contain robot collision data, related to impact-aware robotics, the primary purpose of the datasets is developing techniques to avoid impacts, instead of exploiting them.

The Columbia grasp database [3] can be considered as an example database, related to our work. It contains 238737 distinct form closure grasps, where each recorded grasp consists of a pre-grasp and the final grasp pose. For each experiment, the authors provide the necessary data to recreate each grasp, in the form of joint angles, hand positions, and contact points between the hand and the object. However, although the web page [15] related to the database is online, unfortunately it appears the data is not available for download, and the authors were not available for contact.

Similar to the Columbia database, in [4], the authors present a dataset containing approximately 650000 examples of robot grasping attempts. The dataset was mainly created to support the accompanying paper and focuses on training a convolutional neural network to servo a gripper in real time to achieve successful grasps. The authors provide a link to download the dataset, which directs to a page without an

option for download, making it impossible to use the dataset.

In [16] and [5], the authors present a dataset gathered from different robotic manipulators that execute pick-and-place tasks. The datasets contain trajectories generated during the execution of these tasks. However, the authors only provide MATLAB files without metadata supporting the dataset, which makes it hard to interpret the data.

A robotic pushing dataset is presented in [7] in the context of unsupervised learning from video for robotic control tasks. It contains approximately 59000 recorded motions of 10 robotic arms pushing hundreds of objects in bins. In addition to the RGB images, the authors collected the corresponding gripper poses and actions commanded to the gripper pose. Unfortunately, the authors provide a link in the paper that directs to a webpage stating that the files are no longer available.

Another dataset related to robotic pushing is presented in [8], where the authors used demonstrations of manipulation tasks for training a robot to push flasks filled with a level of 200, 300, and 400 milliliters. The datasets were recorded with a KUKA arm that pushes the flasks, with training set sizes ranging from 100 to 100000 samples. The authors recorded joint angles, joint velocities, joint accelerations, and joint torques, as well as the commands given to the robot.

Similarly, in [9], the authors present a dataset containing 6145 different pushing motions. The authors recorded the robot tip pose, object pose, and the wrench exerted on the pusher. The authors vary the push interaction by the surface material, the shape of the pushed object, the contact position, the pushing direction, the pushing speed, and the pushing acceleration. The purpose of the dataset is to facilitate research

