Project report

Study, modeling and simulation of pedestrian walk with regard to the improvement of stability and comfort on walkways

Pavlos Paris Giakoumakis

Database properties

The database used for the modelisation of human walk consists of valid measurements for **215 pedestrians**. More specifically, each subject was asked to cross a 10x1m plane walkway twice at a normal tempo. The equipment used to conduct the experiments included 10 plates that occupy $1m^2$ each, placed sequentially in a horizontal formation in order to create the 10m walkway. Beneath each plate, four load cells were positioned on each corner of the plate in order to measure the forces induced on each one.

Subsequently, the database provided for the analysis was in the form of a cell table of 229 **rows (subjects)** and differed number of **columns** depending on the **steps measured** for each individual (minimum is 21 while maximum is 32 steps) for a sampling frequency of 50Hz. It is important to note that each subject (row) includes the measurement of **both walks** for the specific individual as in if, for example, the first cross was consisted of 14 steps and the second of 13, the columns 1-14 include the measurements for the first walk and 15-27 include those of the second one. Furthermore, some of the 229 rows are empty (do not contain measurements) and as a result, will be considered invalid for the analysis.

Considering the subject (row) index as i and the step (column) index as j, each cell $\{i, j\}$ in the database matrix includes a struct containing the measured information about **force**, **time** and **position** on the plane for the i^{th} pedestrian at the j^{th} step.

Thus, database cell $\{\boldsymbol{i}, \boldsymbol{j}\}.\boldsymbol{time}$ is an array of measured times (a timer was used in the experiments) that step j lasted corresponding to the forces induced during this step which are found on $\{\boldsymbol{i}, \boldsymbol{j}\}.\boldsymbol{force}$ (maximum 70 measurements for each step). Consequently, $\{i, j\}.force$ is an array of measured forces induced during each step (in Newtons) corresponding to the time array.

The $\{\boldsymbol{i}, \boldsymbol{j}\}.\boldsymbol{x\_step}$ and $\{\boldsymbol{i}, \boldsymbol{j}\}.\boldsymbol{y\_step}$ attributes are denoting the relative position of feet through the walkway on the x and y axis for the j-th step. As described above, each plate covers an area of $1m^2$ and the plates are placed subsequently in a horizontal manner. Thus, placing the center of axis in the middle of the plates on height and the start of the first plate on width (as depicted on the figure below), it is concluded that the y_step is a floating point number between -0.5 and 0.5 and the x_step is a positive floating point number between 0 and 10.

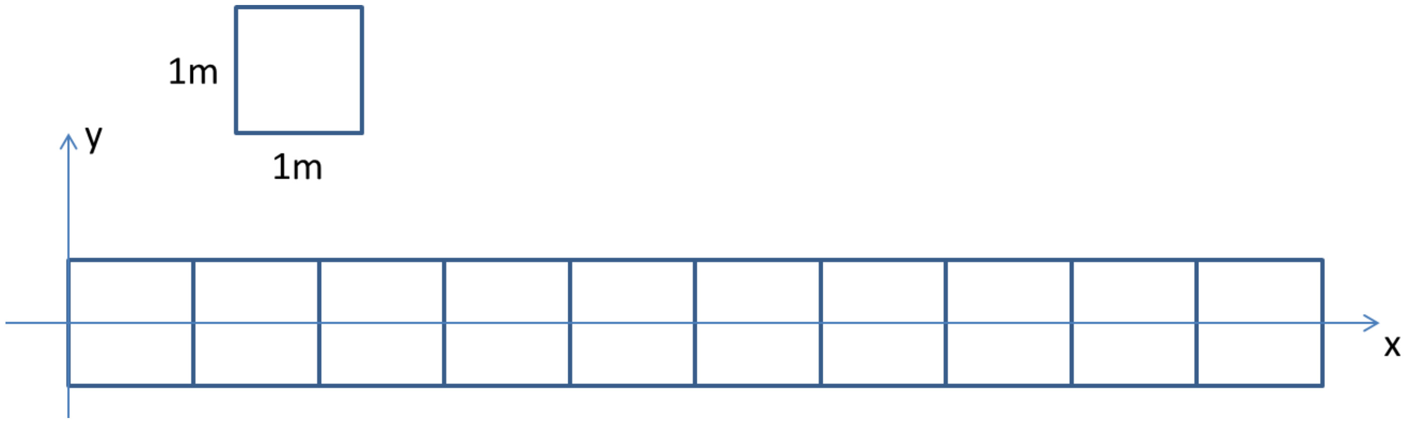


Figure 1: Position measurements axis

Lastly, the first column of each pedestrian ($\{i, 1\}$), includes some extra measurement information. More specifically, $\{\boldsymbol{i}, \mathbf{1}\}.\boldsymbol{x}$ and $\{\boldsymbol{i}, \mathbf{1}\}.\boldsymbol{y}$ contain the information about which points of each plate came in contact through the path, $\{\boldsymbol{i}, \mathbf{1}\}.\boldsymbol{Ptot}$ represents the sum of measurements acquired by all the force sensors beneath the instrumented floor and $\{\boldsymbol{i}, \mathbf{1}\}.\boldsymbol{P\_div}$ collects the signals acquired by all the force sensors (10 plate, 4 sensors for each plate, for a total of 40 measurements for each clock cycle). Those are raw data and will not be used on the specific implementation.

The figure below depicts a simplified form of the initial database which was provided for the analysis. In this example, it is obvious that **each subject differs at number of steps** needed to fulfill the 2 walks on the plane. Also, some subjects do not contain any measurements (subject 228 in this example) and ascribed to this, are considered invalid.

Steps (min. 21 - max. 32)
Steps for 2 walks on the plane

219 Subjects
Some of them contain empty measurements

	step 1	step 2	step 3	step 4	step 5	step 6	step 7	step 8	step 9		step 24	step 25	step 26	step 27	step 28	step 29	step 30	step 31	step 32
Subject 1	force time x_step y_step x y Ptot P_div	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step		force time x_step y_step	force time x_step y_step	force time x_step y_step	[]	[]	[]	[]	[]	[]
Subject 2	force time x_step y_step x y Ptot P_div	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	•••	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step
Subject 3	force time x_step y_step x y Ptot	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step		force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	[]	[]	[]
					⋮					⋮					⋮				
Subject 227	force time x_step y_step x y Ptot P_div	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step		[]	[]	[]	[]	[]	[]	[]	[]	[]
Subject 228	[]	[]	[]	[]	[]	[]	[]	[]	[]	•••	[]	[]	[]	[]	[]	[]	[]	[]	[]
Subject 229	force time x_step y_step x y Ptot	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step	force time x_step y_step		force time x_step y_step	force time x_step y_step	[]	[]	[]	[]	[]	[]	[]

Figure 2: Simplified form of the initial database

Database analysis

The initial database, as described above, needed some modifications in order to satisfy the requirements of the modelisation. First of all, it needed to contain valid measurements only, but also force-time arrays contained in each cell needed to have the same indices on all cells on the basis that the measurements would later be unified in tables for easier and more effective use. For this purpose, the function $clearDb(database)$ was implemented.

Initially, the subjects (rows) with empty measurements were identified and deleted from the database. As a result, the database now contains measurements for **215 subjects** as opposed to the 229 initial subjects. This modification is depicted in *figure 3*. Furthermore, as it is obvious, the force values must always be positive, but due to some errors during the measurement procedure, this was not the case on some occasions. Hence, force measurements with negative values as well as the corresponding time values have been replaced with **nan values** in order to practically remove them from the database.

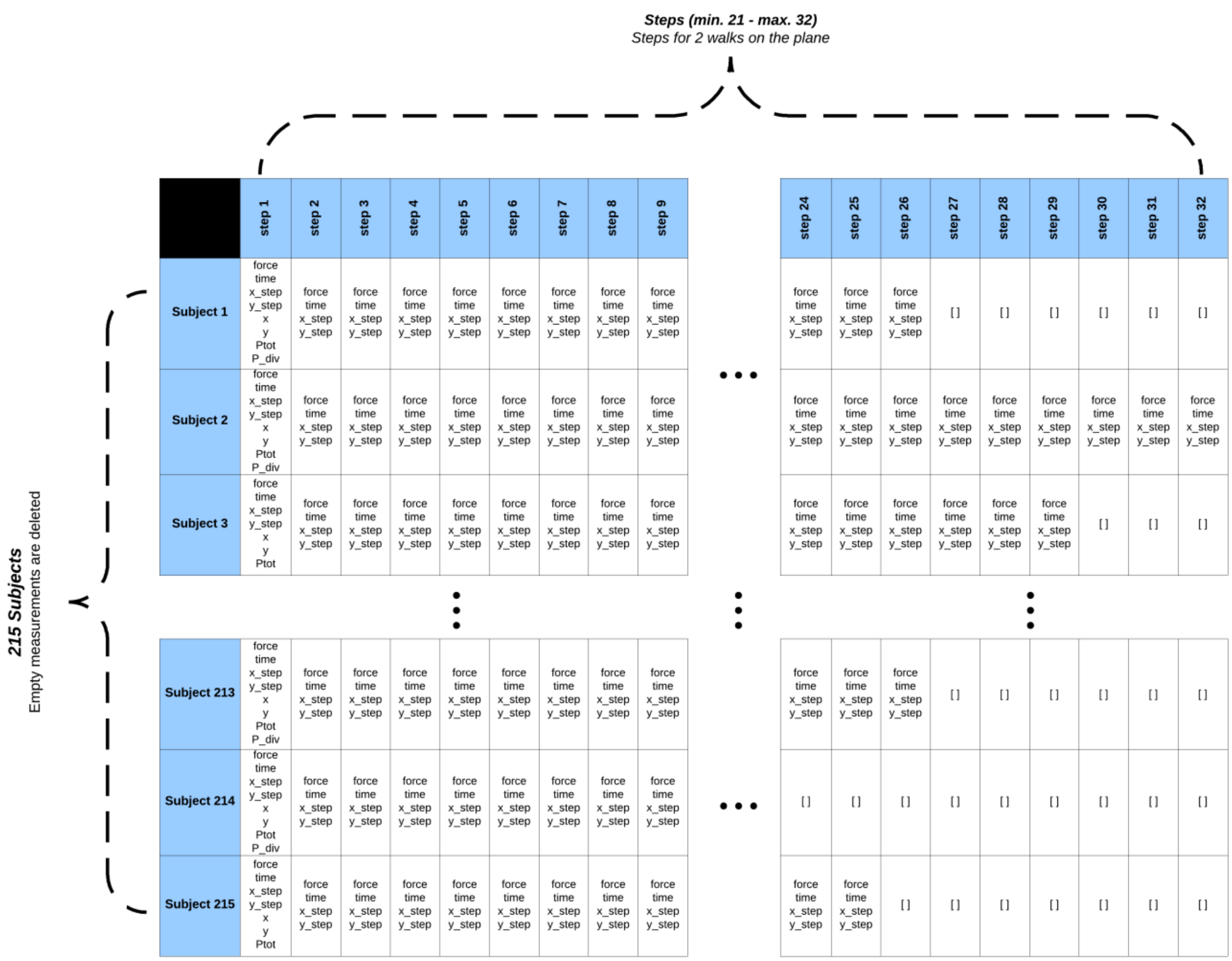


Figure 3: Simplified form of the modified database

Finally, the **force-time arrays** were further modified to have the **same indices on all cells** (these arrays had different indices on each cell but mutually the same) as depicted in *figure 4*. These modifications were crucial as a unified table would be needed for each of those variables. To do so, the maximum force-time array indices in the database were identified and each array was modified to fit these indices by adding nan values to the bottom of the array until the indices reach the maximum number. More specifically, the maximum array indices were calculated as 70 and therefore, each **force-time array** contains up to 70 measurements but has **70 tuples**.

Figure 4 shows an example of how $clearDb(database)$ function works. In this example, it is visible that the pair of force-time will have a length of 70 tuples and all the negative forces (as well as their pairs in time) are deleted after the execution of this function.

