

User manual

Index1	1
1 The PManalyzer - Basic Theory	3
1.1 PCA on kinematic data	3
1.2 Comparing trials - the optimal basis for a group	4
1.3 Pre-implemented GUI - Variables	6
1.3.1 Descriptive measures - Mean, standard deviation	6
1.3.2 Mean Power/Signal activity (P/A)	6
1.3.3 Trial specific eigenvalues/activity ($rVAR_k$, $rSTD_k$, $rACT_k$)	6
1.3.4 Trial specific cumulative and residual eigenvalues/activity	7
1.3.5 Number of zero-crossings/peaks (NoZC, NoP)	8
1.3.6 Standard deviation of times between zero crossings/peaks (stdTZC/stdTP)	8
1.3.7 Zeros to peaks ratio (RZCP)	8
2 The PManalyzer GUI - General information	8
2.1 The interface	8
2.2 Code structure and computation	9
3 Using the GUI	10
3.1 Preparation of Input data	10
3.2 Interface options	11
3.2.1 First main panel - "Input data"	11
3.2.2 Second main panel - "Computation and output"	13
3.2.3 Third main panel - "Plots"	16
3.2.4 Fourth main panel - "Videos"	17
3.2.5 Fifth main panel - "SAVE/LOAD SETTINGS"	19
3.3 Output data	20
3.3.1 Folder - Computed data	20
3.3.2 Folder - Plots	20
3.3.3 Folder - Videos	20
3.4 Customization	20
3.4.1 Gap filling	21
3.4.2 General pre-processing	21
3.4.3 Coordinate transformations	21
3.4.4 Normalizations	21
3.4.5 Weights	22
3.4.6 Computed variables	22
3.4.7 Coloring options	23
3.4.8 Personalized plots	23
3.4.9 Enable Extras	23
3.4.10 Core - functions	23
4 Further code details	24
4.1 Interface self-regulation	24
4.2 Fail safes	24
4.3 Common errors	24

1 The PManalyzer - Basic Theory

This software is a tool, for beginners and experts, that utilizes principal component analysis PCA to extract correlated whole-body movements out of kinematic data called principal movements PM. The advantage of reducing the complexity of human motion has been discussed extensively in literature [1–8]. The PManalyzer code was originally based on the analysis of kinematic data with the purpose of clinical research [9], and was then adapted to enable generalized applications. The main purpose of this user’s manual is to explain the software’s features in detail. Nevertheless, the basic theoretical concepts will be summarized in the following subsection.

1.1 PCA on kinematic data

In the context of kinematic measurements, typical data consists of the 3D positions in time of anatomical landmarks in Cartesian coordinates (x, y, z of each of the “n” markers). A kinematic data set can be written in TxN matrix form in which the $N = 3n$ columns represent the time-series of the respective x-y-z - coordinates of each of the n markers. Each of the T rows then contains the measured 3D positions of all markers at one time-point.

Mathematically, each column is attributed to one “basis-vector”. All of the N basis vectors together form an orthonormal basis ONB, which is the NxN unity matrix, i.e. each basis vector refers to one column, specifically, to one coordinate. Applying a PCA to such a data matrix is motivated by the idea of finding a new ONB, denoted by PC_k , that describes most of the variance in the data, using as few basis vectors as possible. Therefore, each of the N new basis vectors PC_k ($1 \leq k \leq N$) of the new ONB will be a linear combination of the original basis vectors [10]. In other words, each PC_k attributed to a combined variation of various correlated coordinates, i.e. it is not attributed to the movement of a single marker coordinate, but to one specific linear movement of every marker, that happens simultaneously. Therefore, each PC_k describes one linear motion pattern, called principal movement PM_k . The space spanned by PC_k is called posture space. A point in posture space is a specific postural configuration, or position that the body can be in.

The information about the specific manifestation in time of each PM is quantified by the TxN score-matrix S. The manifestation of each linear movement pattern PM over time will be determined by one respective column of the score time-series, i.e. each column of S will determine the evolution

of the respective movement pattern. Each position that the body is in (=postural configuration), is a position in posture space obtained by computing the PCA. Therefore it makes sense to refer to the k-th column as principal position $PP_k(t)$. Besides from the new basis vectors PC_k and the principal positions $PP_k(t)$, applying a PCA yields further results, the EV_k . Each PM_k has a respective EV_k quantifying the amount of variance that can be explained with the linear movement pattern PM .

Note: Literature often refers to the PC_k simply as eigenvectors or eigenmodes. The $PP_k(t)$ are sometimes simply referred to as scores or denoted by $\xi_k(t)$. We hope that the name conventions [3,4,9] used in this user's manual can be agreed upon for future consistent usage in literature.

Kinematics in posture space and variables of postural control

In the context of kinematic PCA-applications the $PP_k(t)$ can be differentiated to obtain principal velocities PV_k and principal accelerations PA_k . Any postural reconfiguration can be ascribed to acting forces, hence the PA_k are directly linked to postural reconfigurations. Besides from external forces, such as gravity - which is constant - acting muscle forces are responsible for such reconfigurations. Therefore, the PA_k reflect the neuromuscular control of the respective PM_k . It has been shown that the PA s can be associated with center of pressure trajectories [3,5] or gain information about the neuromuscular control system [7,9].