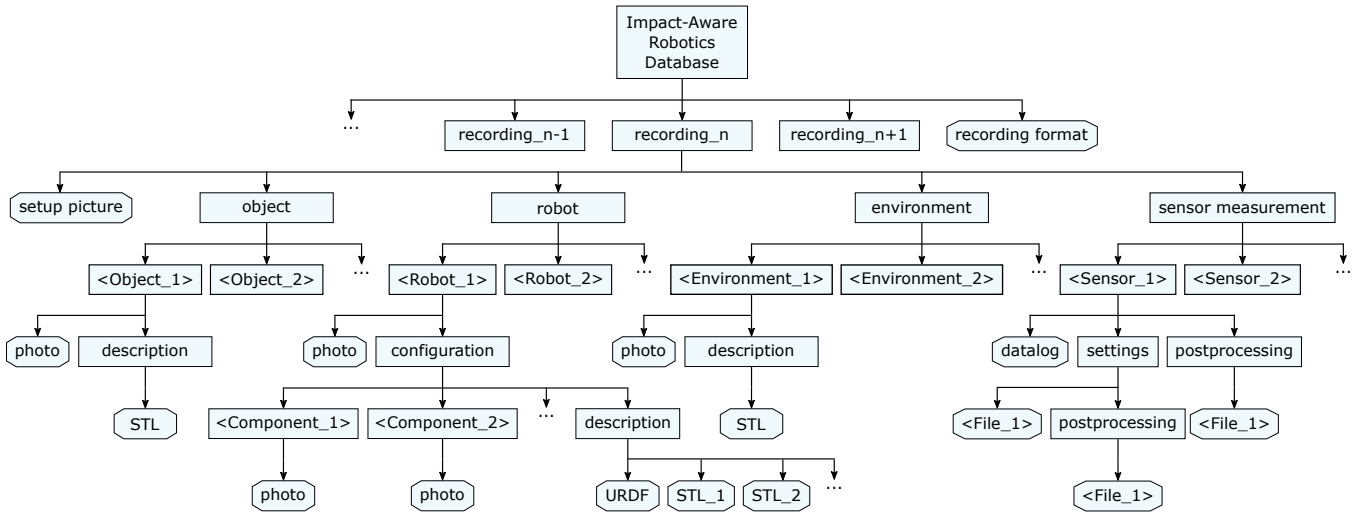


Fig. 2: Structure of a dataset. Each dataset contains a set of recordings, as well as a description of the recording format. Each recording is subdivided into object, robot, environment, and sensor measurement subfolders and contains a picture of the setup used during the data recording.

in model development. The data is stored in the open HDF5 format [17], a format specifically designed to handle large quantities of heterogeneous data, which is widely supported by various programming environments (C++, MATLAB, Python, etc.). The data can be downloaded from [10] and although the metadata is not provided together with the data itself, it displayed on the website [10] instead.

In [11], the authors present a dataset containing 3354 daily performed fine motions in which a robot manipulates an object to interact with another object. The datasets contain the recorded positions and orientations, the force and torque sensor data, and the vision data of the tools and objects manipulated by the robot. The dataset can be downloaded from [12] together with the code to interpret the data, but no instruction files are provided to understand the code.

Similar to our work, but focusing on a different field of research, the KIT Whole-Body Human Motion Database [18] provides an excellent example as a database, that we took as inspiration in the development.

### B. Paper Structure

The remainder of this paper is organized as follows. In Section II, we describe what type of data is stored in the database and we describe the general structure of the a dataset. In Section III, we describe the data collection and storage procedures to obtain experimental data and to store it in a structured way. In Section IV, we describe the contact transition taxonomy that provides the datasets with specific metadata tags. In Section V, we explain how to contribute to the database and how to access 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 on the type of data that can be stored. The conclusion and recommendations for future work are given in Section VII.

## II. THE IMPACT-AWARE ROBOTICS DATABASE

The Impact-Aware Robotics database consists of multiple datasets that are stored using the open HDF5 format [17].

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

and corresponding release motions related to

- objects detaching from the environment,
- objects separating from a robot (in particular, its end effector),
- 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 content structure. 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 (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 time of the experiment.

The **sensor measurement** folder contains all times 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.

Considering Table I, we opt to store a large variety of sensor data corresponding to various robotics scenarios, in contrast with existing databases and datasets. 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. As the main purpose of using the motion capture system is to extract information about the position and orientation of objects, we provide post-processed data (including all tools to make the conversion from raw sensor-data to post-processed data) that simplifies access to this information.

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.

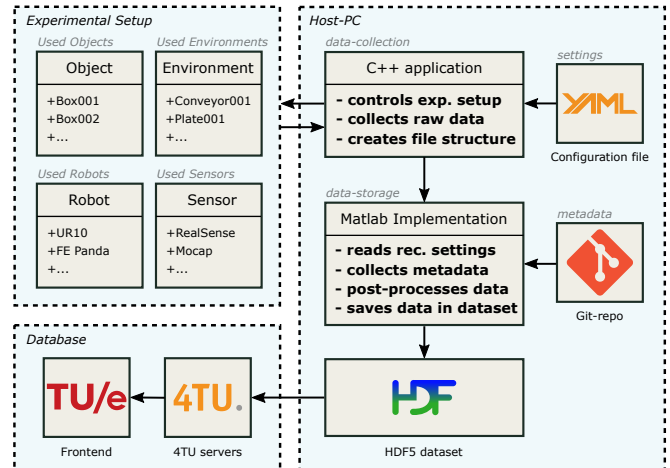


Fig. 3: Overview of the software architecture. The host-PC contains the data-collection application that takes the recording settings from a YAML file to control the experimental setup, collects the data from the recording, and stores the data in a session/recording file structure. The data-storage application takes this raw data and stores it with the associated metadata imported from a YAML file to an HFD5 file. This file is then uploaded to the 4TU servers, and a back-end 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 the following subsections, we will detail our approach on data collection and storage. We share the software online<sup>1</sup>, which provides an example and possibility for other researchers that want to contribute to the Impact-Aware Robotics Database. Further details on our approach can also be found in [19].