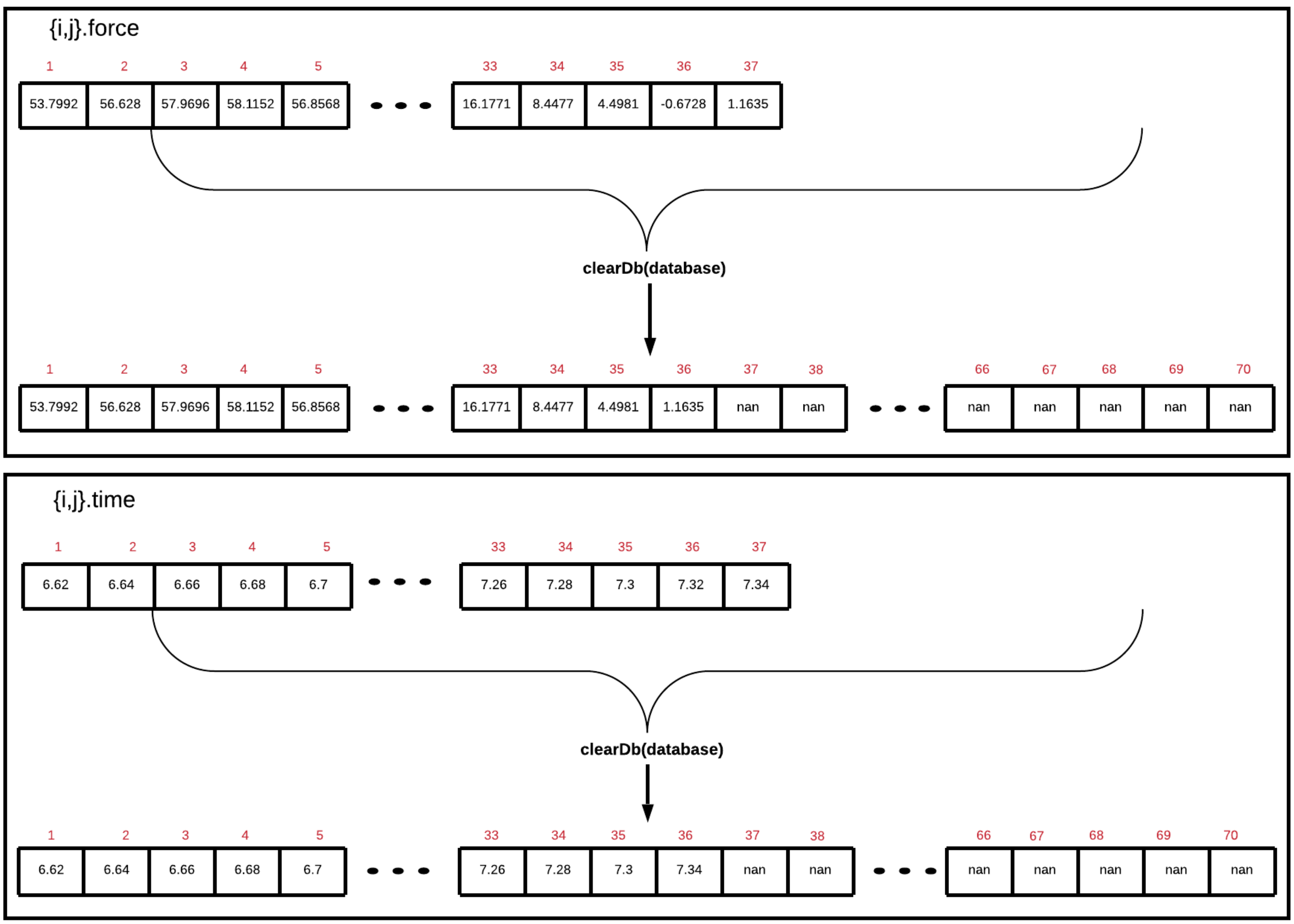


Figure 4: An example of the force-time arrays before and after the execution of clearDb function

Variable extraction

The next big step to perform the modelisation was that of extracting the variables needed. To do so, the 4 different variables contained in the database (force, time, x_step, y_step) were extracted in different tables as hinted previously using the developed function $retrieveAllVariables(database)$. The 4 different tables created had the following forms:

- **time**: 70x32x215 (max measurements x steps x subjects)

  Times in which measurements occurred (1st dim) for each step (2nd dim) and each subject (3rd dim).

- **force**: 70x32x215 (max measurements x steps x subjects)

  Force induced on each moment (1st dim) for each step (2nd dim) and each subject (3rd dim).

- **x_coord**: 215x32 (subjects x steps)

  The step coordinates in x axis for all subjects(rows) and steps (col)

- **y_coord**: 215x32 (subjects x steps)

  The step coordinates in y axis for all subjects(rows) and steps (col)

However, these variables were not fitted to use for the statistical modelisation of human walk. More specifically, the extraction of the following variables is needed:

- **Interarrival time (Dt)**: 215x32 (subjects x steps)

  Interarrival time between each step (col) for each subject (row).

- **Mean Force**: 215x32 (subjects x steps)

  Mean force induced on each step (col) for each subject (row).

- **Length of step**: 215x32 (subjects x steps)

  The distance between each step

- **Angle of step**: 215x32 (subjects x steps)

  The angle of each step along the gait horizontal direction

To extract those, $computeAllDesiredVariables(force, time, x\_coord, y\_coord)$ function has been developed.

Extraction of interarrival time (Dt) and mean force (meanF)

As denoted, the **time** table contains up to 70 time measurements for each step of each subject. These values are indicating the duration of the step, but it would be preferable if there was only one time value characterizing the time in which it has been performed. The selected value for this was the **mean time** of each step. Therefore, for each time array (each step of each subject), the mean value (ignoring the nan values) has been calculated and thus, the mean time of each step has been extracted as depicted in *figure 5*.

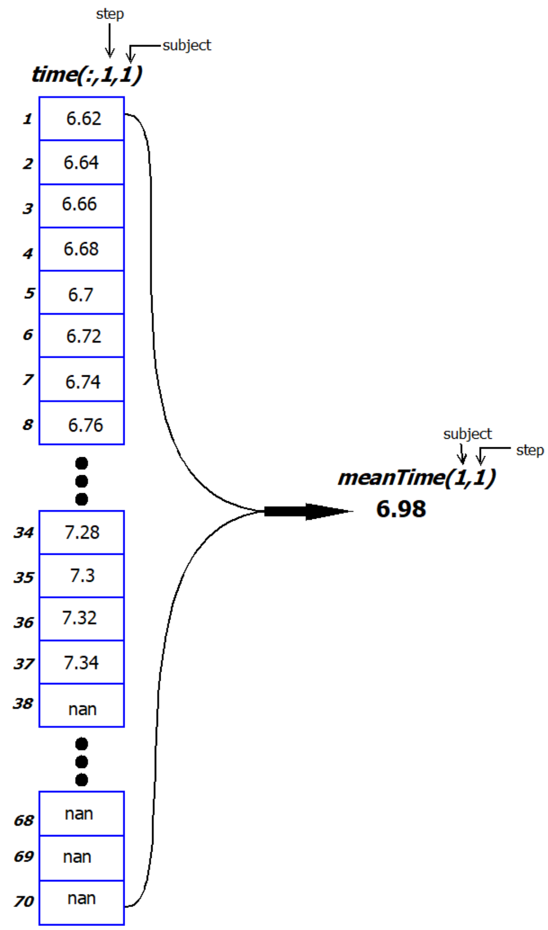


Figure 5: Construction of meanTime table

Let j be the current step of subject i. The **interarrival time ($\boldsymbol{Dt}$)** of step j will be calculated as:

$$Dt(i,j) = meanTime(i,j) - meanTime(i,j-1)$$

Subsequently, each interarrival time between steps can be calculated using this formula. This indicates that the first step ($Dt(i,1)$) has no interarrival time, although the mean time of the first step will be used to calculate the interarrival time between the first and second step. Thus, $Dt(i,1)$ will be nan.

As stated above, each subject includes measurements for 2 walks on the pathway. That means that each time the second walk starts, the first step of the second walk must ignore the previous steps which belongs to the first walk (the values in the $time$, $force$, x_coord and y_coord arrays are continuous as if for example, the last step of the first walk was the 13th, the new walk starts at the 14th step). In order to identify in which step the new walk starts, the x_coord variable is used. More specifically, as x_coord identifies the position of the subject on the plane, on the same walk it will be: $x\_coord(i,j) \geq x\_coord(i,j-1)$. In the case that $x\_coord(i,j) < x\_coord(i,j-1)$, a new walk has started as the subject has returned to the beginning of the walkway. Using this condition, **the step at which the new walk begins can be identified**.

When a new walk is identified, the first measurement will be used to calculate the interarrival time between the first and second step of the second walk and the equivalent

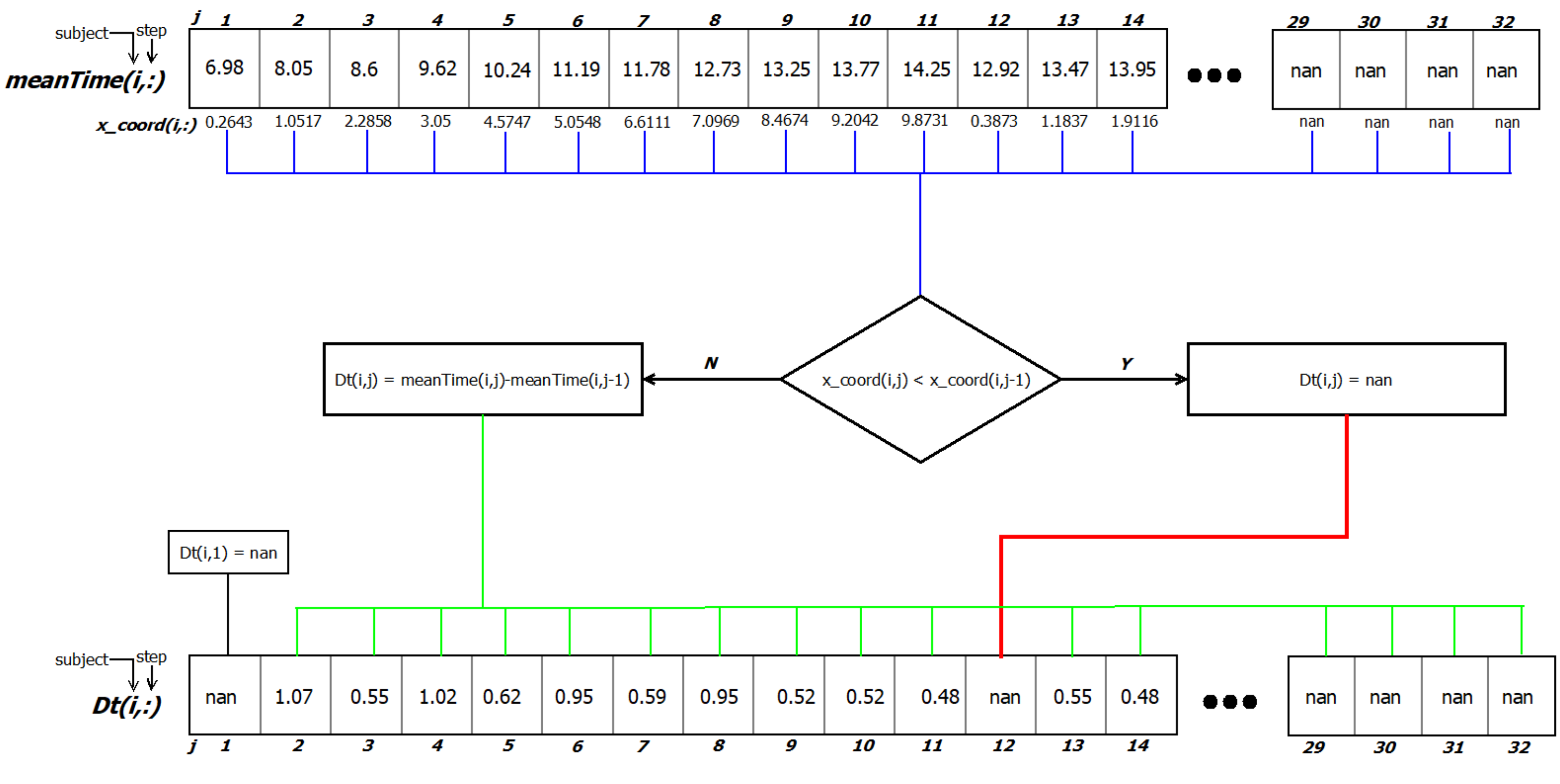


Figure 6: Dt extraction algorithm

interarrival time will take a nan value.

In order to extract the **mean force** induced in each step, a similar procedure will be used. This indicates that using the $force$ table which contains up to 70 measurements for each step of each subject, a mean force induced in each step will be extracted similarly to the way $meanTime$ is extracted (*figure 5)*. Using this method, the $meanF$ table will be constructed.

However, in order to keep the necessary correspondence between Dt and $meanF$, the values of the first step of each walk will be deleted. In other words, in whichever tuples Dt has the value of nan, $meanF$ will get the nan value as well. Lastly, the nan values of Dt and $meanF$ tables are shifted to the end of each row as shown in *figure 7*.