Note: Despite noise amplification due to twice differentiation, it has been shown that variables computed on the PA s may be validated and show group effects [9]. Supporting conventional time-series filtering techniques, a PCA automatically works as a noise filter, typically representing noise in higher components.

1.2 Comparing trials - the optimal basis for a group

The three quantities PC_k , EV_k and $PP_k(t)$ contain all information about the input data in terms of movement patterns. Different trials can then be compared in several different ways. Some of the first applications (**CITE**) computed a new basis for every trial and compared the amounts of contribution of each individual PM (EV spectrum). The advantage is that each trial is represented in a trial specific coordinate system that represents the respective trial. The disadvantage is that the comparison of trials

with respect to the $PP_k^{trial}(t)$ is not possible, since the PP_k are projections to the individual bases. To make the PP comparable it has been proposed to compute a basis PC_k that represents the analyzed group as a whole in the best way possible and compute the $PP_k^{trial}(t)$ as projections on the same basis.

When comparing kinematic data of various trials both the positioning in space and anthropometrical differences distort the results. To reduce such differences, the data of each subject can be normalized [3,4,11]. The first step is to center the data, by subtracting the mean of each individual time-series (each column) to the respective time-series, reducing differences between data sets that occur due to anthropometric or positioning differences. Furthermore, anthropometric differences do not only influence the mean position in space, but also the amount of movement produced by a participant. To compare individual subjects it makes sense to normalize each data-set according to application specific criteria. Normalization factors such as the mean Euclidean distance MED [4,11] or the body height [3,9] have been proposed to ensure that all subjects contribute equally to the overall variance or that anthropometric influences are reduced, respectively. Both techniques normalize each trial's data by dividing it through a scalar normalization factor. Future applications might benefit from normalizations that consider individual segment lengths, yet such normalizations and the effects of differing normalization techniques on the resulting variables have yet to be discussed in literature, a task that can benefit from this software.

Another anthropometric influence that can be considered is a participant's weight distribution. As an example, when moving a hand, less mass has to be accelerated and controlled, in comparison to moving the thigh. Therefore it can be argued that the movement of each marker time-series should be scaled according to the weight that the respective marker represents [3,4]. This shall allow all markers to contribute equally to the overall variance, regardless of the mass that has to be moved. The concept of weighting the data has already been applied in literature [5,7,9]. However, the effects of weighting on the results and the comparison of weighting choices have not been explored.

1.3 Pre-implemented GUI - Variables

The three PCA time-series ($PP_k(t)$, $PV_k(t)$, $PA_k(t)$) can be used to run any sort of time-series analysis that fits the research question. Here, the pre-implemented variables that the GUI can compute are described mathematically for computation and saving options). Note that all selected variables are computed on all three time-series.

1.3.1 Variable - Descriptive measures (Mean, standard deviation)

For each of the three time-series (PP, PV, PA) both the mean and the standard deviation are computed. Additionally, both descriptive measures are computed separately on the positive and the negative part of each time-series, respectively.

1.3.2 Variable - Mean Power/Signal activity (\bar{P}/A)

A common way of describing the overall power in a signal is the RMS. All steps of the computation are saved as separate results. The instantaneous power $P_i(t)$ is a time-series with the squared values of the respective time-series. The average power \bar{P} is the mean of the instantaneous power. Finally, the RMS is the square-root of the average power and scales the result back to the original scale.

The term “power” is accurate for electrical voltage signals since in this case specifically the power is proportional to the voltage squared. However, squaring terms will emphasize higher and downscale smaller deviations. This distortion might be involuntary for kinematic movements where every position should be weighed identically. For such cases the average activity \bar{A} is computed by computing the mean of the instantaneous activity $A_i(t)$ (absolute values of the respective time-series). Note that the average activity scales equally to the RMS (not the average power \bar{P}).

1.3.3 Variable - Trial specific eigenvalues/activity ($rVAR_k$, $rSTD_k$, $rACT_k$)

The sum of all variances of each trial's individual $PP_k(t)$ time-series

$$totVAR^{trial} := \sum_k VAR_k^{trial} := \sum_k VAR(PP_k^{trial}(t))$$

can be interpreted as the total amount of movement performed in the respective trial. Similar to the EV_k , the VAR_k^{trial} can be interpreted as trial specific eigenvalues that quantify the contribution of each $PP_k^{trial}(t)$ to the total amount of movement in the respective trial. The subject specific relative variance defined by

$$rVAR_k = \frac{VAR_k^{trial}}{totVAR^{trial}} \cdot 100$$

is therefore the percentage of the trial's overall variance that can be explained with the respective $PP_k^{trial}(t)$ [4]. To obtain variables that scale as the original movement the variance in the $rVAR_k$ computation can be replaced by the standard deviation to compute trial specific relative standard deviations $rSTD_k$ [3,9] or by the activity (mean absolute deviation) to compute trial specific relative activities $rACT$. Such variables have been used to analyze the structure of a movement [3,9], i.e. to quantify the respective amount of relative contribution of the interacting principal movements.