#### 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 [20] OpenDAP servers as part of the Impact-Aware Robotics Archive Collection [21]. A web application extracts information from the 4TU servers, which is then displayed on the front-end web interface. Further details about the web application are given in Section V.

<sup>1</sup>The code used for collecting and storing the data is made publicly available and can be found on <https://gitlab.tue.nl/robotics-lab-public/impact-aware-robotics-database>.



## 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 added to the setup, 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) that contains the recording settings (e.g., to decide what sensors will be used to record the experiments) and initializes and configures a recording of a specific experiment. Furthermore, this configuration YAML file contains metadata describing 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 then stored with the raw sensor data in the folder created for that specific experiment. This ensures that all information about one recorded 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 together with the associated metadata in a single dataset according to the structure described in Section 2. From the recording configuration YAML files we know for each recording what objects, environments, robots, and sensors were used. This information is used to retrieve the associated metadata (e.g., pictures, STL files, robot configuration files) from an external Git repository (middle right of Figure 3), and store this information in the dataset together with the raw and post-processed data. Storing the metadata together with the sensor data ensures 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 was first introduced in [22], and 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 (eventually large) repository.

As an illustration, the classification of impact events related to grabbing a box laying isolated 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.

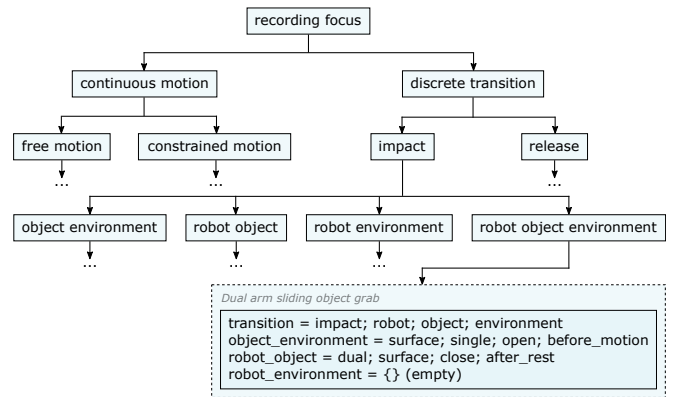


Fig. 4: Illustration of the taxonomy. The drawing illustrates a special leaf in the taxonomy tree associated to a dual-arm grabbing of a box, standing isolated on a surface before being picked up.

We propose a qualitative metric based on the type of contact transitions involving objects, robots, and environment 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 **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 driver causing 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. There are situations where it will be important to allow the creation of 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). Besides the metadata **transition**, three other metadata 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 **object\_robot** metadata is specified as **object\_robot = dual; point**.

To allow for a more detailed level of classification, for each object-environment, object-robot, 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 ante- and post-event combinations of the contact interface state between two entities. Finally, the following ante- 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 position-level 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 position-level entries **open** and **bounce**, as there is no contact after the event. Although it could be possible to further specify the contact motion state, 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**

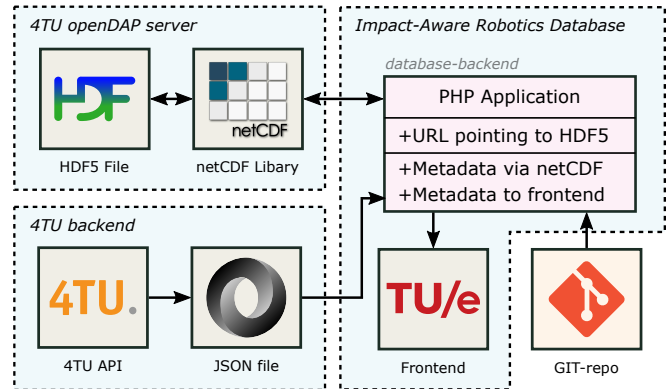


Fig. 5: Overview of the backend structure of the database. A PHP application crawls information from the 4TU backend about the datasets stored under the Impact-Aware Robotics archive collection and retrieves metadata from these datasets stored on the 4TU openDAP servers via a netCDF library. This information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

- **object\_environment = surface; single; open; before\_motion**
- **robot\_object = surface; dual; closed; after\_rest**
- **robot\_environment = (empty)**

Further details about the taxonomy can be found in [22] and on the Impact-Aware Robotics Database website [13].