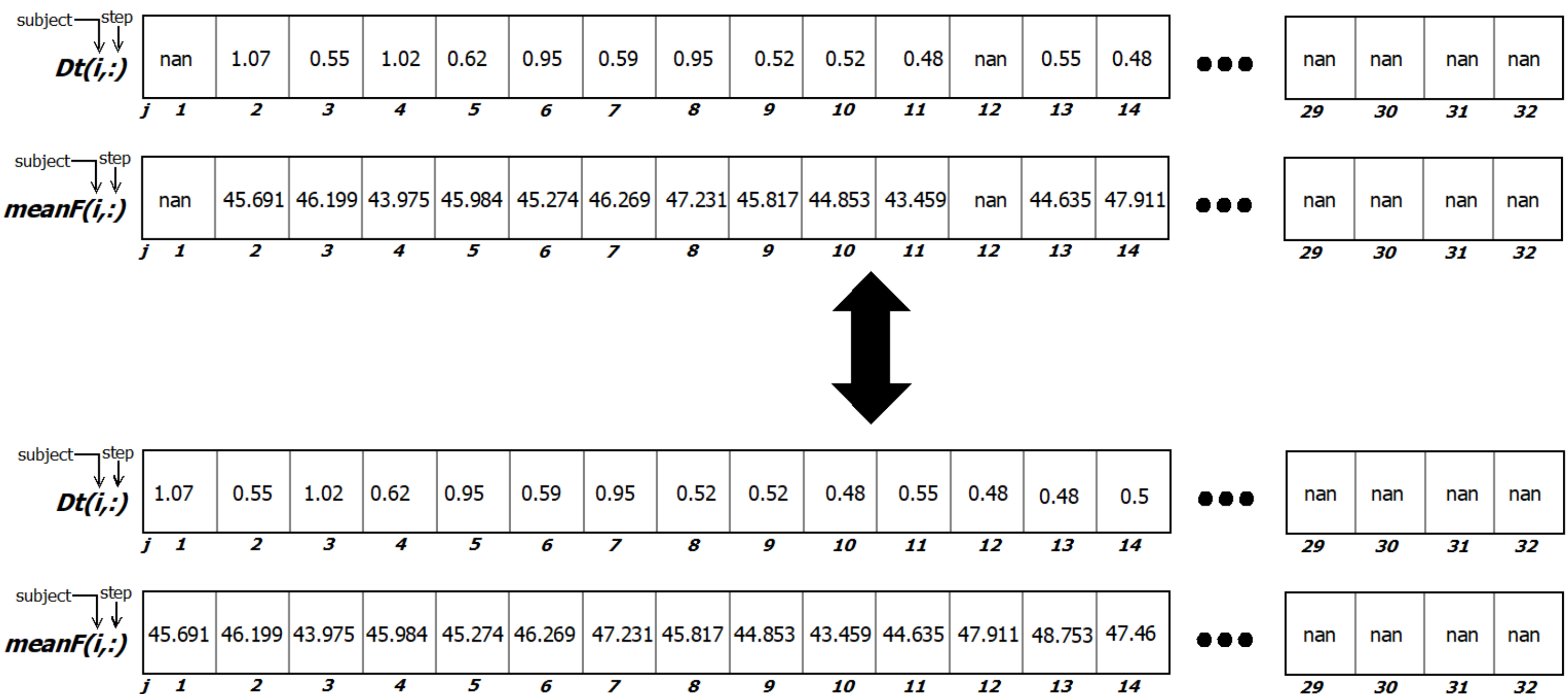


Figure 7: Shifting the nans to the end of each row

Extraction of length (len) and angle

In order to extract the **length** and **angle** of each step, the x_coord and y_coord tables are used. More specifically, having the x and y coordinates of each step, the **length** can be calculated as the Euclidean distance between one step and the next, while the **angle** is the arctangent of the traversed y and x.

This indicates that the first step of each walk will be used to calculate the length and angle between the first and second step but there are no values corresponding to these steps and thus, their values will be nan and shifted to the end of each row following the procedure described for Dt and $meanF$.

Figure 8 illustrates the way length and angle are calculated. In this example, 3 consecutive steps of a subject are depicted as circles in the x and y domain based on the coordinates taken by the x_coord and y_coord tables.

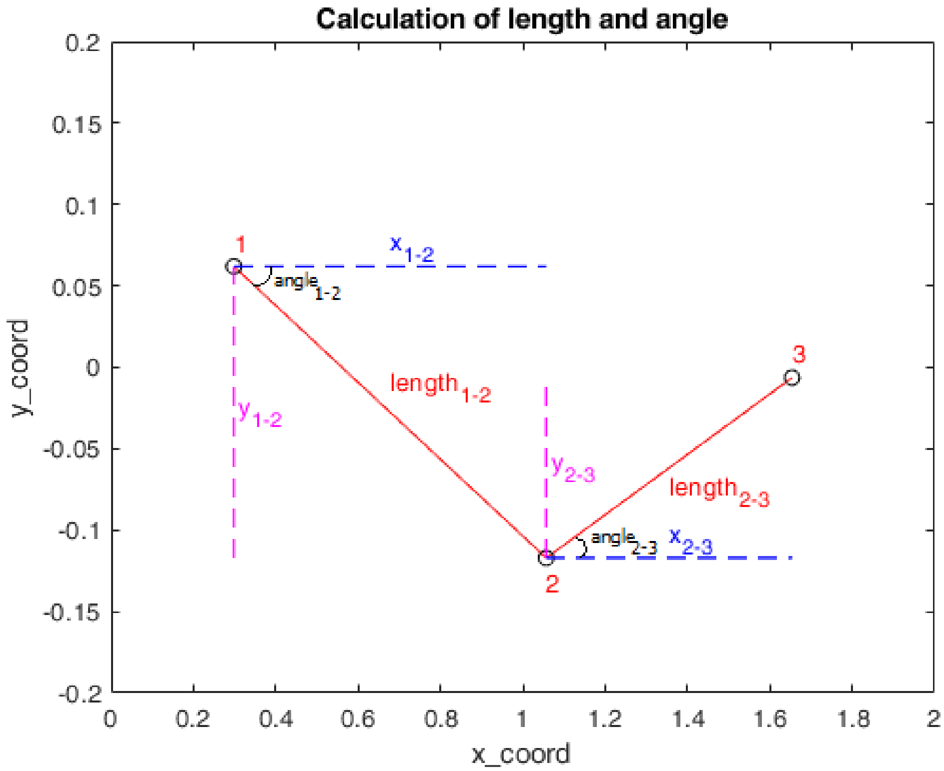


Figure 8: Calculation of length and angle on 3 steps of a walk

The x_{1-2} and y_{1-2} in this example are the traversed x and y distances between the first and second step and thus:

$$x_{1-2} = x_2 - x_1 \text{ and } y_{1-2} = y_2 - y_1$$

Using the Euclidian distance, the distance between the first and the second step (**length** of the 1st step) can be calculated as:

$$length_{1-2} = \sqrt{x_{1-2}^2 + y_{1-2}^2}$$

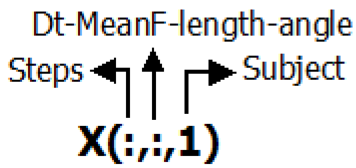
While the **angle** (in radians) between the first and the second step (angle of the 1st step) can be calculated as:

$$angle_{1-2} = \arctan\left(\frac{x_{1-2}}{y_{1-2}}\right)$$

Using this procedure, all lengths and angles traversed on each step of a walk can be determined.

Extraction of the unified table

The 4 extracted variables were combined into table $\boldsymbol{X}$ which is a 32x4x215 table including mean force, interarrival time, length and angle of each step for each person. For each person (represented in the 3rd dimension), 1st column represents the Dt, 2nd column is the mean force, 3rd column is the length and 4th is the angle for the specific step (row). An example of this table's contents is shown in $figure$ 9.



Dt	*meanF*	*length*	*angle*
0.7100	77.6064	0.6944	-0.3491
0.6800	71.6359	0.6703	0.2484
1.2300	72.7029	0.6258	-0.3080
0.6300	76.0684	1.2705	0.0437
1.2300	73.8084	0.7408	0.2756
0.6200	73.4365	1.1367	-0.0108
0.6200	78.9823	0.7407	-0.3258
0.6300	73.4075	0.6407	0.3289
0.6100	74.8464	0.6779	-0.2614
0.5900	78.2279	0.7145	0.2655
0.6100	73.0446	0.6631	-0.2858
0.6100	71.4293	0.6822	0.3033
0.6500	73.9144	0.6752	-0.3757
0.6900	77.1466	0.6518	-0.3095
0.7700	64.1832	0.9149	0.1316
0.4800	67.6896	0.4314	-0.2450
0.5800	72.8245	0.6545	0.3365
0.5900	73.2152	0.6894	-0.3130
1.1800	75.7124	0.7050	0.2969
0.5900	68.2049	1.2786	0.0364
1.2000	74.0852	0.7664	-0.3263
0.7000	66.3817	1.5980	0.0504
0.5200	69.6101	0.4399	0.2893
0.6000	75.1784	0.7305	-0.2810
0.6700	75.6555	0.7438	0.2582
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN

Figure 9: The form of the unified table

Distribution fitting

Martina Fornaciari, a student who conducting a thorough study on the subject for her thesis project, has concluded to the fact that the **angle does not present any correlation with the other three random variables** (Dt, mean force and length). From these results it is clear that the mathematical model used to describe the human walk will need to take into account the other three random variables and **encounter the angle variable separately**. Using this deduction, another student named Thomas Caleri, tried to fit a variety of distributions in order to identify which describes the process of human walk better, during his thesis project. It was concluded that a gaussian mixture model fitted on the 3 variables of Dt, mean force and length was a good approach in most cases.

Based on this deduction, a **gaussian mixture model (GMM)** was fitted for each subject using the MATLAB function $fitgmdist(X, k)$ and the implemented function $fitGMMtoData(X, components, mode)$ resulting to a total of **215 GMMs**. After that, samples based on each GMM were generated and compared with the initial data using the two-sample Kolmogorov-Smirnov test.

The **two-sample Kolmogorov-Smirnov test** is a nonparametric hypothesis test that evaluates the difference between the cdfs of the distributions of the two sample data vectors over the range of x in each data set. In other words, it returns a test decision for the null hypothesis that the initial and generated data are from the same continuous distribution. The result of the test is 1 if the test rejects the null hypothesis at the 5% significance level, and 0 otherwise.

While performing the tests with different number of parameters for the GMM, it was inferred that a GMM with **5 components** is the ideal fit to the samples as in this case almost all tests succeed.

Figures 10, 11 and *12* show a GMM (curve) fitted on the pdfs of a subject for the 3 random variables of Dt, mean force and length.