Analogously, $rACT$, $rSTD$ and $rVAR$ can be computed using PV and PA time-series. However, such variables have not yet been discussed in literature.

1.3.4 Variable - Trial specific cumulative and residual eigenvalues/activity

In applications where the dimensionality of a movement is of interest, it makes sense to define cumulative eigenvalues as

$$CUM_rVAR_k := \frac{\sum_{1 \leq n \leq k} VAR_n^{trial}}{totVAR^{trial}} \cdot 100,$$

or analogously CUM_rACT_k , CUM_rSTD_k , which explain the cumulative contribution of the respective component order. This can be used, as an example, to compute residual variances

$$RV := 100 - CUM_rVAR_n = \frac{\sum_{k > n} VAR_k^{trial}}{totVAR^{trial}} \cdot 100$$

with n as the limit of the included PC-order [6,11]. The higher the residual variance RV , the less variance can be explained with the included PCs and the more meaningful patterns of movement were utilized while performing a movement, i.e. the higher the dimensionality of the movement [6]. With a similar line of argument, low values of $rSTD_k$ in highly contributing low-order components have also been used as arguments for higher dimensional movements or more complex movements in the sense of neuro-mechanical degrees of freedom [3].

Analogously, CUM_rACT , CUM_rSTD and CUM_rVAR can be computed using PV and PA time-series. However, such variables have not yet been discussed in literature.

1.3.5 Variable - Number of zero-crossings/peaks (NoZC, NoP)

A zero-crossing happens when a time-series passes the zero line and changes sign. A peak is either a local maximum or local minimum of a time-series. The number of zero-crossings/peaks of a time-series may be an indicator of change that occurs in the position/velocity/acceleration of a PM. Both variables are influenced by noise and should rather be used to obtain qualitative, not as exact quantitative, information. The number of zero-crossings of the PA has been validated and used in two studies to quantify the activity of the neuromuscular control system [7,9].

1.3.6 Variable - Standard deviation of times between zero crossings/peaks (stdTZC/stdTP)

The standard deviation of the time between zero-crossings/peaks is a measure designed to measure the variability of the timing of the changes that occur in a time series described by another variable such as number of zero-crossings/peaks. The standard deviation of time between zero-crossings of the PA has been used in two studies to quantify the variability in the activity of the neuromuscular control system [7,9].

1.3.7 Variable - Zeros to peaks ratio (RZCP)

The ratio between number of zero-crossings and peaks was originally designed to quantify the amount of drastic changes that occur in a time-series (zero-crossings) in comparison to any changes observable.

2 The PMAalyzer GUI - General information

2.1 The interface

As depicted in Fig. 1, the PMAalyzer interface is organized into five main panels with respective title in red font for each of the self-explanatory parameter selections: 1. “Input data”, 2. “Computation and output”, 3. “Plots”, 4. “Videos” and 5. “SAVE/LOAD SETTINGS”. Each main panel is highlighted with red margin and the respective main options are highlighted with red font color. Furthermore, the subpanels with their titles allow the interface layout a more intuitive structure. Following the

subpanels one by one allows the user to follow the conventional steps for a PCA applied to kinematic data as described in the methods section.

