# A new median filter application to deal with large windows of missing data in eye-gaze measurements

Arnaud Gucciardi[1,2,*], Monica Crotti[3], Nofar Ben Itzhak[3], Lisa Mailleux[4], Els Ortibus[3], Umberto Michelucci[1,5], Vida Groznik[2,6,7] and Aleksander Sadikov[2,6]

[1] *TOELT llc, Machine Learning Research and Development, Birchlenstr. 25, 8600 Duebendorf, Switzerland*

[2] *Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia*

[3] *Katholieke Universiteit Leuven, Department of Development and Regeneration, Leuven, Belgium*

[4] *Katholieke Universiteit Leuven, Department of Rehabilitation Sciences, Leuven, Belgium*

[5] *Computer Science Department, Lucerne University of Applied Sciences and Arts, 6002 Lucerne*

[6] *NEUS Diagnostics d.o.o., Ljubljana, Slovenia*

[7] *Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, Koper, Slovenia*

**Abstract**

Eye-hand coordination is a challenging skill to measure objectively, especially in children with motor disabilities such as Cerebral Palsy (CP). The recent development of robotic technology provides non-invasive tools for the simultaneous acquisition of eye and hand movement data. One such technology is the remote eye-tracking and virtual-reality systems namely the Kinarm Gaze-Tracker$^{TM}$ installed in the Kinarm Exoskeleton$^{TM}$. Unfortunately, no standard software interface exists to extract the data contained in the Kinarm proprietary files for an efficient further analysis in common programming languages such as Python. Additionally, in the standard Kinarm reports only hand movements parameters are available, while eye movements are only stored as raw data files. These limitations lead to difficulties in performing a complete analysis of eye-hand coordination in research settings. Additional problems can arise in the case of missing data (due to loss of tracking). The software described in this paper allows the extraction of the hands and eye-gaze time series for efficient further analysis directly from the raw data. Furthermore, a study of the distribution of missing data is presented. Finally, this paper describes a revised median filter application to deal with large windows of missing data.

**Keywords**

Eye-Gaze, Software, Median Filter, Unilateral Cerebral Palsy, Kinarm

## 1. Introduction

Cerebral Palsy (CP) describes, based on an international consensus, *”a group of permanent disorders of movement and posture, causing activity limitations, that are attributed to non-*

*DETERMINED 2022: Neurodevelopmental Impairments in Preterm Children — Computational Advancements, August 26, 2022, Ljubljana, Slovenia*

[*] Corresponding author.

arnaud.gucciardi@toelt.ai (A. Gucciardi); monica.crotti@kuleuven.be (M. Crotti); nofar.benitzhak@kuleuven.be (N. Ben Itzhak); lisa.mailleux@kuleuven.be (L. Mailleux); els.ortibus@uzleuven.be (E. Ortibus); umberto.michelucci@toelt.ai (U. Michelucci); vida@neus-diagnostics.com (V. Groznik); aleksander.sadikov@fri.uni-lj.si (A. Sadikov)

https://github.com/toelt-llc/KiPy (A. Gucciardi)

*progressive disturbances that occurred in the developing foetal or immature brain. The motor disorders of CP are often accompanied by disturbances of sensation, perception, cognition, communication, and behavior, by epilepsy, and by secondary musculoskeletal problems"* [1]. The development of movement capacity in addition to muscle tone and posture [1, 2] is affected by brain injury in the prenatal, perinatal, and postnatal phase up to the age of two years [3]. CP is a neurological disorder recognized as the leading cause of childhood motor disability and its appearance is estimated from 1 to nearly 4 per 1,000 live births [1]. Children with CP develop a wide range of conditions that may affect their functional abilities [4]. The clinical variability of children with CP can be explained by the heterogeneity of the underlying brain injury [5], which also affects the nonmotor pathways of the developing brain. Among these, the visual network is often affected in children with CP [6]. This leads to impairments in visual function [7] which is a prerequisite for typical eye-hand coordination [8] since it is crucial for planning and performing movements [9, 10]. Therefore, children with CP also suffer from difficulty in grasping objects [11].

Accurate reaching develops in children between 5 to 13 months of age [12] and is fine-tuned over a longer period of several years (often more than 8) [12, 13]. In this process, eye-hand coordination plays a fundamental role. Despite a large amount of research in this area, several aspects of the development of eye-hand coordination remain unsolved in children with CP. In addition to motor problems, 60 to 75% of children with CP also have visual deficits [14, 7]. Eye-tracker systems, which allow quantification of looking behaviour, nowadays are considered a valid tool for investigating visuomotor coordination in CP children [15, 16]. Furthermore, their implementation with robotic technology can provide an in-depth quantification of eye-hand movement impairments in the pediatric neurological population. Previous studies [15, 16, 17] attempted to quantify eye-hand coordination in children with CP using different methodologies. Results showed that children with CP have increased visual attention towards the impaired limb during object grasping and reaching [18, 17] and impaired anticipatory visual control in eye-hand coordination when compared to typically developing children [15, 19].

One novel application is the use of the Kinarm Exoskeleton [20] which allows an in-depth quantification of bimanual motor control during symmetrical and asymmetrical tasks and the simultaneously recording of eye movements via the Kinarm Gaze-Tracker [21]. With this technology, both motor and gaze measures can be seamlessly integrated for effective experimental control and data analysis. To our knowledge, no previous work fully evaluated eye-hand coordination in children with CP with the use of such a technology, although investigating this relationship would provide a better understanding of the complex function of the visual motor system. In addition, such results would provide useful information for clinicians and researchers to be applied in diagnosis and possible rehabilitation settings.

The first step for such an analysis is the extraction of information, such as gaze and hands parameters over the time course of a movement, from the Kinarm saved files. This is not a trivial task as the data is stored in proprietary file formats, making the desired analysis very difficult. This work addresses this problem and describes a possible solution. The contributions of this paper are fourfold. First, it describes a software framework able to extract hands and eye-gaze coordinates with time from the Kinarm Exoskeleton files as time series (a time series is, in its most common occurrence, a sequence of points taken at successive equally spaced points in time). Secondly, this work describes and discusses a variation of the median filter for data with large windows of missing values. Thirdly, this paper analyses missing data windows in terms of distributions of width and frequency of the gaps in the data in two separate cases. Finally, it demonstrates the median filter variation described in this paper applied to two different examples with very different distributions of missing data windows.