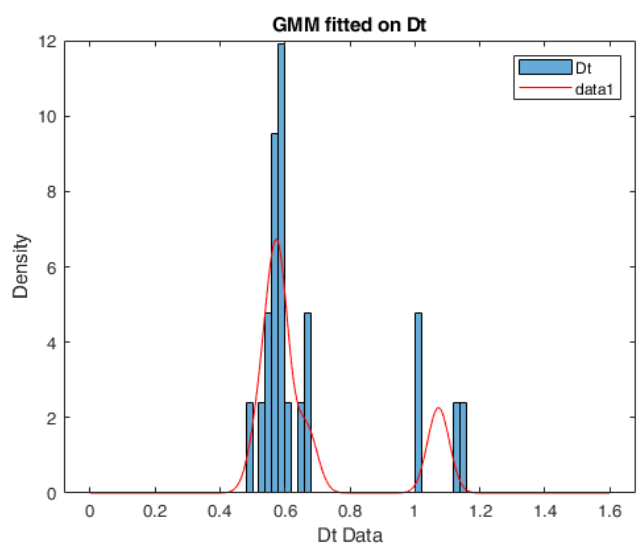


Figure 10: GMM fitted on the interarrival time data of a random subject

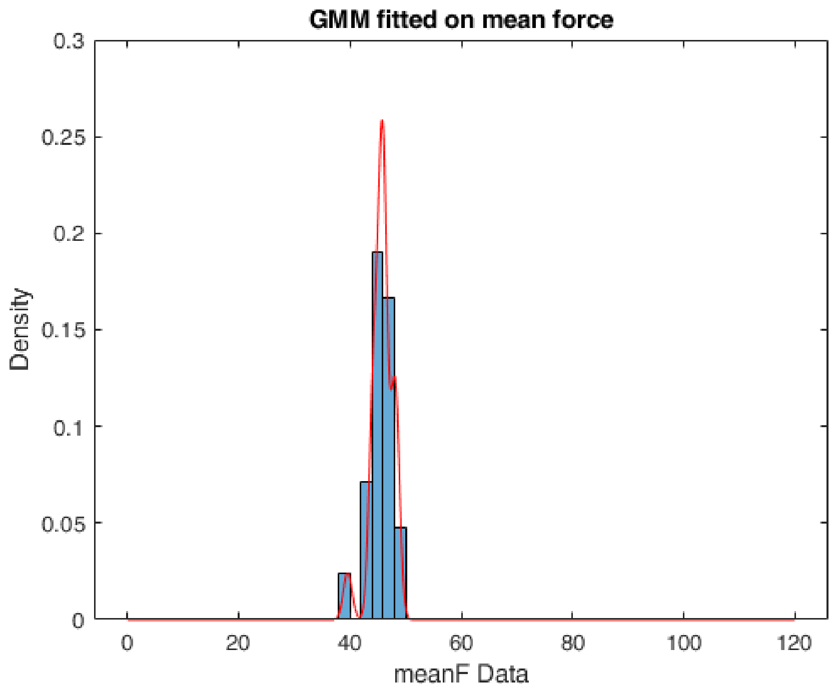


Figure 11: GMM fitted on the mean force data of a random subject

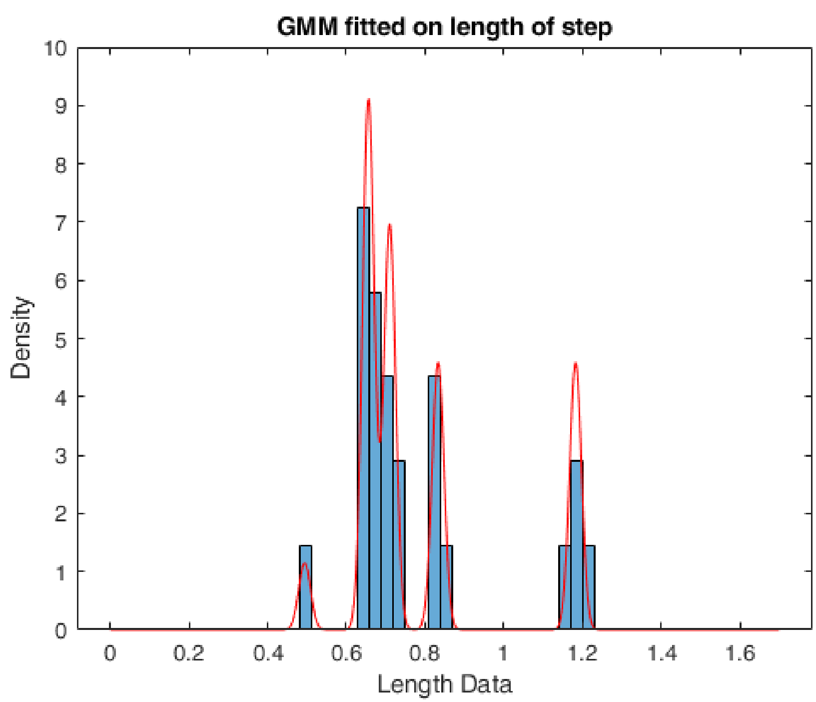


Figure 12: GMM fitted on the length of step data of a random subject

Statistical description of the parameters

The final goal of this project is to create a simulator which will successfully generate a random walk with any given number of steps. This simulator must extract the 4 variables for each step similarly to the experimental data based on a multivariate distribution that describes the human walking process accurately. In order to generate a GMM distribution that describes the walking process, the statistical description of the parameters of the GMM is needed.

More specifically, each GMM is characterised by 3 parameters. Those in fact are the **mean table (mu)**, the **covariance matrix $\boldsymbol{\Sigma}$ (Sigma)** and the **mixing probability $\boldsymbol{\pi}$ (component proportion)**.

As was the case previously, the statistical description of the parameters will be concerning **the 3 variables** (Dt, mean force and length). The angle is a trivial parameter and will be handled separately in a later chapter. Consequently, it will be conducted by mixing the parameters of the 215 GMMs created on the previous step and fitting GMM models on each one of them using different methods in order to describe them as accurately as possible.

Statistical description of the mean table (mu)

The **mean table (mu)** parameter of a GMM which describes 3 variables with 5 components is a **5x3 table (components x variables)**. Each column corresponds to a different variable and each of the 5 rows describes the mean value of each component on a specific variable. This indicates that each variable (column) must be described by different GMMs.

The approach followed for the most accurate statistical description of the mu of each variable, will be illustrated using an example. The tables depicted in *figure 13* are an example of the mean tables used as parameters on each subject's GMM. The statistical description of the mu parameter for the first variable (Dt) will be conducted.

variables
components ← ↑ → subject

mu(:,:,1)

	1	2	3
1	0.5468	47.5699	1.1028
2	0.4833	43.4949	0.6884
3	0.976	45.8564	0.6167
4	0.5525	45.8911	1.4806
5	0.4869	48.0387	0.7523

mu(:,:,2)

	1	2	3
1	0.5367	51.688	1.4102
2	0.9581	50.413	0.4895
3	0.68	22.4753	0.7793
4	0.5424	50.7402	0.7444
5	1.0566	50.6988	0.7528

⋮

mu(:,:,215)

	1	2	3
1	0.4901	51.8569	0.5034
2	0.5491	56.0912	0.7042
3	0.5567	55.62	1.4044
4	1	55.8091	0.5066
5	0.6675	57.1842	1.051

Figure 13: An example of mu tables

The first step to this approach is to take the **5 values** of each mean table that correspond to the Dt variable and put them in a **unified 215x5 mu_table (subjects x components)**. Then, **sort each row** in ascending order. This procedure will produce a table of 5 columns with the mu values of each subject sorted. The outcome depicted in *figure 14*.

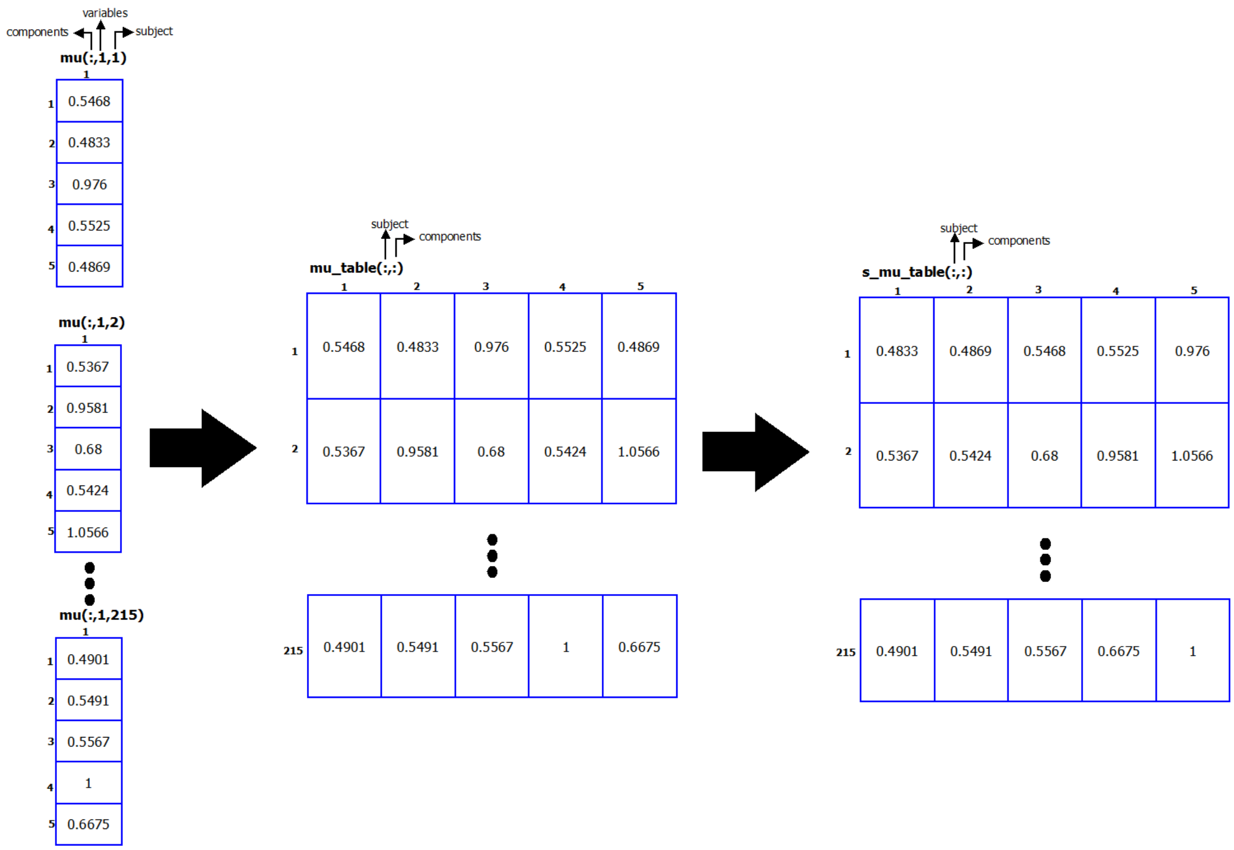


Figure 14: Unifying the mu arrays in a table and sorting the rows

The decision of sorting after creating the unified table was based to the fact that this ensures that the modelisation of mu **does not ignore the low values** which play a significant role. The final step is to **fit a GMM of 3 components on each mu component**. Thus, in each column, a GMM is fitted, resulting in 5 GMMs the first variable. The same procedure is performed for all 3 variables which means that finally, the statistical description of mu consists of **15 GMMs**, 5 for each variable. *Figures 15, 16* and *17* show the fitted GMMs for each variable.