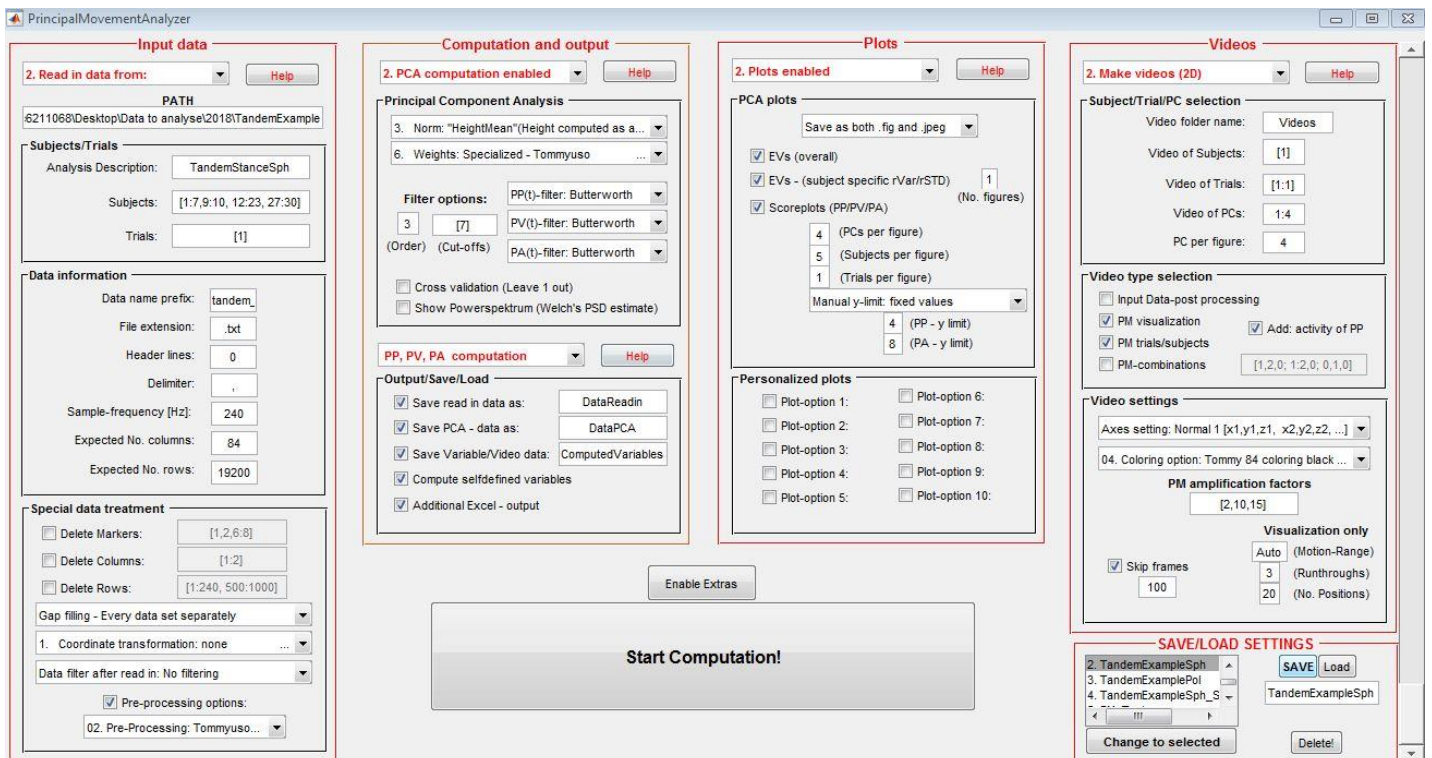


Fig. 1. User interface of the PMA analyzer software. The selected features are an example input that reads in data, computed all PM related variables and saves the data, but has plot and video options disabled.

Note: First, the interface was created with the “guide” tool in “Matlab 2015a” in “Windows 10” on a screen with a 1920x1080 resolution. Both “Units” and “FontUnits” were set to “normalized” with respect to screen size. Nevertheless, we have encountered scaling issues on smaller screens and Mac books that we have not been able to solve yet. Also plotting features that are non-existent in previous Matlab versions can lead to unexpected errors.

2.2 Code structure and computation

The computational steps performed by the source code comply with the structure of the kinematic PCA described in the previous section. First, the data is read in, pre-processed and submitted to the PCA (the code used SVD), according to the selection of the user. Then variables are computed and chosen visualizations (plots and videos), if selected. To be able to follow the software’s computational steps a descriptive text containing information about the current computational step is printed in the output-console. More involved users can explore the main function “PrincipalMovementAnalyzer.m”,

which contains all the create- and callback-functions of the respective interface buttons. Helpful comments are provided for source code variables and computational steps that are important for the code's understanding.

Functions

When hitting the “Start Computation!”-button, the function “pushbutStartComputation_Callback” is called, executing the identification of the selected input parameter and computational options. Based on the computational steps described in the theoretical section, there are five main computational parts highlighted in the source code of this function: “Part 1: Input/Read-in”, “Part 2: PCA computation”, “Part 3: Variables computation”, “Part 4: Plots” and “Part 5: Videos”. In each of the five parts, most computations and output tasks are outsourced to other functions to maintain a better overview of the code structure. Functions that fulfill outsourcing jobs and are necessary for the conventional computations are contained in the folder “CoreFunctions”, organized in subfolders that contain the description of the related sections (“1. Readin”, “2. PCA”, “3. Variables Extraction” and “4. OutputVidsandExcel”). The folder “ExtraFunctions” is meant to contain any non-essential functions created by users for specific extra computations. Functions containing computational options that the user might want to edit are contained in the folder “FunctionsToEdit” (pre-processing-, normalization-, coloring-, plot-options, etc.). When starting, the GUI loads the functions of all three folders (and any subfolders) automatically enabling the usage of all contained functions. Furthermore, the code is documented by comments to identify the task of each code section and help identify important computational variables and their respective role in the code.

3 Using the GUI

3.1 Preparation of Input data

PREFIX: Quick description of the data. Prefixes are optional and can be left out.

Numeration: s*t* identifies subject and trial numbers.

File extension: Must be compatible with Matlab function “importdata()”.

(Example: bal_s4t16.txt stands for the “.txt” file of subject 4, trial 16 with prefix description “bal_”).

Note: Beforehand, the kinematic data must be named and sorted according to the example (“PREFIXs*t*.fileExtension”). Spaces are allowed between the prefix and the rest of the name but must be consistent throughout the data! The data should be an ASCII file, sorted in the form: Rows = Time, Columns = markers.

3.2 Interface options

The interface consists of five main panels and two buttons (“Start Computation!” - button and “Enable Extras”).

3.2.1 Main panel 1 - “Input data”

The first main panel is comprised of a popup-menu with a help button, an edit box to specify the folder path that contains the input data and three subpanels. The popup-menu allows to enable reading or loading data, or to disable all read in options. The help button will display this information to a user.