## V. ACCESSING THE DATABASE

All datasets are published in the Impact-Aware Robotics Archives collection [21] and can be viewed and downloaded via the web interface of 4TUResearchData. However, the 4TU web interface does not provide further options 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 [13] 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 crawls information from the 4TU backend via a JSON file about the datasets stored under the Impact-Aware Robotics archive collection. This JSON file provides the URL pointing to the stored HDF5 files on the OpenDap [23] servers of 4TU. A request for metadata or content of the HDF5 file can then be made by issuing an HTTP request to the OpenDAP server via the NetCDF

[24] library, allowing us to display all information on the frontend of the database, see [13]. It then displays this information, together with all metadata of the objects, robots, and environments retrieved from an external Git-repository, on the frontend of the database.

### A. Contributing to the database

The database contains, at the moment of writing, 16 datasets, 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 have been collected during the research project I.A.M. (Impact-Aware Manipulation)<sup>2</sup>, funded by the European Union through the H2020 Horizon Program. However, the database is intended to continue and grow beyond this project and to provide a platform for data storage for any experiment related to impact-aware robotics.

To this end, we stimulate 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 the experimental data in a predefined structure used for the Impact-Aware Robotics Database. This software provides the end-user with the tools to create a dataset that 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 [13] for the latest information about contributing to the database.

To upload a dataset, the end-user must create an account on 4TU.ResearchData [20]. 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 eduID<sup>3</sup>, 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 licenses<sup>4</sup> and linked to the Impact-Aware Robotics Archive Collection [21] on 4TUResearchData by adding a metadata tag pointing to this collection. Best practice is to contact a TU/e contact mentioned on the website [13] 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.

<sup>2</sup>This work was partially supported by the Research Project I.A.M. through the European Union H2020 program under GA 871899.

<sup>3</sup>An 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/>.

<sup>4</sup>All 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.

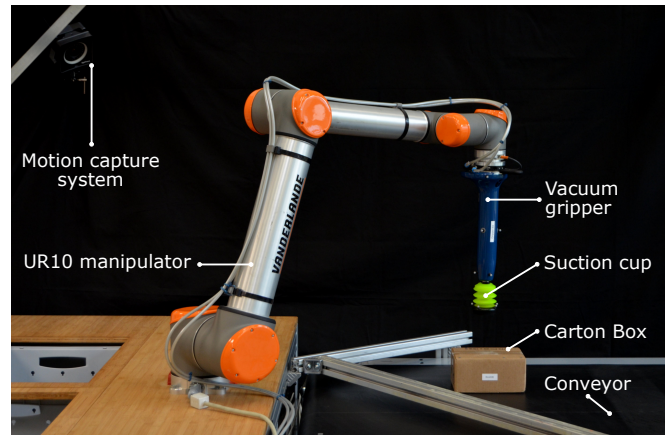


Fig. 6: Picture of the experimental setup.

## 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 in an Impact-Aware Robotics dataset. We consider a dataset that contains 150 recordings of a UR10 manipulator tossing a carton box on a conveyor [25]. We encourage the reader to download and inspect the content of the dataset for better understanding of the content and structure of the dataset<sup>5</sup>. 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. Table II shows the technical details of the equipment used during these experiments.

In each of the experiments, the UR10 robot arm picks up the box from a known location on the conveyor belt. The robot then moves forward and turns the object such that when the object is released, the object impacts the conveyor with a single corner. The conveyor belt is then turned on to transport 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. This means that, considering Figure 4, we can classify these impacts under the *object environment* class, as part of the discrete transition / impact branch. Therefore, the **transition** metadata tag for this dataset is given as

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

The three additional metadata tags are then given as

- **object\_environment:** point; single; bounce
- **robot\_environment:** (empty)
- **robot\_object:** (empty)

as the impact is happening on a single corner of the box, 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

<sup>5</sup>To view and inspect a HDF5 file, one can use the open HDF View software [26].