The paper is organised as follows. In Section 2, the Kinarm Exoskeleton and the eye-gaze module are briefly described. The data and the median filter are also discussed and defined respectively. In Section 2.4, the different tasks possible with the Kinarm exoskeleton are described. In Section 2.6, the monitored parameters are listed. In Section 3, the Software is described. In Section 4, the results are presented and in Section 5, the conclusions are discussed.

## 2. Methodology and Data

### 2.1. The Kinarm Exoskeleton

Data collection was carried out with the Kinarm Exoskeleton Lab (BKIN Technologies, Kingston, ON, Canada) [20] combined with an integrated EyeLink 1000 Plus eye tracking system (SR Research, Ottawa, ON, Canada) [21]. The Kinarm Exoskeleton can be seen in Figure 1. The Kinarm Exoskeleton Lab (BKIN Technologies, Kingston, ON, Canada) allows movement of the arm in the horizontal plane such as flexion and extension of the shoulder and elbow joints [20]. The hands are free to interact with objects in the environment surrounding the subject. Patterns of joint motion are recorded and the system computes muscular torques, allowing the study of upper limb movement and coordination. The use of Kinarm Lab's operating system and its control software, Dexterit-E$^{\text{TM}}$ [20], allows data collection in a user-friendly way. At the end of each experimental task, reports can be extracted from the Dexterit-E$^{\text{TM}}$ software in a Comma-Separated Values format (CSV) where a division in LEFT and RIGHT-hand parameters is available.

### 2.2. Eye-Gaze tracking system and parameters

Eye tracking and gaze estimation systems are well-established techniques used to study eye movements and position, both in clinical and research settings [22]. In eye tracking systems, the eye position is calculated through different sequential steps (detection of the eyes, interpretation of eye positions, and frame-to-frame tracking) with the help of the pupil or the iris centre [23]. Gaze estimation, that is, the process made to estimate and track the 3D line of sight, is calculated from the analysis of eye movements through a device called gaze tracker [23]. A gaze tracker simultaneously records the location of the eye position and its motion to determine the direction of the gaze [24]. The EyeLink 1000 Plus system (SR Research, Ottawa, ON, Canada) integrated into the Kinarm Exoskeleton Lab (BKIN Technologies, Kingston, ON, Canada) allows recording binocular eye movements at up to 2000 frames per second. Camera images are processed using a real-time operating system from which gaze data is recorded. More information on the eye gaze estimation system can be found in the work of A. Kar et al. [25]. Eye tracking systems allow the quantification of different types of eye movements such as fixations, saccades, and smooth pursuit.

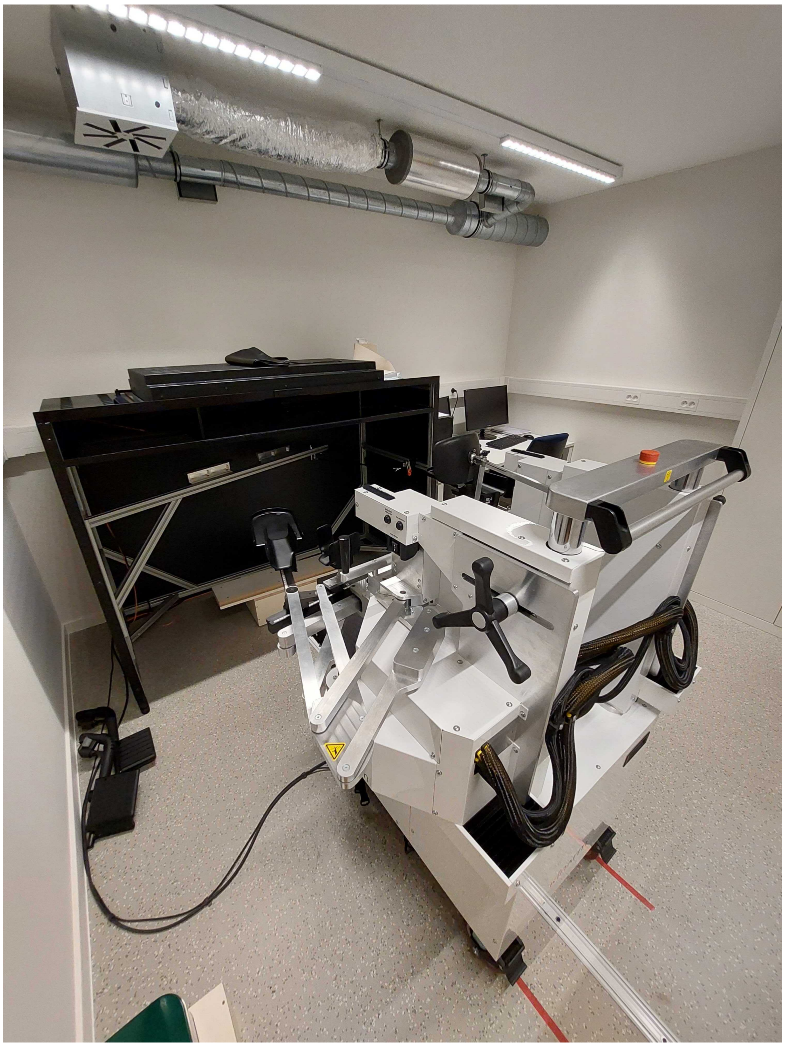

**Figure 1:** Kinarm Exoskeleton Lab (BKIN Technologies, Kingston, ON, Canada) [20] with integrated EyeLink 1000 Plus system (SR Research, Ottawa, ON, Canada) [21].

This paper will specifically focus on the data used to estimate fixations and saccades [26]. A visual fixation is the maintenance of gaze in a single location or area [27]. Fixations phases are defined as moments where the eyes are stationary between movements while the visual input occurs. A saccade is a quick and simultaneous movement of the eyes between phases of fixation in the same direction [27]. Saccades are mainly used for orienting the gaze towards an object of interest. They can be triggered voluntarily or involuntarily, with both eyes moving in the same direction.

Fixations and saccades can be quantified in terms of different parameters which can be used for further analysis (i.e., eye-hand coordination). The mathematical algorithms to compute them are not discussed in the present paper. For further information, the interested reader can refer to the following papers [28, 25].

The data output from Kinarm software includes gaze position (*x* and *y* positions), gaze direction, pupil position (*x*, *y* and *z* positions) and area, time stamps, and events such as start and end of fixations and saccades. The Kinarm software automatically saves these features in stored files, making them available to researchers. If necessary, averages and other statistical analyses of the available metrics are then possible. Additionally, by adding the formerly listed features, the parameters mentioned below can be calculated.