3.2.1.1 Subpanel - “Subjects/Trials”

Analysis Description: All data related to an analysis will be saved in a subfolder of the folder that contains the data. The string specified under “Analysis description” will be included in the name of the subfolder enabling a user to easily save different computations under different names. The interface input to specify trials and subjects is of standard Matlab row-vector form.

Subjects/Trials: The subjects/trials that shall be read in and computed with must be specified in this edit box. Numbers shall be comma separated. The code also recognizes the Matlab colon “:” operator.

Example: [1, 4:7, 10:2:14, 18] will read in subjects/trials 1, 4, 5, 6, 7, 10, 12, 14 and 18.

3.2.1.2 Subpanel - “Data information”

Data name prefix: The purpose of the prefix is to identify the data. Any **prefix** must be specified in the respective interface field and will be cut out for computation. Prefixes may contain spaces.

File extension: Common file types with extensions such as .csv and .txt that are compatible with the Matlab function “importdata()” may be used.

Header lines: The number of header lines specified will be cut out for the analysis.

Delimiter: Common delimiters include space “ ”, comma “,” or tab “\t” and must be specified accordingly.

Sample-frequency: The sample frequency of the input data must be specified as a scalar value.

Expected No. columns/rows: The expected number of columns/rows of the trials data should be specified as they allow the user to monitor the data in a better way. Specifically, pre-processing the data or when cutting out rows or columns this can be useful, as the software will recognize discrepancies between the expected numbers and the actual data and alert the user if needed.

3.2.1.3 Subpanel - “Special data treatment”

Delete Columns/Rows/Markers: To delete markers (x,y,z), columns or rows the respective checkboxes must be activated and the numbers specified according to standard Matlab row-vector input. “Columns” are removed before “Markers”. Typically, the “remove column” option is used to remove the first two columns containing time-frames and numbering. Numbers shall be comma separated. The code also recognizes the Matlab colon “:” operator.

Gap-filling popup menu: The PMAalyzer uses a code based on PCA that will preserve the structure of the data [12]. To fill all gaps, the data is not allowed to have any empty columns or rows. If empty columns/rows are detected, the software will alert the user and stop the computation. This PCA based gap-filling can be used by applying it to each of the data sets separately, to the entire data set concatenated or to all trials of one subject concatenated.

Coordinate transformation popup menu: Default coordinates are considered to be Cartesian. Available coordinate transformations are to spherical and polar coordinates in 3D. **Note:** Transformation back to Cartesian coordinates is only done for the video options (in functions

“movie_selfmade.m” and “movie_selfmade_pca.m”). Saved input data (“DataReadin.mat”, “DataReadin_additional.mat”) are in the selected coordinate transformation basis and the PCs are combinations of the respective coordinates, i.e. the PM time-series (PP, PV, PA) describe linear movements in the transformed coordinate system.

Data filter after read in: Any sort of input data can be filtered with a Butterworth filter directly after readin, gapfilling and coordinate transformation. Any user defined data pre-processing can be added in the code and selected by enabling the respective checkbox and specifying the recognition string.

Pre-processing options: To select pre-processing options the checkbox must be activated, enabling the selection of specified pre-processing steps. One pre-processing step can be specified before the coordinate transformations and gap filling, and another can be specified after. Any additionally produced data can be saved. See section “3.4.2 Customize General pre-processing” for implementation.

3.2.2 Main panel 2 - “Computation and output”

The second main panel is comprised of two popup-menus, each with a respective help button a respective subpanel. The popup-menus enable computation or loading data, or disable the respective subpanel with the computation options. The help button will display information about the enabled/disabled section to a user.

3.2.2.1 Subpanel - “Principal Component Analysis”

Normalization popup menu: Per default, no normalization is selected. The data can either not be normalized or normalized according to six pre-implemented options:

1. No normalization: Data is not normalized.

2. MED (Mean Euclidean distance): The mean Euclidean distance is calculated for the data set.

3. SpecifyHeight: User can specify bottom and top markers and has the option to compute the mean height over all time-frames or to specify time-frame at which the height shall be computed (Distance between the z-coordinates of the specified markers). **Default:** Mean over all time-frames of markers 26 (bottom) and 1 (top).

4.AutoHeight: The code detects the highest and lowest marker (z-axis) at the first (definable in code) timeframe and classifies them as top and bottom markers, respectively. The user has the same options as with specified height normalization (mean or at specifies time-frame). **Default:** Height determined through top and bottom markers at time-frame 1.

5.-7. MAXRange1-3: This option determines the maximum range spanned by any two markers at any time-point (1-3 symbolize the coordinate number in which the range shall be determined). The maximum range is then the normalization factor.

(See section “3.4.4 Customize: Normalizations” on how to customize options.)

Weights popup menu: Per default, no weight option is selected. The user can use a pre-defined weight model for the standard 39 and standard 37 plug-in gait or added customized weight options (see section “3.4.5 Customize: Weights”).