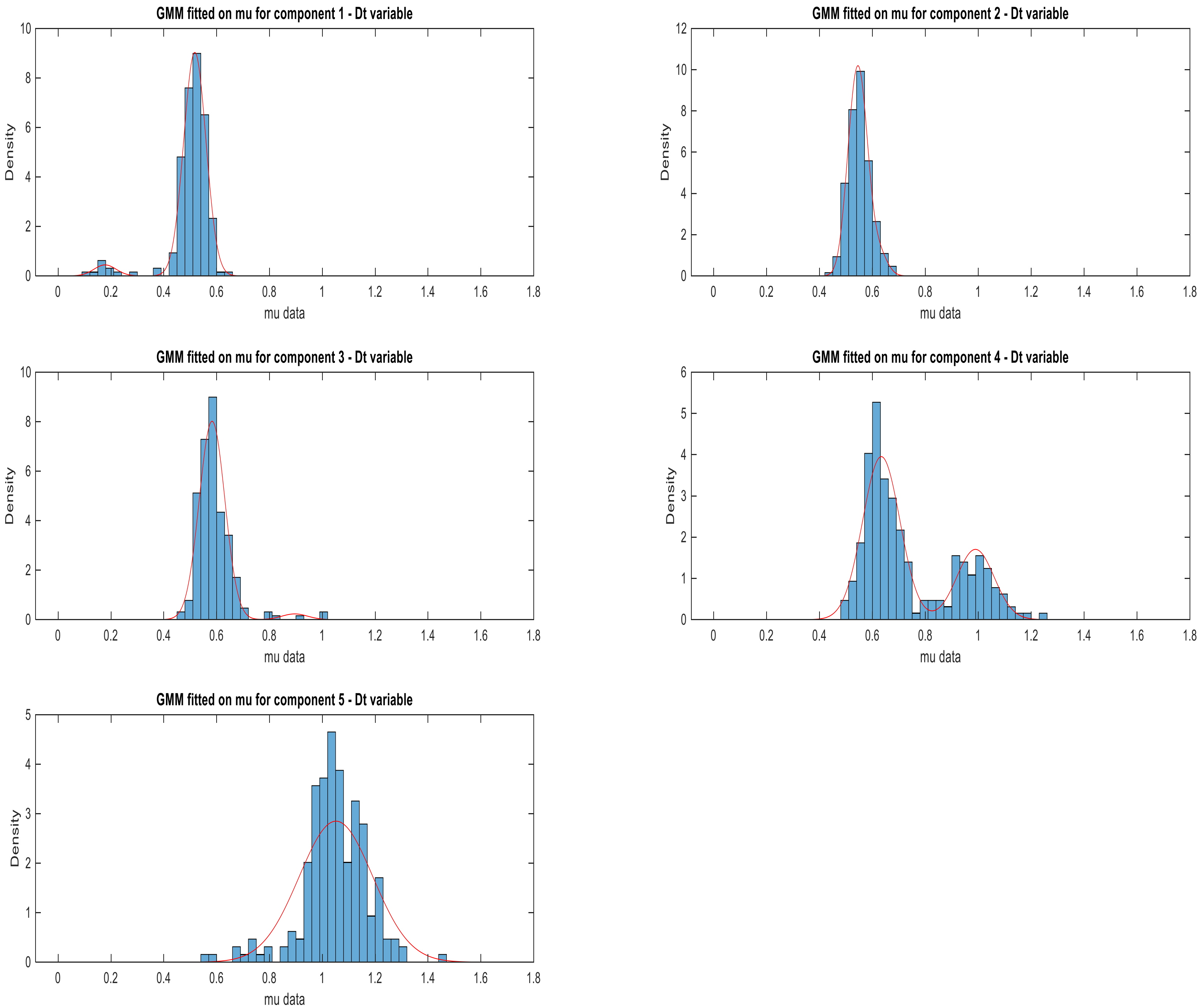


Figure 15: GMMs fitted on mu - Dt variable

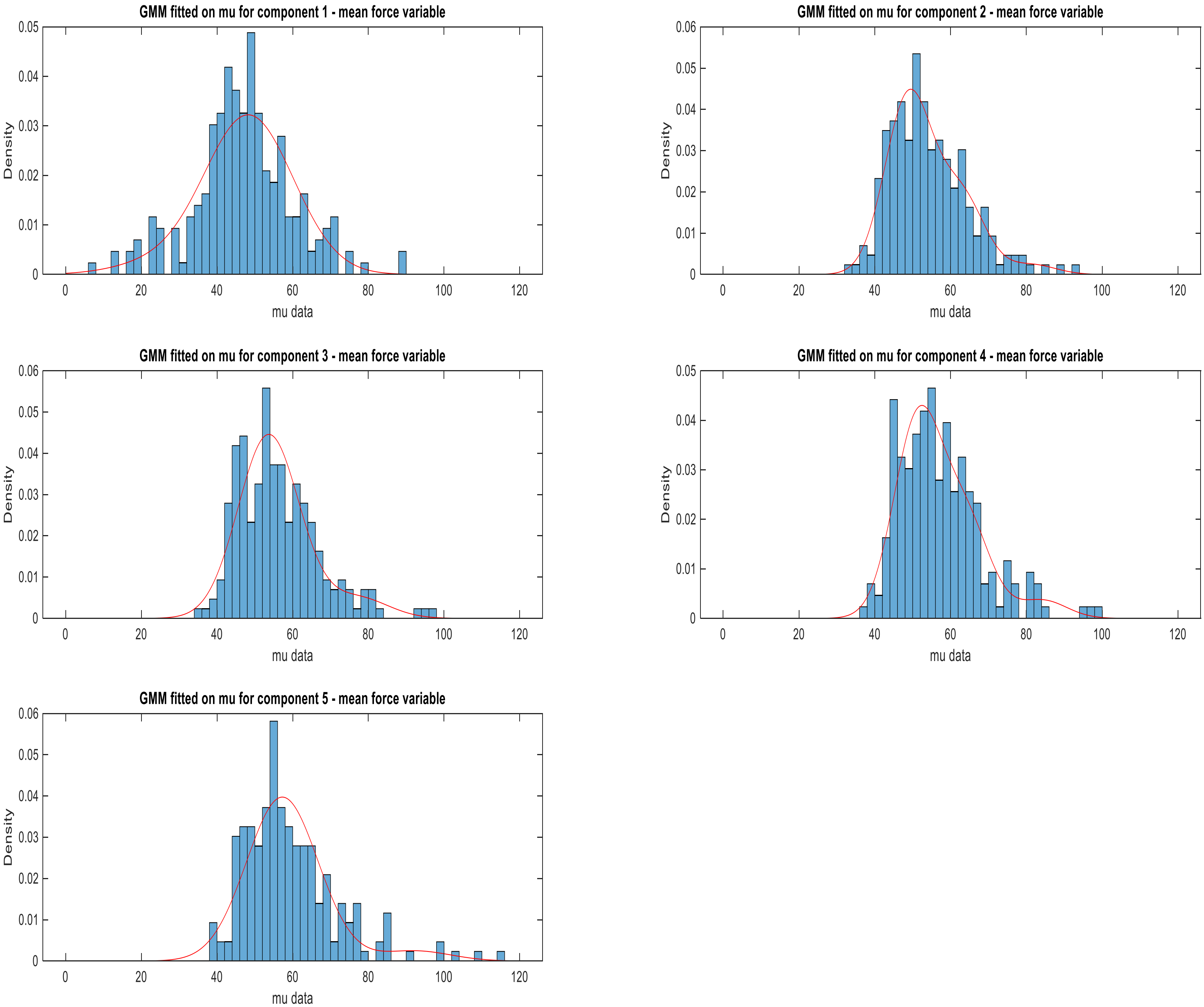


Figure 16: GMMs fitted on mu - meanF variable

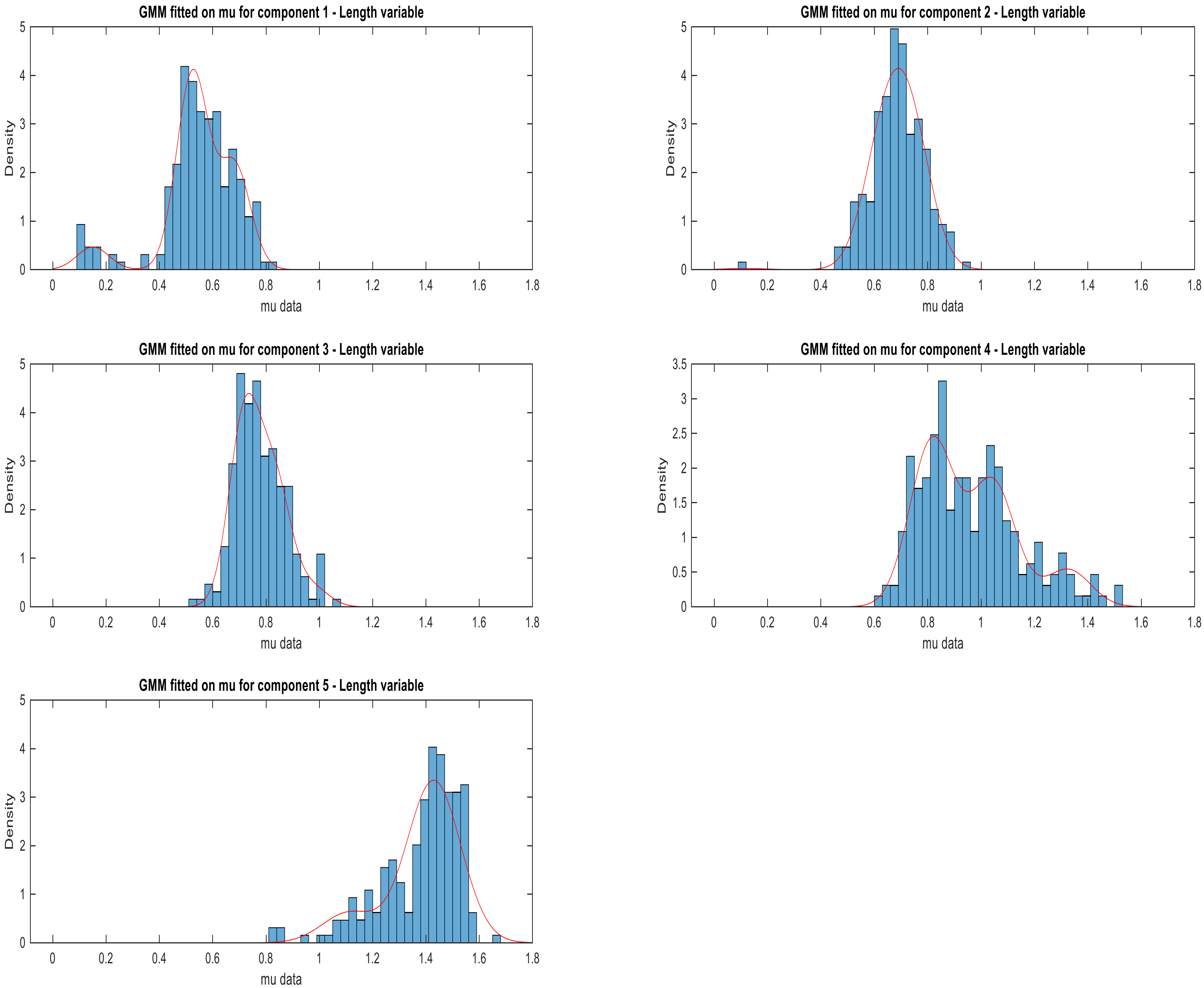


Figure 17: GMMs fitted on mu – Length variable

Statistical description of the mixing probability (component proportion)

The **mixing probability** parameter (**ComponentProportion**) of a GMM of 5 components is a **5-tuple array** (as the number of components of the GMM). Each tuple describes the proportion of each component (weight) and is a number **between 0 and 1**. **The sum of all proportions is 1**. In order to statistically describe this parameter, a GMM will be fitted for each component's proportion.

This parameter needs to be in accordance with the component it describes, thus the approach for its description will be conducted alongside the statistical description of the mu table (as if not, the sorting process will lead to not keeping the correlation between the component proportions and their mean values). Subsequently, as depicted in *figure 18*, the component proportions of each subject are unified in a **215x5 weight_table**. Then, while the sorting of mu_table occurs, the pairs in weight_table are **fixed based on this sorting**. This way, the correlation between the component proportions and the mean values will remain after the sorting.

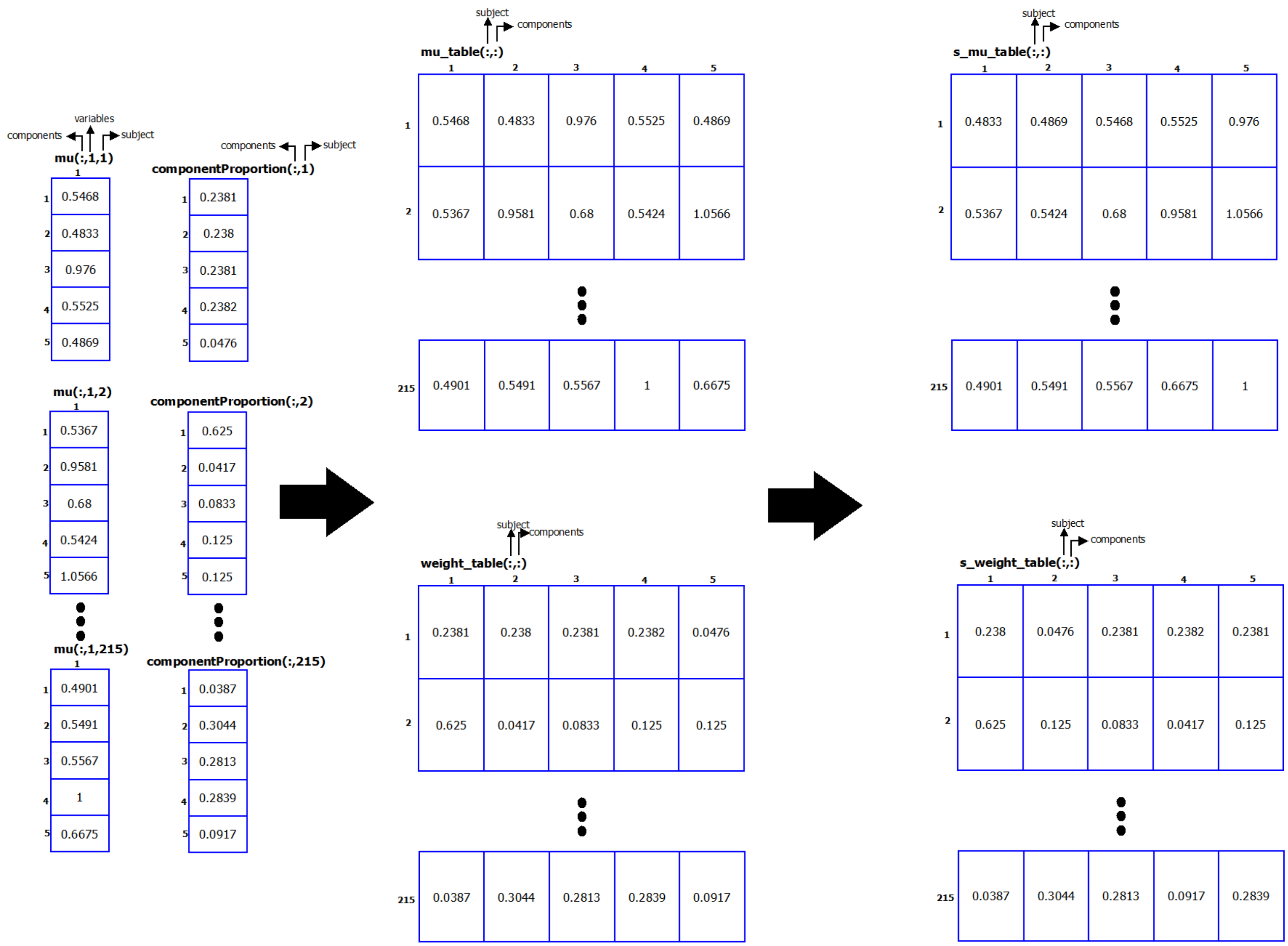


Figure 18: Unifying the component proportion arrays in a table and fixing pairs with mu_table on sorting

The final step is to **fit a GMM of 3 components** on each component proportion. Thus, in each column, a GMM is fitted, resulting in **5 GMMs**. *Figure 19* shows the fitted GMMs for each component.