Fixations and saccade parameters include:

- *Fixation Duration* - total duration of a fixation in seconds.
- *Fixation Area* - position where the fixation is recorded in meters.
- *Saccadic Peak velocity* - the highest velocity recorded during the saccade in metres per second.
- *Saccadic amplitude* - the horizontal displacement during eye movement in meters.
- *Saccade Duration* - total duration of a saccade in seconds.
- *Gaze latency* - time taken from the appearance of a target to the beginning of a saccade in response to that target in seconds.
- *Gaze Accuracy* - the average distance between the target and the participant's eye position in meters.

## 2.3. Data

In the present paper, data from two subjects, namely A and B, are discussed. Both participants have been diagnosed with unilateral CP (mean age: 11y4m). Test subjects are chosen only if they have minimal ability to actively grasp and hold an object and sufficient cooperation to perform the assessments. None of the participants received botulinum toxin injections six months before testing or had a history of arm surgery two years prior to the assessment. Each experimental session lasted about one hour. After the experimental session, the data were anonymised and extracted from Dexterit-E$^{TM}$ software in a Comma Separated Values (CSV) format where a division in the left and right upper limbs and the left and right eye gaze parameters is available.

The available records for this study consist of two separate file groups: group A is defined as the group that contains the first cohort of experiments, and group B is the second group. Both groups contain eight files for the three different tasks studied: *Ball On Bar* task (2 files), *Object Hit* task (2 files), and *Visually Guided Reaching* task (4 files). The files of the second group contain incomplete and damaged data: missing values are detected for the Gaze X and Gaze Y positions. In addition, the remaining values translate into a different gaze behaviour.

A comparative view of the gaze behavior in the three tasks can be seen in Figure 4. Due to the individual differences in the task protocol and the numerous rows that make up the experiment dataframes, the CSV files are of various sizes. In Table 1, the average total lines and the size of the CSV files related to subjects A and B are reported.

Size differences are directly correlated with the duration of recorded trials ( number of attempts) on tasks (i.e., exercise type).

**Table 1**
Time series length from groups A and B. The *Average file size* is the total length of the CSV file, the *Average total lines* represents the average line count of a type of task, and the *Maximum size* is the maximum line count.

| Exercise type | Average file size | Average total lines | Maximum size |
|---|---:|---:|---:|
| Ball On Bar | 81 032 | 79 918 | 130 621 |
| Object Hit | 15 274 | 10 427 | 28 080 |
| Visually Guided Reach. | 49 787 | 46 002 | 107 427 |

## 2.4. Experiment tasks and datasets differences

In this paper, the focus is on three custom experimental tasks (i.e., Kinarm standard test-KST), namely, the *Ball On Bar*, *Object Hit*, and *Visually Guided Reaching* task. Each task is standardised and performed with the Kinarm Exoskeleton, allowing the assessment of upper limbs' motor control and the simultaneous acquisition of eye-movements data [29]. A description of the KST taken from the Kinarm manual is provided below.

**Ball On Bar** The Ball on Bar task assesses the ability of subjects to perform a motor activity that requires coordination of the two arms. [30] A virtual bar is presented between the subject's hands, and a virtual ball is placed on the bar. The objective of the task is to move the virtual ball on the bar to successively presented targets as quickly and accurately as possible.

**Object Hit** The Object Hit task [31] assesses rapid motor skills throughout the workspace. It is developed to assess the ability of a subject to select and engage in motor actions with both hands over a range of speeds and a large workspace. Good performance requires the ability to generate a goal-directed motor action on a moving target, bimanual planning to select which arm to use to hit each object, and spatial awareness across the workspace.

**Visually Guided Reaching** The purpose of the Visually Guided Reaching task is to quantify voluntary control directed toward the goal [32]. This task assesses visuomotor response time and arm motor coordination. During this task, a central target is presented, and the subject must move a cursor (white circle) representing hand position to this target.

For each task, the Kinarm software can automatically compute a standard report (SR), as well as a CSV file [33]. CSV files contain a metadata header with calibration and experimental set-up information such as the Kinarm experiment instructions, the accessories used and the calibration values. The file header also includes the definition of all recorded channels that measure features with their unit of measurement. For each task, a different number of trials are presented, namely an attempt to accomplish the exercise. Note that the number of trials varies depending on the task exercise in both groups (A and B). For the *Object Hit* task, there is a single trial in each file. A total of 1 to 3 trials are presented for the *Ball On Bar*, and 1 to 24 for the *Visually Guided Reaching* task. In the trial information, the different time series are provided in terms of a dataframe where each row contains the data measured at a given time in milliseconds. Rows are separated by 1 ms intervals. The individual trials also include a trial header with several lines regarding metadata that needs to be removed in pre-processing.

## 2.5. Files processing

The experiment files are presented as CSV files containing dataframes. In this case, the columns are the measured kinematics features, and the rows are the individual values of each frame saved. The extraction process allows the reading of the dataframes contained as raw content. Furthermore, the goal is to obtain the experiment data as time series. In the files, each trial is represented as a separate dataframe. The pre-processing algorithm allows us to extract these dataframes by splicing the CSV files and removing the general metadata header and the individual trial headers. This is done for each of the four possible tasks. The methods involved can be used for new experiment files to automate the analysis.

**Table 2**
Parameters of interest selected from the list of features.

| Parameter (measurement unit) | Definition |
|---|---|
| **Sample duration (s)** | Interval between each data sample taken |
| **Sample count** | Total number of data points for each measurement |
| **Frame number** | Individual identifier of each row in the dataframe |
| **Frame time(s)** | Unique timestamp to each row.<br>Starts at 0 for each trial |
| **Event name** | Name of the event, automatically given by Kinarm |
| **Event time (s)** | Frame time of the detected event |
| **Right and left hand position (m)** | Position of each hand, X and Y component |
| **Gaze position (m)** | Gaze position in global coordinates, X and Y component |
| **Right and left hand speed (m/s)** | Individual speed of each hand |
| **Right and left hand acceleration (m/s$^2$)** | Individual acceleration of each hand |
| **Ball position (m)** | Only for *Ball-On-Bar* exercises, ball X and Y position on screen |
| **Ball relative position (m)** | Only for *Ball-On-Bar* exercises, ball relative position on the bar |

## 2.6. Monitored parameters

The features of interest used for visualisation and analysis are listed in Table 2. Note: the total real time of each trial is equal to *sample count* multiplied by *sample duration*. It can also be retrieved by looking at *frame time* in the last row of the selected trial. All tasks from both file groups are saved in a single trial (that is, in one dataframe). Some files contain more than one trial: in the first group of files, the two *Ball-On-Bar* tasks include, respectively, 3 and 2 trials, and the two *Visually Guided* tasks include 3 (for the child practise set) and 24 (for the complete exercise) trials. And in the second group, two out of four *Visually Guided* tasks contain 9 trials.

