

## USER GUIDE

(Beta version 0.03 - 1 January 2021 – [www.ikd1d.org](http://www.ikd1d.org))

### Contents

Release notes.....	2
Installation .....	3
Process overview .....	3
Data input .....	3
Gravity correction .....	3
Trial selection -> Manual.....	4
Trial selection -> Auto .....	4
Curve fitting .....	4
Data output fit -> merge fitted files.....	5

Copyright (C) 2015 Jos Vanrenterghem, Mark A. Robinson, Ian Poole.

IKD1D is free but copyright protected software, distributed under the terms of the GNU General Public License as published by the Free Software Foundation (version 3, as below). Further details on "copyleft" can be found at <http://www.gnu.org/copyleft/>. In particular, IKD1D is supplied as is. No formal support or maintenance is provided or implied. IKD1D is currently developed and maintained by Jos Vanrenterghem, Mark A. Robinson, and Michiel Hagen.

Contact: [jos.vanrenterghem@kuleuven.be](mailto:jos.vanrenterghem@kuleuven.be)

## Release notes

### Version 0.01

Original version

### Version 0.02

- Error in gravity correction debugged.

Formula for gravity correction:  $T_g = g/\cos(a) * \cos(d)$

Tg = limb weight at angle d

d = current angle

g = limb weight measured at angle a

a = angle at which the limb weight is measured

Original code from version 0.1:

```
%Correct for gravity if required
if get(findobj('Tag','checkGravity'),'Value') == 0
    yTor= d(dIndex, TORQUECOL);
else
    g = str2num(get(findobj('Tag','editWeight'),'string'));
    angle = str2num(get(findobj('Tag','editAngle'),'string'));
    for i = 1: length(dIndex)
        Tg =(g/cos(angle))* d(dIndex(i),4);
        if d(dIndex(i), TORQUECOL) >= 0
            %Extension
            yTor = [yTor; d(dIndex(i), TORQUECOL) + Tg];
        else
            %Flexion
            yTor = [ yTor; d(dIndex(i), TORQUECOL) - Tg];
        end
    end
end
```

The following features were adapted in the new code:

- The angles are converted from radians to degrees.
- The value  $g/\cos(\text{angle})$  is multiplied by  $\cos(d)$ .
- The value of d is extracted from the angle values instead of the velocity values.
- For the flexion torque –which is a negative value- Tg is added to the original torque value.

New code in version 0.02:

```
%Correct for gravity if required
if get(findobj('Tag','checkGravity'),'Value') == 0
    yTor= d(dIndex, TORQUECOL);
else
    g = str2num(get(findobj('Tag','editWeight'),'string'));
    angle = str2num(get(findobj('Tag','editAngle'),'string'));
    for i = 1: length(dIndex)
        Tg =(g/cos(deg2rad(angle)))*
        cos(deg2rad(d(dIndex(i),3)));
        if d(dIndex(i), TORQUECOL) >= 0
            %Extension
            yTor = [yTor; d(dIndex(i), TORQUECOL) + Tg];
        else
            %Flexion
            yTor = [ yTor; d(dIndex(i), TORQUECOL) + Tg];
        end
    end
end
```