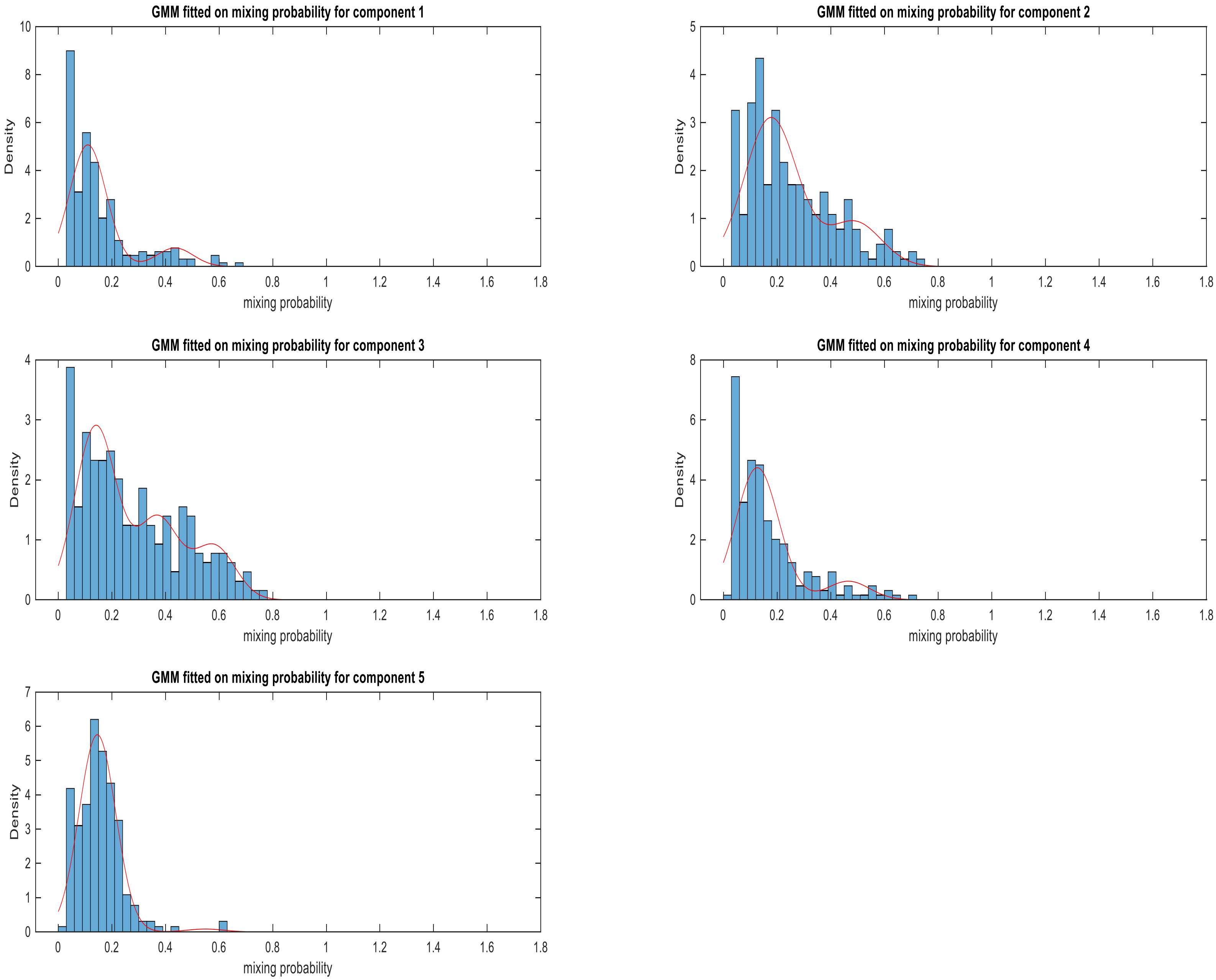


Figure 19: GMMs fitted on mixing probability

Statistical description of the covariance matrix (Sigma)

The **covariance matrix (Sigma)** parameter of a GMM which describes 3 variables is a **3x3 square matrix** giving the **covariance between each pair of their respective variables**. In the matrix diagonal there are variances, i.e., the covariance of each random variable with itself. Every covariance matrix is **symmetric** and **positive semi-definite**. *

Due to this, the differentiated values of a 3-variable covariance matrix are 6. Those are the 3 diagonal values (variance of each variable) as well as the covariances between the 1st and 2nd variable, the 1st and 3rd variable and finally, the 2nd and 3rd variable. Thus, for the most accurate statistical description of Sigma, a **unified table containing the 6 differentiated values of all subjects** (columns in the order described) was created as shown in figure *20*.

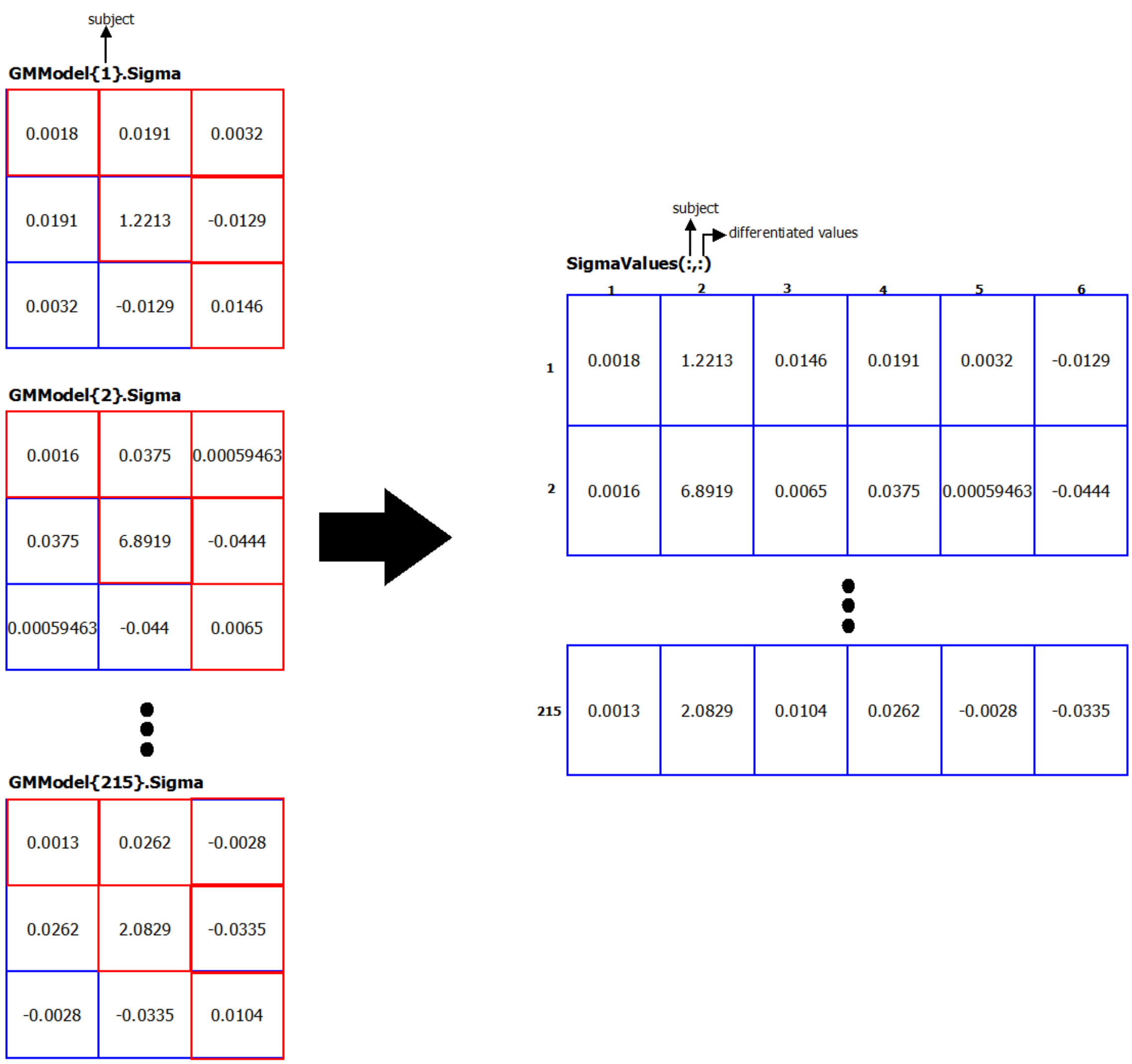


Figure 20: Construction of the unified Sigma table

Finally, a GMM of **4 components** is fitted on each differentiated values (column) of the table resulting in **6 GMMs**. Figure 21 depicts the 6 GMMs fitted on the covariance differentiated data.

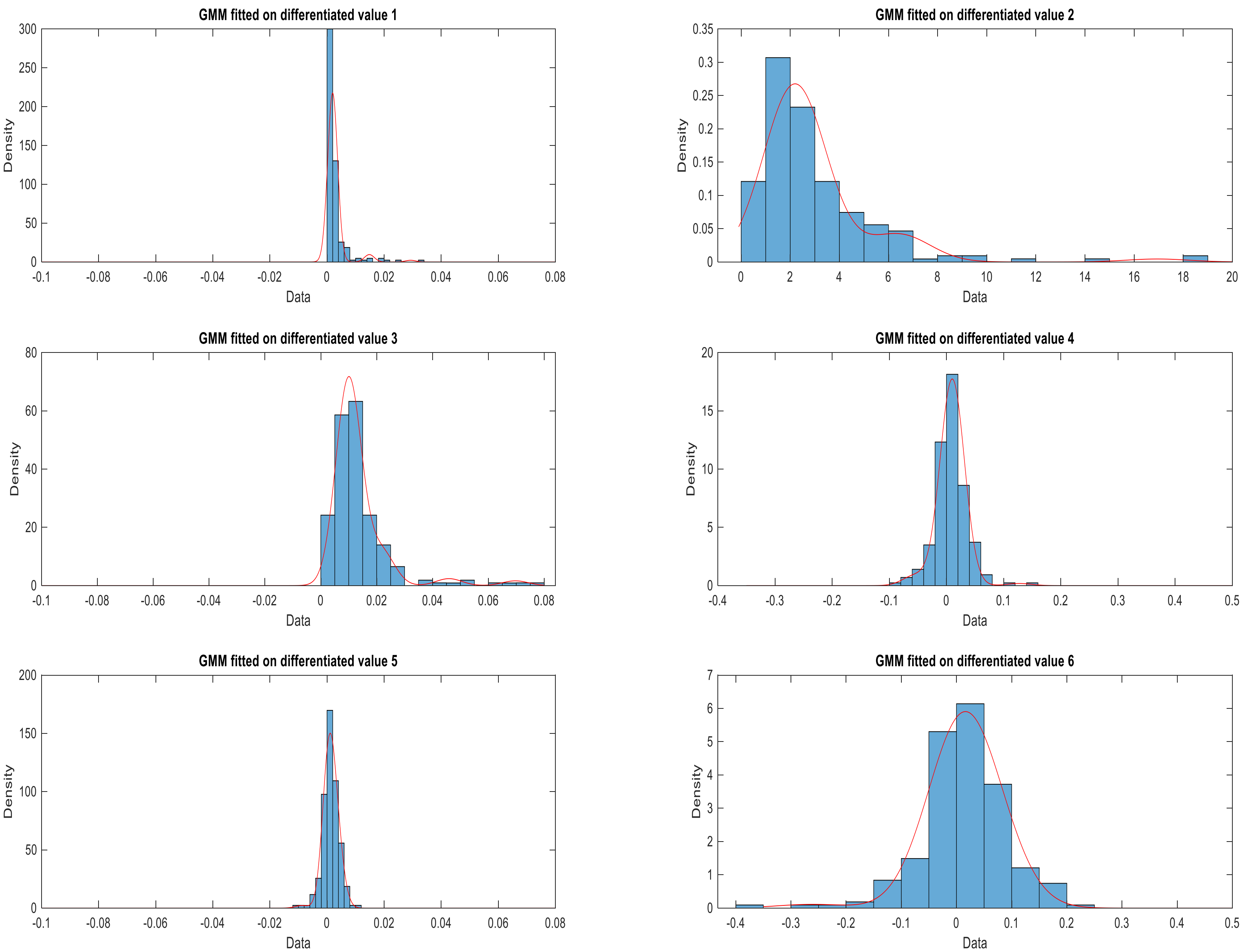


Figure 21: GMMs fitted on the differentiated values of the covariance matrix

Distribution fitting and statistical description of angle

As mentioned before, the random variable which describes the **angle** of a step does not present any correlation with the other three random variables (Dt, mean force and length). Consequently, the distribution that describes the angle variable and the statistical description of its parameters will be conducted separately in this field.