## 2.7. Analysis of Missing Data in the Eye-Gaze Measurements

As mentioned in Section 2.3, some of the experiment data lines contained in the CSV files of the second group of files appear incomplete. The features presented previously require continuous data to study the detailed actions of gaze and hands. In practise, the missing values detected in the files make a complete analysis and visualisations impossible; since both gaze and hands position data in multiple time windows are missing. Indeed, events like fixations and saccades cannot be entirely detected and understood when positions are only partially saved during a given time window. As a first step, the number of extended regions of continuous missing data, or *gaps*, are counted and their respective lengths measured. Depending on the type of trial, the complete length of recorded experimental data changes, but it is possible to estimate the percentage of missing data for each trial to quickly quantify the impact on the final analysis. And this is possible for each experiment regardless of the length. This result is represented in Table 3.

**Table 3**
List of the nine trials contained in the CSV report file of a *Visually Guided Reaching* task for group B. The *Average gap size* column shows the rounded averages over all the detected gap sizes of size 1 and more. The amount of gaps detected in a trial is noted in the *Gaps count* column. *Max gap size* contains the size of the largest gap in the trial. *Not a Number values* (NaN values) column contains the total amount of NaN values in each trial, the *Total length* column contains the time series entire length.

| Trial ID | Average gap size | Gaps count | Max gap size | NaNs values | Total length | **NaNs** % |
|---|---|---|---|---|---|---|
| 0 | 181 | 3 | 281 | 543 | 3353 | 16.2 |
| 1 | 98 | 10 | 318 | 980 | 8414 | 11.7 |
| 2 | 60 | 9 | 192 | 539 | 5857 | 9.2 |
| 3 | 126 | 9 | 576 | 1132 | 4838 | 23.4 |
| 4 | 41 | 32 | 228 | 1299 | 8258 | 15.7 |
| 5 | 65 | 10 | 222 | 654 | 6665 | 9.8 |
| 6 | 65 | 36 | 1051 | 2355 | 8880 | 26.5 |
| 7 | 200 | 1 | 200 | 200 | 1269 | 15.8 |
| 8 | 210 | 1 | 210 | 210 | 2586 | 8.1 |
| Mean | 116 | 12.3 | 364.2 | 879.1 | 5569 | **15.2** |
| Std Dev | 62 | 12.1 | 267.7 | 633.9 | 2591 | 6.0 |

The selected example file from a task of *Visually Guided Reaching* in group B contains 9 separate trials, all of which are made of a large amount of NaN values. The minimum value of missing values is 8.1% (in trial 8) of the total data points, while the maximum is 26.5% (in trial 6). The mean proportion of missing values over the 9 trials is 15.2%. The example is representative of group B, where each file contains randomly placed NaNs gaps of various sizes.

**Table 4**
List of trials containing gaps from a CSV report file of a *Visually Guided Reaching* task from group A. The file contains 24 trials, of which only 4 contain NaN values represented in the table. The remaining 20 trials do not contain gaps or a single NaN value. The *Average gap size* column shows the rounded averages over all the detected gap sizes of size 1 and more. The amount of gaps detected in a trial is noted in the *Gaps count* column. *Max gap size* contains the size of the largest gap in the trial. *NaNs values* column contains the total number of NaN values in each trial, the *Total length* column contains the time series entire length.

| Trial ID | Average gap size | Gaps count | Max gap size | NaNs values | Total length | **NaNs** % |
|---|---|---|---|---|---|---|
| 4 | 24 | 1 | 24 | 24 | 5559 | 0.4 |
| 10 | 104 | 1 | 104 | 104 | 4941 | 2.1 |
| 20 | 144 | 1 | 144 | 144 | 5303 | 2.7 |
| 21 | 83 | 1 | 83 | 83 | 4312 | 1.9 |
| Mean | 14.8 | 1 | 14.8 | 88.7 | 4177 | **1.8** |
| Std Dev | 43.3 | 0 | 43.3 | 43.3 | 1340 | 0.8 |

## 2.8. Median Filter

To preserve the recorded data saved in the incomplete files and to reduce noise, a median filtering technique is applied. The method is compatible with missing data. Due to the noise reduction obtained, the local trend can be preserved by replacing incorrect data. The median filter eliminates extreme or empty values without having to do a mean averaging of the neighbour values, which would heavily impact the correct values. Although typically used for image pre-processing [34], the algorithm can be applied to one-dimensional signals [35] as is the case here. When used on one-dimensional input, the process is simplified : the neighbourhood includes values before and after the index.

**Definition 1.** *Given an array of M values* $(X_1, ..., X_M)$*, the median filter (MF) of size Q (in this paper Q is taken to be odd for simplicity of notation) is a mapping* $\mathbb{R}^M \to \mathbb{R}^M$*. By defining* $Q = 2n + 1$*, the output of the MF will be an array with elements* $X_i^f$ *given by*

$$X_i^f = median(\{X_i - n, ..., X_i + n\}) \tag{1}$$

*The median filter algorithm replaces each individual value* $X_i$ *(starting from* $i = n$*) of the original array by the median of the following and previous n values. The* ***window*** *or* ***filter size*** *of total size Q defines the amount of neighbour values considered to compute the new filtered signal* $X^f$*. Since there are no values preceding the first and last elements of the signal, the first and last values are repeated until enough values are reached to fill the window.*

Figure 2 shows an example of the median filter applied to a hypothetical data set with some quantity ($x_i$) measured at specific time points $t_n$. In Step 1, a time window is selected (in Figure 2 the selected time window includes the time points $t_2$ to $t_6$). In Step 2, the values are sorted, and the median value is selected (Step 3). The time window then moves along the entire data set.