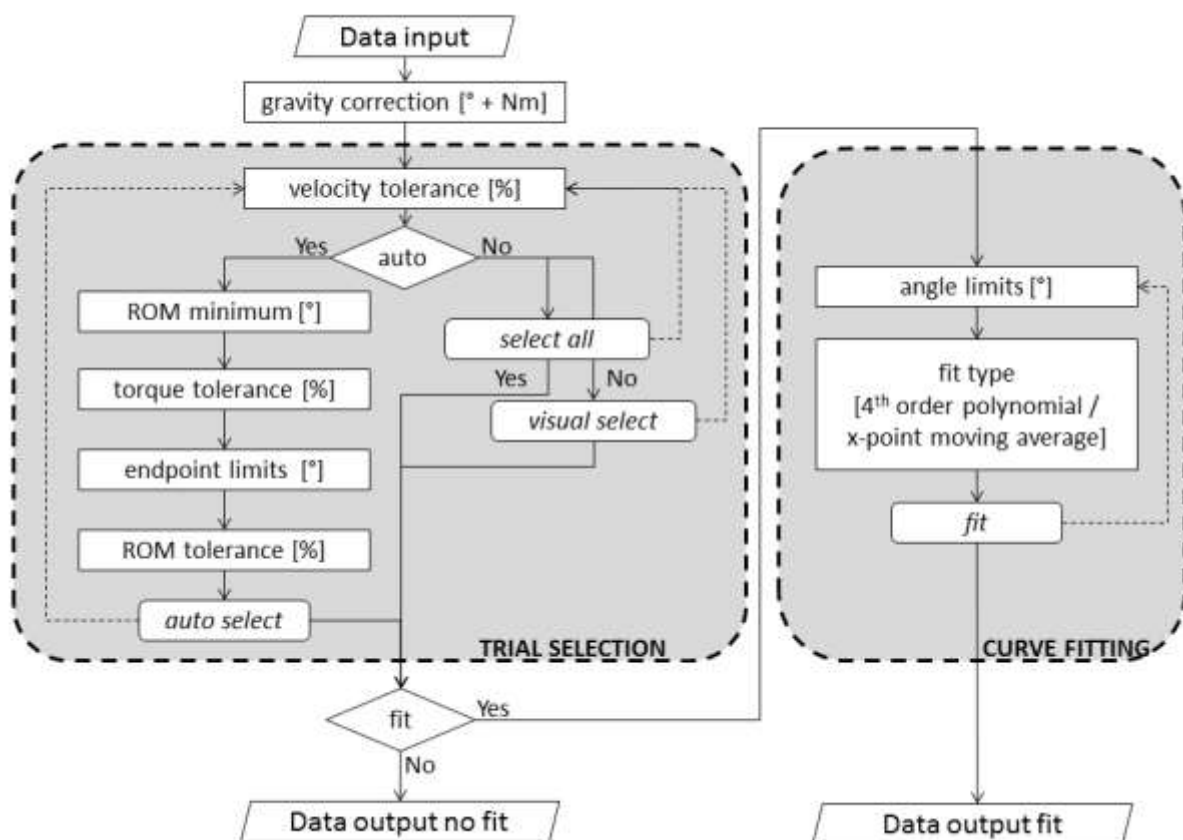
### Version 0.3

File import function modified to automatically recognize files from Biodex System 5. File format is different from earlier versions of the Biodex software.

## Installation

The application is a Matlab script existing of two parts (IKD1D.fig, IKD1D.m). It has been tested for Matlab R2013a. To run the application, extract the files from the zip folder, add the folder path in which the scripts are, and run the m-script.

## Process overview



## Data input

1. Section Input file: browse to folder with your data, load the exported file (currently Biodex and Cybex files can be uploaded)
2. Identify whether the file includes concentric or eccentric data, and torque-angle-velocity data should now be displayed for the file

## Gravity correction

1. Apply the gravity correction. This is necessary because raw export files are not gravity corrected yet.
2. Fill in the limb weight (in Nm) and the angle at which you measured the limb weight.

3. Note: Gravity correction needs to be done before automatic or manual selection. Otherwise the selection procedure will not account for it.

## **Trial selection -> Manual**

1. Check in the section 'velocity selection criteria' whether the application has correctly identified the desirable isokinetic velocities. You can also alter the tolerated velocity.
2. Importantly, it is always possible that one contraction effort is broken up into two parts (typically at start or end of the attempt). Once you have established your criteria, you can easily verify whether this is the case or not by counting the number of trials marked in red on the screen, and comparing that number to the number of regions as displayed in the section select peak regions. If the latter is higher, then it is worthwhile using Select Peaks to scroll through the different regions that are being selected, and accept only the ones that you wish to keep. It is for example also possible to remove a first and last trial systematically. You do this by using the Select Peaks button, and this will highlight each region and ask you whether you wish to include it or not.

## **Trial selection -> Auto**

1. You will hopefully be able to quite easily understand the criteria of automatic fitting. When data collection is done according to highly standardized procedures, then it should in theory be possible to do the entire process in an automated way. We have therefore attempted to automate the region selection and fitting based on standard criteria. The benefit of this automatic selection is that it becomes very straightforward to describe your trial selection and curve fitting processes.
2. The automatic selection will override anything you may have already done via manual fitting, and will basically try to achieve the same but now according to strict rules. After the automatic selection procedure however, it is possible to use the Select Peaks button and manually remove trials that were selected automatically.

## **Curve fitting**

3. Once you have decided on the trials to include, you can now press Fit Curves in the analysis section of manual fitting. If you think that there is unreliable data at the beginning of the end of a contraction, identified for example by high variability between trials, it is then possible to reduce the range of motion for the fitting at either ends by selecting min/max angle and pressing fit curves again. You can also select the preferred fit type, which is either a 4th order polynomial or a simple point by point average with predefined averaging window to generate a smoother curve (defined by number of smoothing points).
4. Press Save and this will save three csv files. One file will contain all data, another will contain the averaged curves, and another will contain the polynomial curve (even if you may not have selected this option). For now, we have decided to just export everything. You can open these files in Excel (importing as comma separated file may be needed) to see how data is structured.

## **Data output fit -> merge fitted files**

In order to apply statistical data comparison between averaged curves there is the option to 'merge fitted files'. This section allows you to select the output files that were saved (e.g. a number of subjects for one experimental condition) and will re-sample the fitted curves into one output file that aligns the data from all files for their angular range of motion. After selecting the files the application asks where to save the merged csv file. This merged file can be opened in excel (importing as comma separated file may be needed) and will help for preparing the data for population-based hypothesis testing, e.g. through using one-dimensional Statistical Parametric Mapping ([www.spm1d.org](http://www.spm1d.org)).