Searching for a distribution that describes the angle variable accurately, a **GMM was fitted for each subject** using the matlab function $fitgmdist(X, k)$ and the implemented function $fitGMMtoData(X, components, mode)$ on the angle data resulting to a total of 215 GMMs. The GMMs fitted are having **3 components** as this number was proven to be the ideal since it gives no errors on the Komogorov-Smirnov tests. *Figure 22* shows a GMM fitted on the pdf of a subject for the random variable of angle.

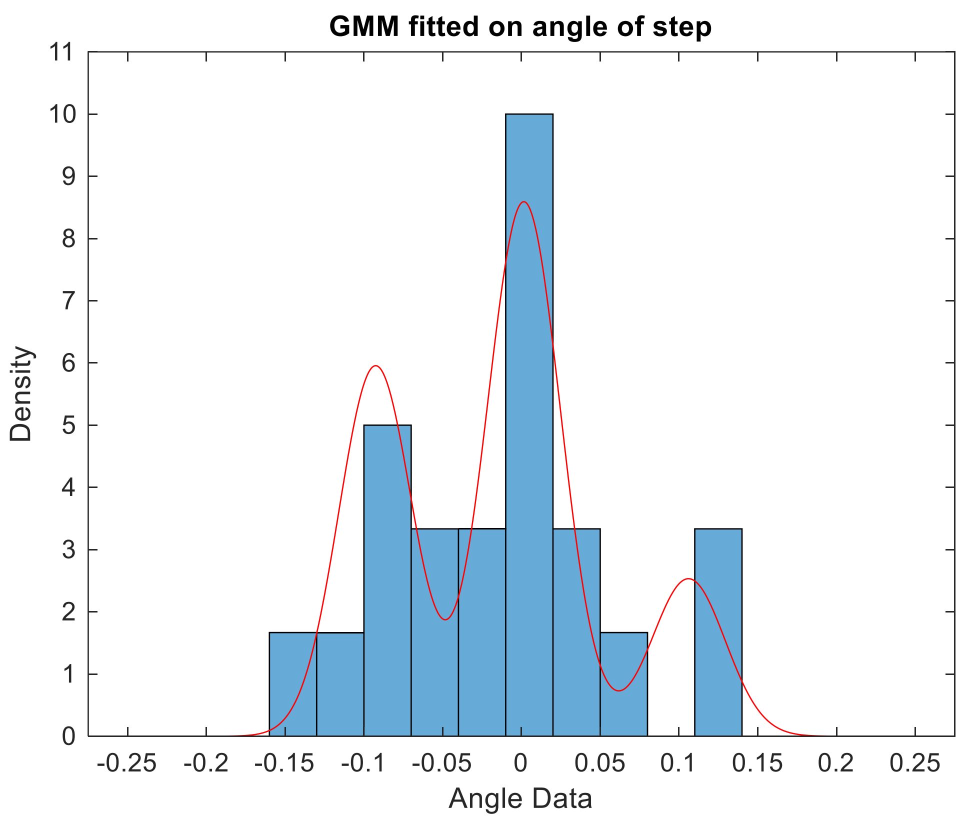


Figure 22: GMM fitted on the angle data of a random subject

Similarly to the case of the 3 random variables, a statistical description of the parameters of the GMMs is needed in order to create a final GMM that describes the angle variable accurately and which will be used in the simulator. In this case, each of the 215 GMMs whose parameters will be used to perform the statistical description, are GMMs of **1 variable and 3 components** counter to the previous case of 3 variables and 5 components.

The **mean table (mu)** which describes the angle variable (for a GMM of 1 variable and 3 components) is actually a **3-tuple array**. As obvious, each tuple corresponds to the mean value of each component. Thus, a **215x3 mu_table_angle** is created (each column contains the mean values for a specific component). Then, **each row** of the mu_table_angle is **sorted** in ascending order as previously. Finally, a **GMM of 3 components** is fitted on each column resulting in **3 GMMs** to describe the mu of the angle.

The statistical description of the **mixing probability** parameter **(ComponentProportion)** follows the same procedure as previously. In this case (GMM of 1 variable and 3 components), it consists of a **3-tuple array** with each tuple corresponding to the proportion of each component. Therefore, a **215x3 weight_table_angle** is created and then, while the sorting of the mu_table_angle occurs, the pairs in this table are **fixed** based on this sorting. As a final step, another **GMM of 3 components** is fitted on each column resulting in **3 GMMs** to describe the mixing probability of each component.

Figure 23 depicts this procedure to extract the final tables upon which the GMMs are fitted, while *figure 24* and *25* display the fitted GMMs on each parameter.