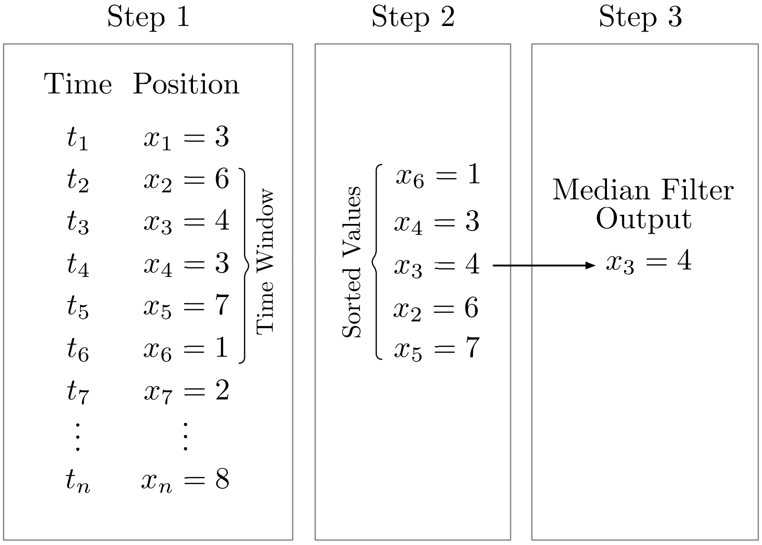

**Figure 2:** Diagram of an example application of the median filter to a hypothetical dataset. A hypothetical dataset with some quantity ($x_i$) measured at specific time points $t_n$ is considered. In Step 1, a time window is selected (in this example the selected time window includes the time points $t_2$ until $t_6$). In Step 2, the values are ordered, and the median value is selected in Step 3.

The median filtering is applied to the experimental data in Figure 5, with two filter sizes presented. Fixations and saccades rely on specific windows of gaze position. It is important to note, as a downside, the risk of losing short movement detections as the filter size selected gets larger. Movements shorter than the filter window might be lost. This is a risk in the time series containing gaze positions if the filter is longer than some of the actual events happening within the task. The window size choice is an important parameter further developed in the article.

## 2.9. The Median Filter for Large Windows of Missing Data (MFLWMD)

A statistical analysis of the features of saccades and fixations is made difficult, if not impossible, when large windows of missing data are present. The most appropriate solution is to split the time series each time a gap larger than a specific size is encountered. As a possible solution to the presence of large windows of missing data, the following median filter application is presented. Given a certain measurement of a generic quantity $x_i$ (for example, the $x$ coordinate of the eye-gaze) at various $t_i$ time points (for $i = 1, ..., M$). A median filter of size $n$ can be applied by sliding a window of size $n$ on the measured data. Let us also suppose that $M \in \mathbb{N}$ windows of missing data, in which gaps of size $s_i$ are present at various positions along the array $x_i$. If a missing data window is encountered, there are two possible scenarios. Let us indicate with $g \in \mathbb{N}$ an integer that can be called *threshold*.

1. $s_i \geq g$: the array $x_i$ is split at that point and every calculation of the statistical estimators is stopped. The two parts are considered for all purposes as separate files.
2. $s_i < g$: the median filter can be applied simply by removing the missing data and considering only the available values.

The best choice of $g$ is, of course, related to the median filter window size. $g$ should be larger than $n$ to make this proposed variation of the median filter meaningful. From the available missing values files and experiments, and since the filters span across windows of size $n$, the authors propose that a good choice is $g \gtrsim 2n$ to $3n$. If the threshold is less than $n$, complete missing value gaps are never considered since the filter size will cover all possible cases. On the other hand, if the threshold is greater than $3n$ the starting and end points are considered too far apart for the filter to sufficiently fill the gap considered.

## 3. Software Functionalities

The *KinarmPython* extraction library presented in this article, called KiPy, can read CSV files generated from the Kinarm Dexterit-E$^{\text{TM}}$ software [33] and support their analysis. The software provides the user with the ability to parse all accessible kinematics logs. Concerning the research questions previously mentioned in Section 2.3, the software extracts relevant information and can create visualisations which can also be easily changed by the user. The tool currently uses command-line scripts and Python functions; the goal is to keep it simple, configurable, and performant. The application is written in Python 3.8 and requires the additional Pandas, NumPy, Pickle, and Matplotlib libraries. It is open source and is available on GitHub [36]. Once the files are read, all metadata are excluded, and the remaining data are read as a Pandas DataFrame. The Pandas DataFrame is an efficient data structure to store structured data and provides powerful functions to filter and search for specific rows of columns.

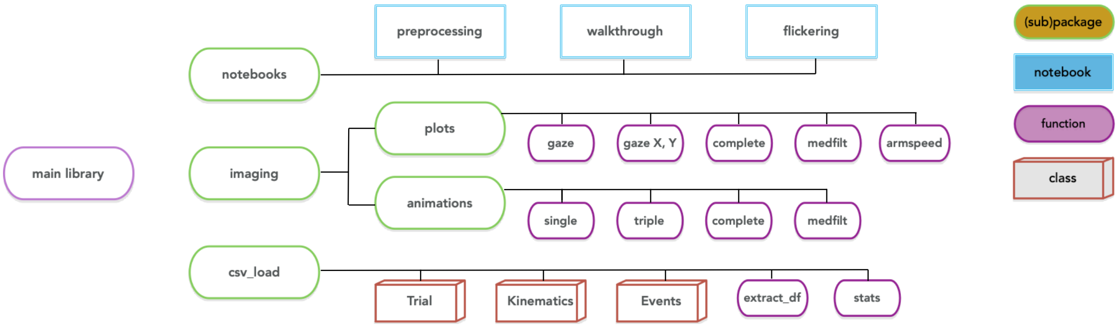

**Figure 3:** Software Component Diagram. The main components of the software package KiPy are depicted in this Figure. The different parts (sub-packages, notebooks, functions and classes) are represented with different colors explained in the legend. Further documentation is available online [36].

From the extracted dataframes, the algorithm accesses the visual data events. For each trial, the duration and count of all the events are given. This information can be further used to analyse the frequency of the events over a single trial, a complete experiment, or a group of experiments (e.g., the events statistics for a given type of experimental task).