TABLE II: Technical details about the used equipment

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

track the pose (position and orientation) of these parts with sub-millimeter accuracy at a frequency of 360 frames per second. A RealSenseD415 is used to record each of the experiments and is pointed to 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 II shows the technical details of the equipment used in these experiments, and these details are all stored in the dataset created from this set of experiments. Furthermore, Table II 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 online [25].

## VII. CONCLUSION

We have presented a new database dedicated to the development of impact-aware robotics. The current version of the database contains 16 different datasets related to manipulation, grasping, pushing, and tossing in robotics. We provide software to collect and store experimental data in a

structured way based on an impact-aware contact transition taxonomy, allowing researchers to contribute to the database with their datasets. Furthermore, we provide a 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 will also plan events on main robotics conferences to create awareness of impact-aware robotics and the database we have created.

## REFERENCES

- [1] 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
- [2] Z. Zhang, “The robot joint torque measurements for accidental collisions and intentional contacts,” Apr 2022. [Online]. Available: 10.5281/zenodo.6461868
- [3] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, “The columbia grasp database,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1710–1716.
- [4] 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>
- [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] —, “Dataset supporting publication: A reservoir computing approach for learning forward dynamics of industrial manipulators,” 2016. [Online]. Available: <https://bitbucket.org/athapoly/datasets/src/master/>
- [7] 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.
- [8] 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.
- [9] 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.
- [10] —, “Dataset supporting the publication: More than a million ways to be pushed. A high-fidelity experimental dataset of planar pushing,” 2016. [Online]. Available: <http://web.mit.edu/mcube//push-dataset/>
- [11] 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>
- [12] —, “Dataset supporting the publication: A dataset of daily interactive manipulation,” 2019. [Online]. Available: <https://rpal.cse.usf.edu/datasets.manipulation.html>
- [13] A. Saccon and M. J. Jongeneel, “TU/e Impact-Aware Robotics Database,” Eindhoven, The Netherlands, 2022, [Accessed 10-Oct-2022]. [Online]. Available: <https://impact-aware-robotics-database.tue.nl/>
- [14] Y. Huang, M. Bianchi, M. Liarokapis, and Y. Sun, “Recent data sets on object manipulation: A survey,” *Big Data*, vol. 4, no. 4, pp. 197–216, 2016. [Online]. Available: <https://doi.org/10.1089/big.2016.0042>
- [15] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, “Dataset supporting the publication: The columbia grasp database,” 2009. [Online]. Available: <http://grasping.cs.columbia.edu/>
- [16] A. S. Polydoros, L. Nalpantidis, and V. Krüger, “Real-time deep learning of robotic manipulator inverse dynamics,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3442–3448.
- [17] The HDF Group, “Hierarchical data format version 5,” 2000-2022. [Online]. Available: <https://www.hdfgroup.org/solutions/hdf5/>

- [18] 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.
- [19] 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. [Online]. Available: <https://www.i-am-project.eu/index.php/results/deliverables>
- [20] The 4TU.Federation, "4TU.ResearchData data and software repository," 2008-2022. [Online]. Available: <https://data.4tu.nl/info/en/>
- [21] 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>
- [22] M. J. Jongeneel, J. D. Ouden, and A. Saccon, "D1.1, Publication of I.A.M. Dataset," 2020. [Online]. Available: <https://www.i-am-project.eu/index.php/results/deliverables>
- [23] J. Gallagher, N. Potter, T. Sgouros, S. Hankin, and G. Flierl, "The data access protocol - dap," 2005-2022. [Online]. Available: <https://www.opendap.org/>
- [24] 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. [Online]. Available: <https://www.unidata.ucar.edu/software/netcdf/>
- [25] 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>
- [26] The HFD Group, "HDF View," 2022. [Online]. Available: <https://www.hfdgroup.org/downloads/hdfview/>