Filter options/Powerspectrum: A PSD-estimate (Welch’s power spectral density estimate) of the first trial of the first subject can be computed to verify if the chosen cut-off frequency is plausible. The user is not restricted to one cut-off frequency. If a vector of cut-off frequencies is inputted the entire GUI selection (read in, computation, plots, videos, etc.) will be run with each cut-off frequency once! The name of the folder containing the results automatically contains the cut-off frequency, resulting in as equally many results folders as entries in a vector. **NOTE:** For frequency comparisons it is usually worth only saving important results, since the computation can generate vast amounts of data.

Cross validation: To assure that the posture space spanned by the PC_k is adequate to describe the input data, a “leave-one-out cross-validation” can be performed. Therefore, the PC_k -basis is computed for the data several times, each time missing one trial. Comparing the angles of the newly obtained basis vectors to the original PC_k -basis allows quantifying the quality of representation. This can be used as an inclusion criterion for the number of PCs used, since only PCs that describe the group as a whole make sense [3,9].

3.2.2.2 Subpanel - “Output/Save/Load”

All computed data will be saved in the subfolder “Computed data” in selected. The excel file containing the variables can also be found in this subfolder. Names should not contain any special characters, “.” or spaces.

Save read in data: When selected, the input data after pre-processing and any additionally computed data during pre-processing will be saved in the folder. The read in input data will be named “*CO__DataReadin.mat”, the additionally computed data will be named “*CO__DataReadin_additional.mat”. The “*” will be the used cut-off frequency that was used for the analysis.

Save PCA and variable/video data: The PCA data, if saved, will be sorted in three Matlab files. One that contains the time-series (PP, PV, PA), another containing the basic PCA data (EV, PC_k, PP) and the third one containing additional PCA data (mean-, norm-, size- and weights-inverse matrices, and leave one out cross-validation). Information about the used cut-off frequency and the available PMs will be included in the filenames, to (“*CO_*PC__DataPCA_TimeSeries.mat”, “*CO_*PC__DataPCA_Basic.mat”, “*CO_*PC__DataPCA_Expanded.mat”).

Compute self-defined variables: As default, the eigenvalues (EV and rEV), relative eigenvalues (rVAR, rSTD, CUM_rVAR, CUM_rSTD) and the time-series (PP, PV, PA) are always computed. However, the user must select the “Compute selfdefined variables” checkbox to enable the computation of additional variables specified in function “optionsVariablesComp.m” contained in the PMAalyzer subfolder “FunctionsToEdit”.

Additional Excel - output: All computed results of the variables with scalar values are automatically exported to an excel-file, if the checkbox “Additional Excel output” is activated. The software will simply neglect variables with non-scalar values. The file will be saved in the same subfolder as the

data saved throughout the GUI computation (subfolder “Computed data” of the folder containing the analyzed data). The filename will display the data information specified in “Analysis Description” in the user interface as well as the filtering frequency and available PM information.

3.2.3 Main panel 3 - “Plots”

The third main panel is comprised of one popup-menu with a respective help button and two respective subpanels. The popup-menu enables/disabled all plotting options, which are further specified in the subpanels. The help button will display information about the enabled/disabled section to a user.

3.2.3.1 Subpanel - “PCA plots”

EVs (overall): Enabling this checkbox will plot the relative eigenvalues rEV and the cumulative relative eigenvalues. As default 20 eigenvalues are plotted, or less, if only less are available.

EVs (subject specific rVAR/rSTD): Enabling this checkbox will generate rVAR, rSTD, CUM_rVAR, CUM_rSTD plots. As default one figure is made containing one trial of all subjects.

Scoreplots: The user can plot the time-series (PP, PV, PA). These plots are commonly helpful to develop analysis strategies and are therefore default in this version of the software. The limits in the time-series plots can be left automatic or customized.

Note: EV and scoreplots must be saved either as .fig, .jpeg or as both.

3.2.3.2 Subpanel - “Personalized plots”

Any types of plots can be generated and automatically displayed in this subpanel, using the function “personalizedPlots.m”.

3.2.4 Main panel 4 - “Videos”

The fourth main panel is comprised of one popup-menu with a respective help button and three respective subpanels. The popup-menu disables all video options, or enables 2D or 3D video options, that are further specified in the subpanels. The help button will display information about the enabled/disabled section to a user.

3.2.4.1 Subpanel - “Subject/Trial/PC selection”

Video folder name: The name of the folder containing the videos can be specified in this field. Avoid special characters. Spaces are possible.

Video of subjects/trials/PCs: The subjects/trials/PCs of which a video shall be made of must be specified in this edit box. Numbers shall be comma separated. The code also recognizes the Matlab colon “:” operator.

PC per figure: Specifies the number of PMs per video file. The input should be one integer value.

3.2.4.2 Subpanel - “Video type selection”

Once the respective trials, subjects and PMs are selected, the type of video that shall be made for each choice must also be selected as follows.

Input Data-post processing: Video of the pre-processed input data.

PM visualization: Video visualizing the PM itself (not the performance of a specific trial). Therefore the mean position of the first available trial is used and a PM motion is added. The motion will oscillate between the minimal to the maximal amplitude of the same trial.