# 4. Results

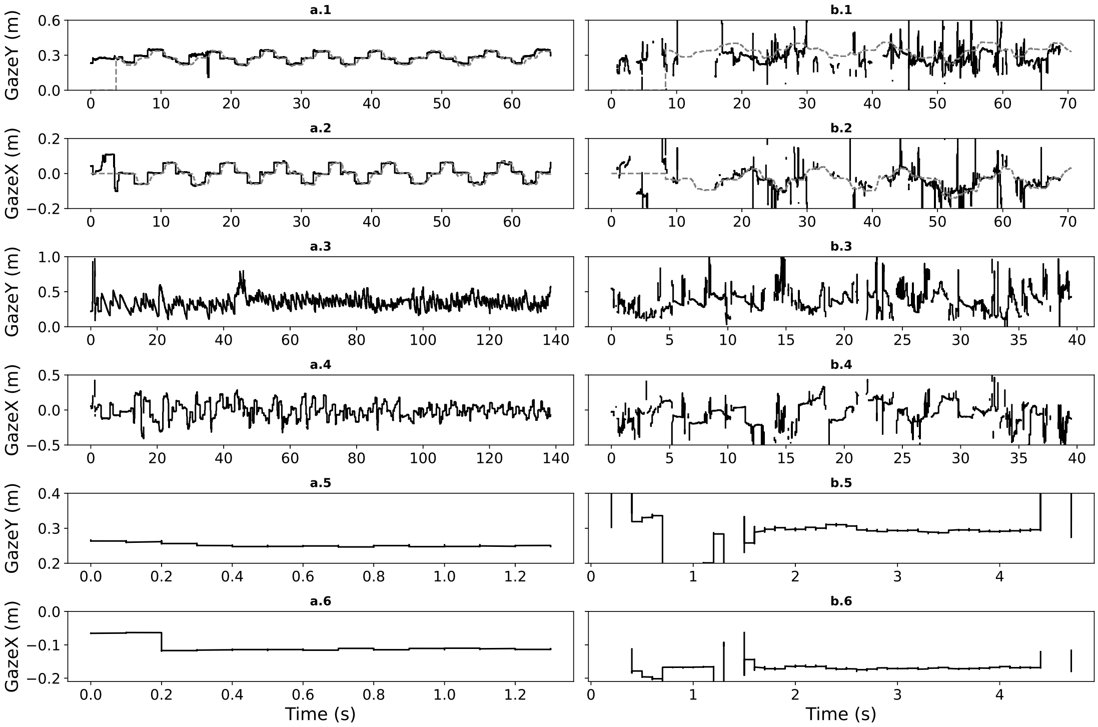

**Figure 4:** Gaze comparison: group A and B. Group A and group B tasks are represented on the left and right sides, respectively. Numbers 1 to 2 represent X and Y movements of the gaze on the *Ball On Bar* tasks. The ball position is shown in dotted grey lines. Numbers 3 to 4 correspond to an *Object Hit* task, and numbers 5 to 6 correspond to the *Visually Guided Reaching* task.

## 4.1. Eye-gaze and Hand Position Data

The central idea behind the software tool is to give the user the ability to read the CSV files and visualise and analyse the Kinarm experiments. The software makes it possible to first, given any task input, visualise the movements of the gaze and hands. The visualisation can be done with both static plots and animations. Animations are short videos that the user can produce at a chosen speed. Examples of visualisations can be seen in Figure 4 for gaze movements and in Figure 6 for hands movements over the duration of the task in the three different tasks. The figures display the differences between the two groups of files. One can identify gaps of missing values, saved as NaNs in files, and numerous peaks, hence the name of noisy *flickering* data.

## 4.2. Median Filter Application

Examples of the distribution of NaN gaps in gaze time series are presented in Tables 3 and 4. The examples selected are described with a precise description of the impact of the missing values. The difference is comparably significant over all the tasks and trials. In group A of the files, only 17 of 65 total trials contain missing values, and the average proportion of missing values within the 17 affected trials is 2.8%. However, 29 of the 29 trials in group B have missing values, which represent 18.8% of the total trial values on average. Due to the high variation rates and missing values in group B, the median filter technique was developed and applied to gaze X and Y time series of this group. An example of a median filter application is shown in Figure 5. In each of the three panels, missing data points are not replaced by the median filter.

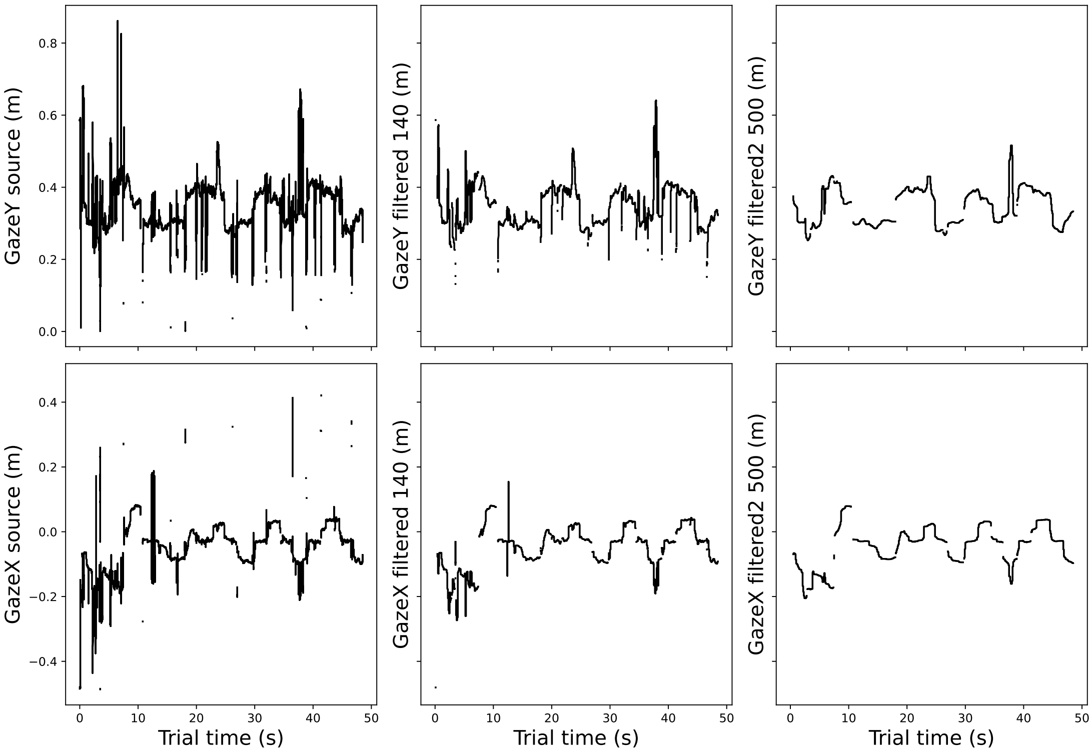

**Figure 5:** Median filter application: two different filter sizes applied to a *Ball On Bar* task result file. In this example application, extreme flickering values are effectively removed. On the left panel are the original Gaze X and Y over time. On the center panel a filter of size 140 is applied. On the right panel the filter is of size 500. The filter is centered on the current value if this value is not a NaN, as presented in Section (2.8).

Two different filter sizes are applied: 140 and 500, respectively, on the centre and right panels, with the filter centred on the replaced value. On the gaze Y, the range of values went from 0.0-0.83, to 0.17-0.61 in the 140-filtered time series, and to 0.22-0.46 in the 500-filtered time series. A similar result is obtained for the Gaze X values. Although some flickering outliers remain in the 140-filtered time series, they are all smoothed in the 500-filtered time series. In this example, the smoothing appears to be effective. In practice, a 500-sized filter may be damaging for some tasks since it represents half a second of gaze movements; ruling out the gaze events with shorter durations. The filter of 500ms is used here as a display example of a high value.

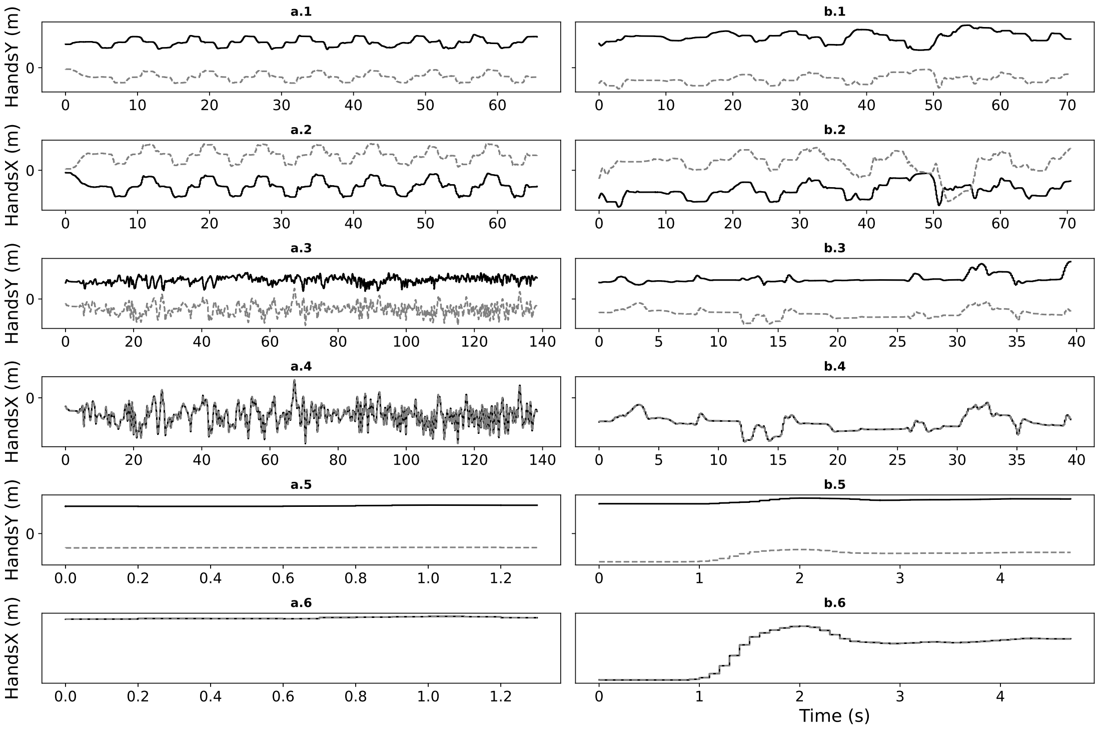

**Figure 6:** Hands movements comparison: group A and B. Group A and B tasks are represented on the left and right sides, respectively. Number 1 to 2 represent X and Y movements of the hands on the *Ball On Bar* tasks. The ball position is not plotted for visibility. Numbers 3 to 4 correspond to an *Object Hit* task, and numbers 5 to 6 correspond to the *Visually Guided Reaching* task. In black: left hand position, gray: right hand position.

# 5. Conclusions

Eye movement tracking combined with bimanual motor movement recording allows a complete experimental analysis during symmetrical and asymmetric tasks. From the combined measurements, precise parameters can be extracted and analysed. In this paper data extracted from measurements obtained with the experimental setup described in Section 2.1 and 2.2, is used to highlight the possible applications of the KiPy software analysis library. For each task, it is possible to extract the recorded parameters, including the gaze and hands positions with the timestamp, thanks to various automated functions. Additionally, statistics of ranges of movements and speeds can be individually calculated. If files with flickering and missing values are present, the software can smooth the flickering signals, detect the size of missing value gaps, and effectively count the number of consecutive data sections. The visualisations and statistics on incomplete experimental data are therefore possible, allowing the analysis of all datasets, including those that present a high percentage of missing data. In the available data, half of the files displayed such problems.

The ability to obtain as much information as possible from the experiments is crucial. The variation of the median filter described in this paper can be applied to all kinds of datasets. To derive an optimal strategy for the choice of the parameter $Q$, more experimental and different data will be needed. This analysis is planned for a future publication.

To the best knowledge of the authors, no previous work has looked at the automatic detection and filtering of missing data in Kinarm report files. With access to more statistical samples, the KiPy software can be confidently used to extract the desired features and make the analysis much more robust. The authors plan to use and apply the KiPy software to the analysis of a larger dataset to study the eye-hand coordination in a group of children with unilateral cerebral palsy.

## Acknowledgments

This work was supported by the project: “PARENT” funded by the European Union’s Horizon 2020 Project MSCA-ITN-2020 – Innovative Training Networks Grant No. 956394.

## References

[1] P. Rosenbaum, N. Paneth, A. Leviton, M. Goldstein, M. Bax, D. Damiano, B. Dan, B. Jacobsson, et al., A report: the definition and classification of cerebral palsy april 2006, Dev Med Child Neurol Suppl 109 (2007) 8–14.

[2] D. R. Patel, M. Neelakantan, K. Pandher, J. Merrick, Cerebral palsy in children: a clinical overview, Translational pediatrics 9 (2020) S125.

[3] M. Sadowska, B. Sarecka-Hujar, I. Kopyta, Cerebral palsy: Current opinions on definition, epidemiology, risk factors, classification and treatment options, Neuropsychiatric disease and treatment 16 (2020) 1505.

[4] Kinarm, National institute for health and care excellence (uk). cerebral palsy in under 25s: assessment and management, 2022. URL: https://www.ncbi.nlm.nih.gov/books/, available online.

[5] L. Mailleux, C. Simon-Martinez, K. Klingels, E. Ortibus, H. Feys, Brain lesion characteristics in relation to upper limb function in children with unilateral cerebral palsy, in: Factors Affecting Neurodevelopment, Elsevier, 2021, pp. 411–420.