PM trials/subjects: These videos will display the performance of the selected trials/subjects with respect to the selected PMs.

PM-combinations: This allows to compute PM video that include combinations of specified PMs added together. Therefore, matlab similar syntax must be used. Each movement may contain any amount of available PMs. Integer numbers separated by comma “,” or colon operator “:” will specify one combined movement. A semicolon “;” will allow to specify a new combined movement (again determined by comma or colon separated integer values). The amount of PMs per combined movement must not coincide. The software is able to generate a usable Matlab-matrix.

Add activity of PP: This option will additionally display the activity of the PP_k for the PM videos of any trial/subject. The combined activation of all PP_k is compared between all time-points for each trial/subject. Then the activity of the first twenty (if available) PP_k will be represented as a percentage of this maximum activation for all time-points.

3.2.4.3 Subpanel - “Video settings”

Axes settings: Any x-y-z coordinate axes setting is available. The correct one must be selected to ensure correct sagittal and frontal views for 2D videos.

Coloring options: Per default all markers of the data are plotted with black color without connecting lines. Customized coloring options can easily be specified (see section “3.4.7 Coloring options”). such options are then available for future usage.

Skip frames: When activating the “Skip frames” checkbox the integer will define the number of time frames that shall be skipped when plotting the figures for the video computation.

PM amplification factors: The amplification can be selected in various ways. One number will amplify all PMs with the respective number. Two numbers will amplify the first half of the PMs selected for video with the first number, the second half with the second, respectively. Analogously more numbers will split the selected PMs accordingly, until finally one amplification factor per selected PM will amplify each PM as chosen.

Visualization only options: The “Motion-Range”, “Runthroughs” and “No. Positions” customize the PM visualization videos specifically (not the trial/subject PM videos or the PM combinations). A range set to “Auto” will use the first available trial to find the maximum, minimum PM amplitude for the visualization. Other options must be specified in the code. Each run through will then enable one oscillation from minimum to maximum amplitude and back again. The number of positions will determine the amount of different positions distinguishable during each oscillation.

3.2.5 Main panel 5 - “SAVE/LOAD SETTINGS”

The fifth main panel is comprised of a list-box that contains previously saved GUI settings, four push-buttons and an edit-box.

SAVE: Hitting the save button will save the current GUI settings in the Matlab file “PManalyzer_settings_struct.mat” contained in the folder that contains the input data, using the name specified in the edit box. If the file does not exist, it will be created. **Note:** No spaces are allowed as names. Furthermore, names contain automatic enumeration ordered according to the time of the creation.

Load: If a “PManalyzer_settings_struct.mat” is available in the folder containing the input data, the settings can be loaded. Otherwise, the file is created and a notification appears.

Change to selected: Hitting this pushbutton changes the interface selection according to selection. If not saved, the previous setting will be lost.

Delete: The “Delete!” button will delete the option selected in the list-box and update the enumeration.

3.3 GUI output and output data

All outputs will be saved in the subfolder of the folder containing the data, which is created automatically when starting the computation. The subfolder will contain further subfolders that separate the data.

3.3.1 Subfolder - Computed data

This folder contains all forms of saved data (pre-processed read in and additional data, computed variables, PCA related parameters and time-series and the excel output). If needed, the PMAnalyzer supports saving “problematic” data, i.e. if the according code in the function “readin_PMAnalyzer.m” is activated manually, data that was gap-filled or did not have the expected row-column format will be saved in an additional subfolder (“Gap filled data” or “Suspect data”). These two options were disabled on purpose, since they might generate large amounts of unnecessary data.

3.3.2 Subfolder - Plots

This subfolder contains all PCA plots described in section 3.2.3.1 Subpanel - “PCA plots”, organized in two subfolders: “EV plots” and “Scoreplots (PP, PV, PA)”. The local variable “persFigDir” in function “personalizedPlots.m” contains a string with the path to another subfolder named “Personalized plots”, enabling the user to follow the saving structure and save personalized plots in the respective directory (see section “3.4.8 Customize: Personalized plots” on how to create personalized plots).

3.3.3 Subfolder - Videos

The default name of this folder when starting the GUI is “Videos”. The user is free to change this name in the edit box “Video folder name”. Furthermore, the computed videos are organized in subfolders (“Original motion”, “PM subject visualization”, “PM visualization”, “PM combinations”) of this subfolder. Note: Subfolders are only created if the video type is selected for computation.

3.4 Customization

All functions designed to customize options, with the exception of the function “optionsVariablesComp.m”, are contained in the folder “FunctionsToEdit” and follow the same

structure when being edited: First the name of the new option must be added, by adding a new line to the character array “available_*_options_names” following the examples in each function (“*” is a placeholder for). When restarting the user interface the added option will be loaded into the GUI automatically with the other options, making the new option available for selection.

3.4.1 Customize: Gap filling

Not available yet

3.4.2 Customize: General pre-processing

Pre-processing can be customized in the function “optionsPreProcessing.m”, at three different stages of the data read in. The first pre-processing can be performed right after reading in a trial and cutting out specified columns, markers and rows (before Gap filling). The second pre-processing step is performed right after gap-filling the data (before coordinate transformations). The third one is performed after transforming the data and filtering (if any of the two respective options are chosen).