[6] A. Guzzetta, B. Fazzi, E. Mercuri, B. Bertuccelli, R. Canapicchi, J. Van Hof-van Duin, G. Cioni, Visual function in children with hemiplegia in the first years of life, Developmental Medicine and Child Neurology 43 (2001) 321–329.

[7] E. Fazzi, S. G. Signorini, R. La Piana, C. Bertone, W. Misefari, J. Galli, U. Balottin, P. E. Bianchi, Neuro-ophthalmological disorders in cerebral palsy: ophthalmological, oculomotor, and visual aspects, Developmental Medicine &amp; Child Neurology 54 (2012) 730–736.

[8] S. Chokron, G. N. Dutton, Impact of cerebral visual impairments on motor skills: implications for developmental coordination disorders, Frontiers in psychology 7 (2016) 1471.

[9] F. R. Sarlegna, R. L. Sainburg, The roles of vision and proprioception in the planning of reaching movements, Progress in motor control (2009) 317–335.

[10] S. A. Mutalib, M. Mace, H. T. Ong, E. Burdet, Influence of visual-coupling on bimanual coordination in unilateral spastic cerebral palsy, in: 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR), IEEE, 2019, pp. 1013–1018.

[11] O. Martinie, C. Mercier, A. M. Gordon, M. T. Robert, Upper limb motor planning in individuals with cerebral palsy aged between 3 and 21 years old: A systematic review, Brain sciences 11 (2021) 920.

[12] C. von Hofsten, L. Rönnqvist, Preparation for grasping an object: a developmental study., Journal of experimental psychology: Human perception and performance 14 (1988) 610.

[13] M. Favilla, Reaching movements in children: accuracy and reaction time development, Experimental Brain Research 169 (2006) 122–125.

[14] A. Schenk-Rootlieb, O. Van Nieuwenhuizen, P. Van Waes, Y. Van der Graaf, Cerebral visual impairment in cerebral palsy: relation to structural abnormalities of the cerebrum, Neuropediatrics 25 (1994) 68–72.

[15] S. M. Surkar, R. M. Hoffman, B. Davies, R. Harbourne, M. J. Kurz, Impaired anticipatory vision and visuomotor coordination affects action planning and execution in children with hemiplegic cerebral palsy, Research in Developmental Disabilities 80 (2018) 64–73.

[16] S. Saavedra, A. Joshi, M. Woollacott, P. van Donkelaar, Eye hand coordination in children with cerebral palsy, Experimental brain research 192 (2009) 155–165.

[17] J. Verrel, H. Bekkering, B. Steenbergen, Eye–hand coordination during manual object transport with the affected and less affected hand in adolescents with hemiparetic cerebral palsy, Experimental brain research 187 (2008) 107–116.

[18] B. Steenbergen, W. Hulstijn, A. De Vries, M. Berger, Bimanual movement coordination in spastic hemiparesis, Experimental Brain Research 110 (1996) 91–98.

[19] S. James, J. Ziviani, R. S. Ware, R. N. Boyd, Relationships between activities of daily living, upper limb function, and visual perception in children and adolescents with unilateral cerebral palsy, Developmental Medicine & Child Neurology 57 (2015) 852–857.

[20] Kinarm, Kinarm exoskeleton product webpage, 2022. URL: https://kinarm.com/kinarm-products/kinarm-exoskeleton-lab/, last accessed on 2022-07-03.

[21] Kinarm, Kinarm exoskeleton integrated gaze-tracking datasheet, 2022. URL: https://kinarm.com/download/kinarm-gaze-tracker/, last accessed on 2022-07-03.

[22] F. R. Danion, J. R. Flanagan, Different gaze strategies during eye versus hand tracking of a moving target, Scientific reports 8 (2018) 1–9.

[23] M. Q. Khan, S. Lee, Gaze and eye tracking: Techniques and applications in adas, Sensors 19 (2019) 5540.

[24] R. A. Naqvi, M. Arsalan, G. Batchuluun, H. S. Yoon, K. R. Park, Deep learning-based gaze detection system for automobile drivers using a nir camera sensor, Sensors 18 (2018) 456.

[25] A. Kar, P. Corcoran, A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms, IEEE Access 5 (2017) 16495–16519.

[26] D. D. Salvucci, J. H. Goldberg, Identifying fixations and saccades in eye-tracking protocols,

in: Proceedings of the 2000 symposium on Eye tracking research & applications, 2000, pp. 71–78.

[27] B. Cassin, S. Solomon, M. L. Rubin, Dictionary of eye terminology, Triad Publishing Company Gainesville, 1990.

[28] S. Tangnimitchok, A. Barreto, F. R. Ortega, N. D. Rishe, et al., Finding an efficient threshold for fixation detection in eye gaze tracking, in: International Conference on Human-Computer Interaction, Springer, 2016, pp. 93–103.

[29] Kinarm, Kinarm standard tests summary, 2022. URL: https://kinarm.com/kinarm-products/kinarm-standard-tests/, last accessed on 2022-07-03.

[30] C. Lowrey, C. Jackson, S. Bagg, S. Dukelow, S. Scott, A novel robotic task for assessing impairments in bimanual coordination post-stroke, International Journal of Physical Medicine and Rehabilitation S3:002 (2014). doi:10.4172/2329-9096.S3-002.

[31] K. Tyryshkin, A. M. Coderre, J. I. Glasgow, T. M. Herter, S. D. Bagg, S. P. Dukelow, S. H. Scott, A robotic object hitting task to quantify sensorimotor impairments in participants with stroke, Journal of neuroengineering and rehabilitation 11 (2014) 1–12.

[32] A. M. Coderre, A. Abou Zeid, S. P. Dukelow, M. J. Demmer, K. D. Moore, M. J. Demers, H. Bretzke, T. M. Herter, J. I. Glasgow, K. E. Norman, et al., Assessment of upper-limb sensorimotor function of subacute stroke patients using visually guided reaching, Neurorehabilitation and neural repair 24 (2010) 528–541.

[33] Kinarm, Dexterit-e., 2022. URL: https://kinarm.com/kinarm-products/dexterit-e/, last accessed on 2022-07-13.

[34] G. Gupta, Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter, International Journal of Soft Computing and Engineering (IJSCE) 1 (2011) 304–311.

[35] B. Justusson, Median filtering: Statistical properties, Two-Dimensional Digital Signal Prcessing II (1981) 161–196.

[36] TOELT, Github repository, 2022. URL: https://github.com/toelt-llc/KiPy, last accessed on 2022-08-20.