When computing results that shall be saved as additional data, the variable “Data_additional_output” in function “optionsPreProcessing.m” must contain this information and will be saved in the struct “Data_Readin_TrialSubj_additional(Trial, Subj).FromPreProc_*”. (“*” will represent the pre-processing step 1, 2 or 3).

3.4.3 Customize: Coordinate transformations

Coordinate transformations can be customized in the function “optionsTransformData.m”. Default coordinates of input data are considered to be Cartesian. The PMAalyzer can transform data in Cartesian coordinates to spherical or cylindrical coordinates (and back to Cartesian for the video options).

For PCA applications in which data is not in Cartesian coordinates or shall be transformed differently, the first step is to define a transformation function from the input coordinates to the computation coordinates and another function that transforms data in computation coordinates to Cartesian (for the video options), following the two example functions at the end of the function “optionsTransformData.m”. The second step is to enable the transformation option that specifies a handle to these two functions (four such additional transformations have already been pre-

implemented for users, but disabled in the function). The third and last step is to define the name in the “available_transformation_option_names” character array. The GUI will then load the name the next time when it is started.

3.4.4 Customize: Normalizations

The user can include normalization options by following the examples at the bottom of the function and including the name of the normalization option in the “available_normalization_option_names” character array.

3.4.5 Customize: Weights

Adding a weight option consists of adding a name for the weight procedure to the gui in the character array “available_weight_option_names” in the function “optionsWeights.m”. Then, following the example of the pre-defined weight options a weight vector must be specified that coincides with the read in data.

3.4.6 Customize: Computed variables

Variables specified in the function “optionsVariablesComp.m” (see section “1.3 Pre-implemented GUI - Variables” for already available variable computations) will be automatically computed on all three time-series (PP, PV, PA). The user can manually disable computations of specific variables by changing their respective “allComputeVarsYesOrNo”-value from 1 to 0, or disable the computation on one of the three time-series by setting “compP*Vars = 0” (* = P, V or A).

To add any kind of variables computed on the time-series the four necessary variable options must be specified (name, unit, function handle, computation decision) by following the instructions of the template “Add variable template” in the function “optionsVariablesComp.m”, or by adding the four items to the list following the example of the pre-implemented variables. The function corresponding to the function handle must be specified according to the examples, at the bottom of “optionsVariablesComp.m” .

NOTE: The names and units are used explicitly for the excel-output. Once specified, all scalar valued variables are automatically included in the excel-output. Other variables are ignored by the excel-

output but nevertheless saved with the other variables in the file “*CO_*PC___ComputedVariables.mat” (see 3.2.2.2 Subpanel - “Output/Save/Load” part “Compute self-defined variables”).

3.4.7 Customize: Coloring options

After specifying the name of the new coloring in the “available_coloring_option_names” character array the user can follow the example of pre-implemented coloring options, specifying as many “plot lines” as desired with the four necessary items per line (sorted markers for line, symbol with connection line of the line, color of the marker faces of the line and marker sizes). The template at the bottom of the function “optionsColoringVids.m” can be used.

3.4.8 Customize: Personalized plots

Any name changed in the character array “available_peronalizedPlot_option_names” in function “personalizedPlots.m” will be displayed in the according checkbox of the subpanel “Personalized plots” when the GUI is restarted. When the checkbox is activated, the specified computation in the according if statement “if get(handles.chbPlotPersonalized*, 'Value') == 1” (with “*” determining the respective checkbox number). If needed, all GUI handles are available and are self-explanatory or explained in the code.

3.4.9 Customize: Enable Extras

The “Enable Extras” button is meant as a tool for expert MATLAB users to synchronize the PMAAnalyser to further computations or to other applications. The callback function activated when hitting the button is “pushbutExtras_Callback()” in the main function “PrincipalMovementAnalyzer.m”.

3.4.10 Customize: Core - functions

The core functions are not meant to be customized. However, expert users may freely experiment with the GUI as they please, as long as they use the GUI for research and **do not use the GUI for commercial purpose** and with it violate the license agreement (see **license agreement**). Bugs or comments may be forwarded to the main developer of the software Thomas Haid (thomas.haid@uibk.ac.at).

4 Further code details

4.1 Interface self-regulation

aaaaaa To improve efficiency when repeating calculation steps, data can be saved and loaded. The compatibility of the computing vs. loading vs. disabling options is also regulated over the interface, to avoid the selection of incompatible features.

4.2 Fail safes

Despite the self-regulating interface, it is still possible to select options that do not match the data. For a number of cases the code has implemented fail-safes, that are able to identify obvious selection errors, for example, when data has less columns than necessary for the selected normalization, weighting or coloring options, or when trying to create a movie of a trial that does not exist. In all of these cases the code outputs an error dialog to the user and an error message in the console containing information about the error. In obvious cases the interface selection is changed to a possible selection and highlights the change in the error dialog.

All failsafes described:

Being edited

4.3 Common errors

Being edited