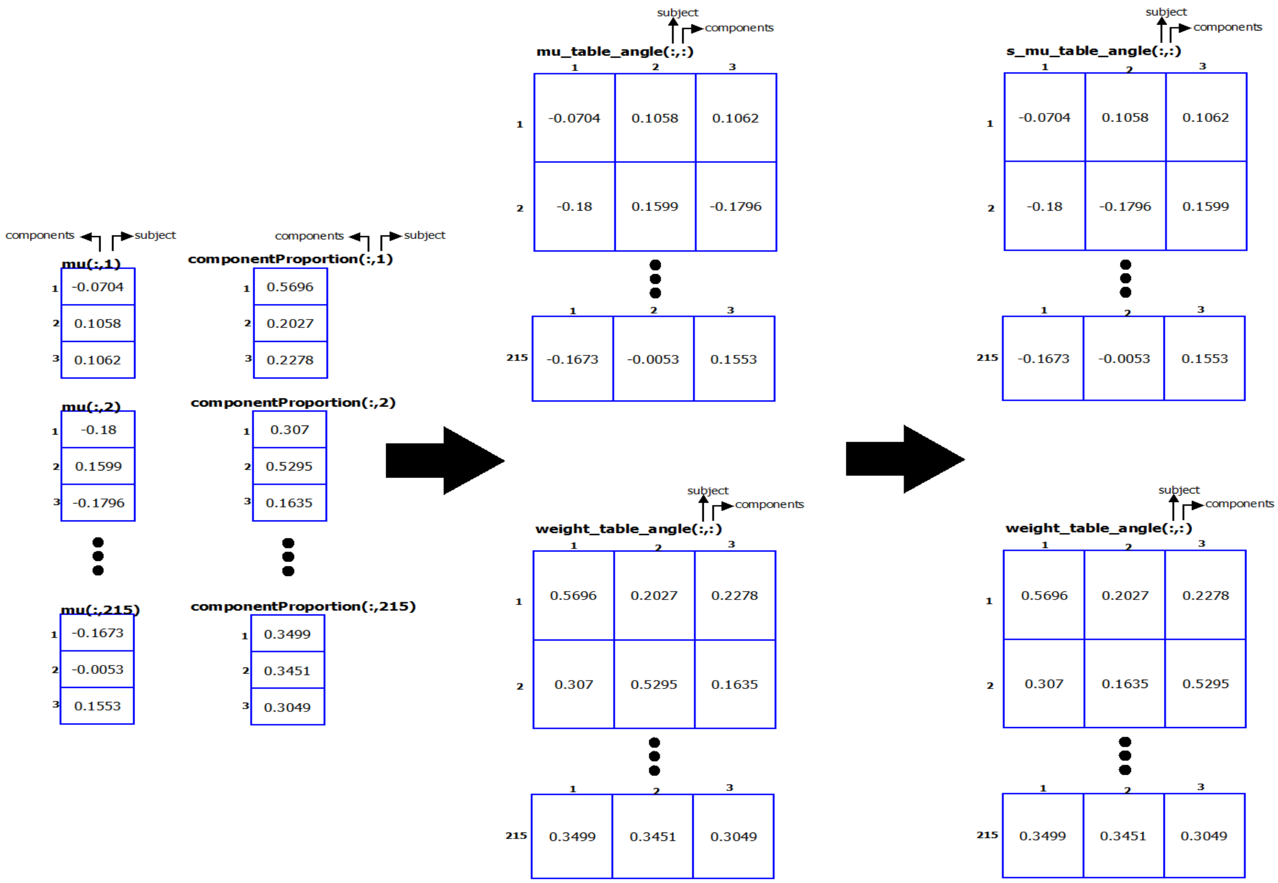


Figure 23: Creation of mu and proportion tables and sorting-fixing pairs

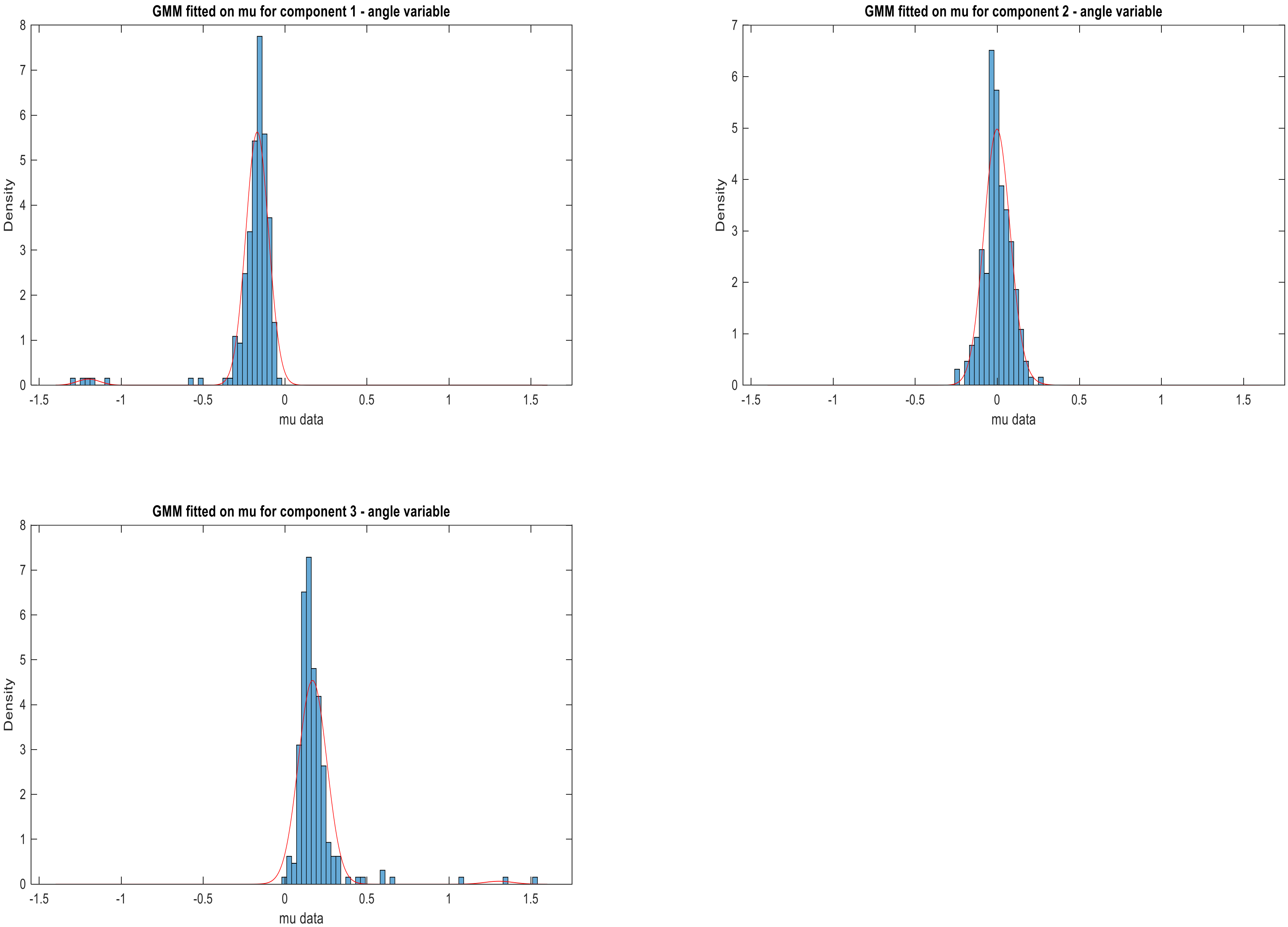


Figure 24: GMMs fitted on mu – Angle variable

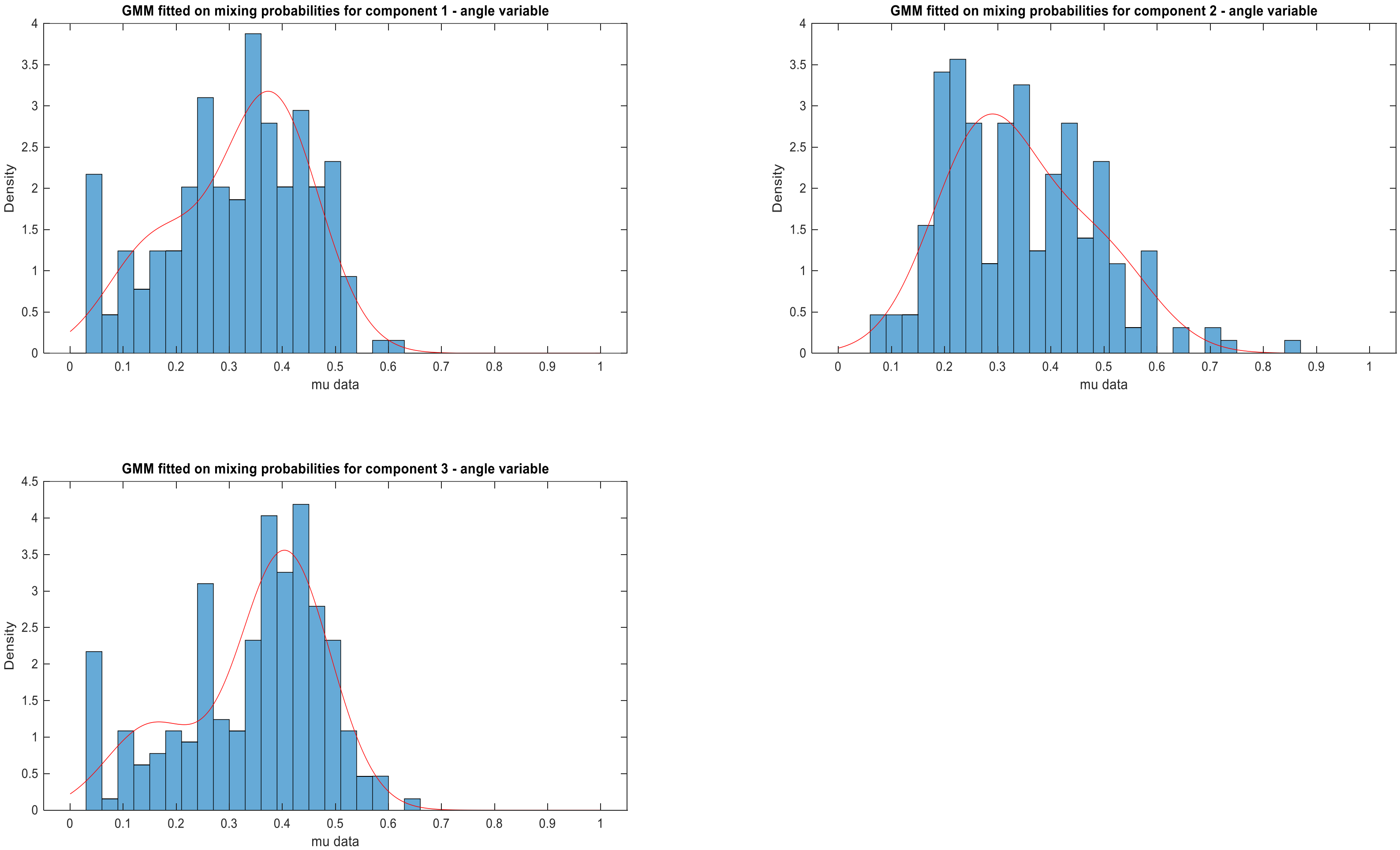


Figure 25: GMMs fitted on mixing probability – Angle variable

Finally, the **Sigma** parameter in the case of a GMM with **1 variable** is just a **value** containing the variance of the variable. Therefore, the statistical description of the **Sigma** parameter was conducted by creating an array containing the 215 values of Sigma and fitting a **GMM with 3 components**. *Figure 26* depicts how the GMM is fitted on the Sigma array.

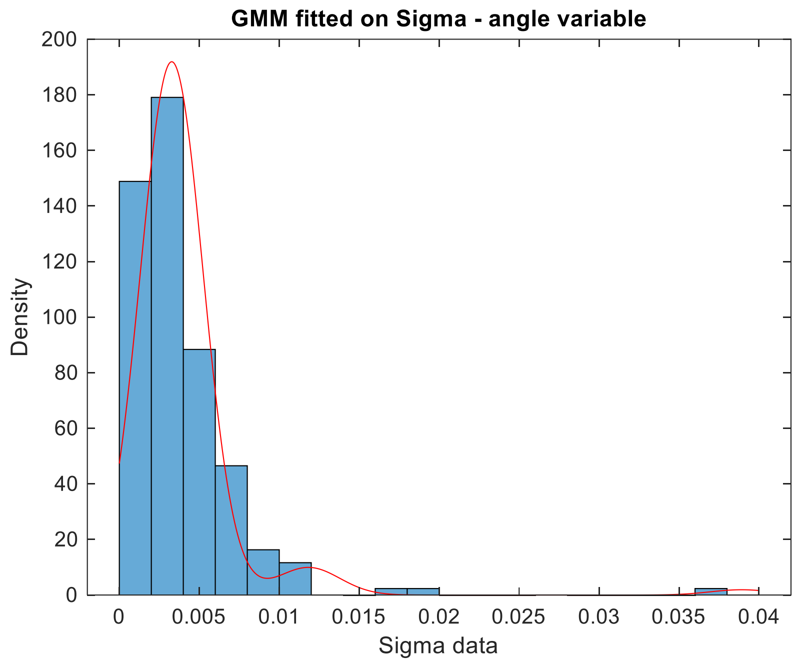


Figure 26: GMM fitted on Sigma – Angle variable

Synopsis of the modelisation

The statistical description of the parameters of the Gaussian Mixture Models (GMMs) concludes the process of **modelisation of human walk**. This modelisation was, synoptically, implemented following the steps described in the previous sections.

The first step was to extract the necessary variables to describe a human walk using the data contained in the provided database. These variables where the **interarrival time (Dt)**, the **mean force (meanF)**, the **length (len)** and the **angle** of each step. Thus, those variables were extracted for each step of each person. Then, 2 GMMs were fitted in each person's walks; one for Dt, meanF and len **(GMModel)** and one for the angle variable **(GMMAngle)**.

Finally, the statistical description of the parameters of those GMMs was conducted. More specifically, each GMM is parametrized by the **mixing probability (ComponentProportion)**, the **mean table (mu)** and the **covariance matrix (Sigma)**. Hence, upon some procedures, these parameters were fitted in GMMs in order to statistically describe them. The process of the statistical description of the **3 variables** resulted in:

- **15 GMMs to describe the mu parameter** – 5 for each variable (as the GMModel consists of 5 components)
- **5 GMMs to describe the ComponentProportion parameter** (as the GMModel consists of 5 components)
- **6 GMMs to describe the Sigma parameter** - each Sigma is a 3x3 symmetric matrix (as the GMModel describes 3 variables) and thus, there are 6 differentiated values.

While the statistical description of the **angle** variable resulted in:

- **3 GMMs to describe the mu parameter** (as the GMMAngle consists of 3 components)
- **3 GMMs to describe the ComponentProportion parameter** (as the GMMAngle consists of 3 components)
- **1 GMM to describe the Sigma parameter** (as the GMMAngle describes 1 variable)

The diagram of *figure 27* shows this procedure more analytically.

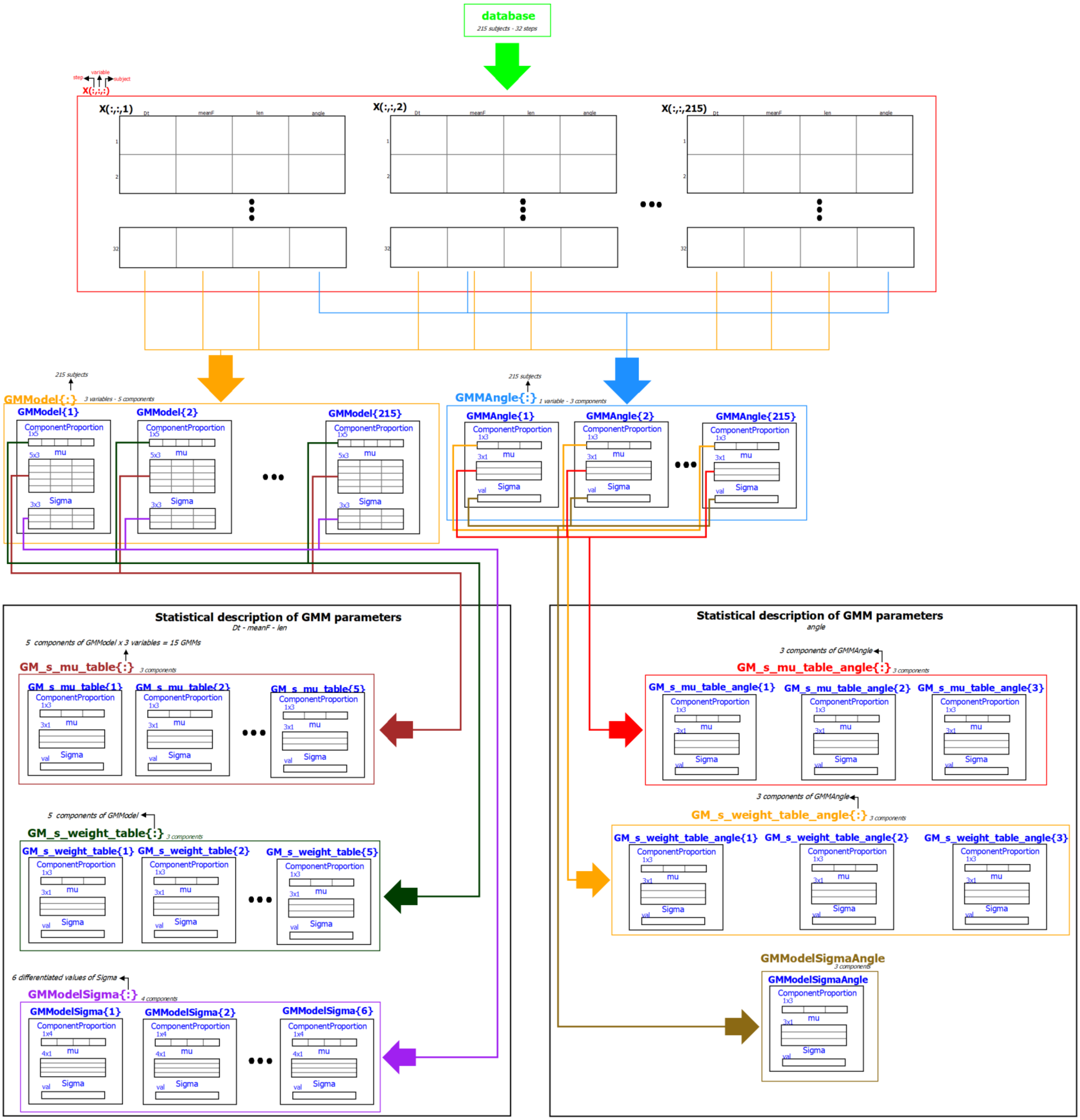


Figure 27: Implementation of the modelisation of human walk

Implementation of the simulator

The scope of the **simulator** is to **describe a random human walk** accurately by **extracting a set of the 4 variables for a desired finite number of steps (*n*)**. In order to achieve this, a **reversed process** than the one followed for the modelisation will be conducted. More specifically, the goal is to extract a random walk using the **statistical description of the parameters** acquired through the modelisation, as input. Thus, using the GMMs of mu, mixing probabilities and Sigma, **a random generation of those parameters** is performed. Then, **a GMM is fitted to those parameters**. This GMM will be used as a distribution that best describes the process of human walk. Consequently, using this GMM, n sets of the Dt, meanF and len random variables will be acquired and the same procedure will be followed to extract the angle variable. **Those sets will describe a random walk of n steps**. This process is depicted graphically in the diagram of *figure 32*.

Parameter extraction

As reported previously, the first step of the implementation is to **generate a set of parameters** to use on the GMM which will statistically describe the human walk. Same as before, the first **3 variables** will be extracted first.

Consequently, using the **5 GMMs describing the mu** of each variable (totally 15 GMMs), a **random value** is generated out of each GMM. Thus, the 5 GMMs produce 5 random values for each of the three random variables and the unification of those produces the **final mu table** which will be used as a parameter in the final GMM (*figure 28*).

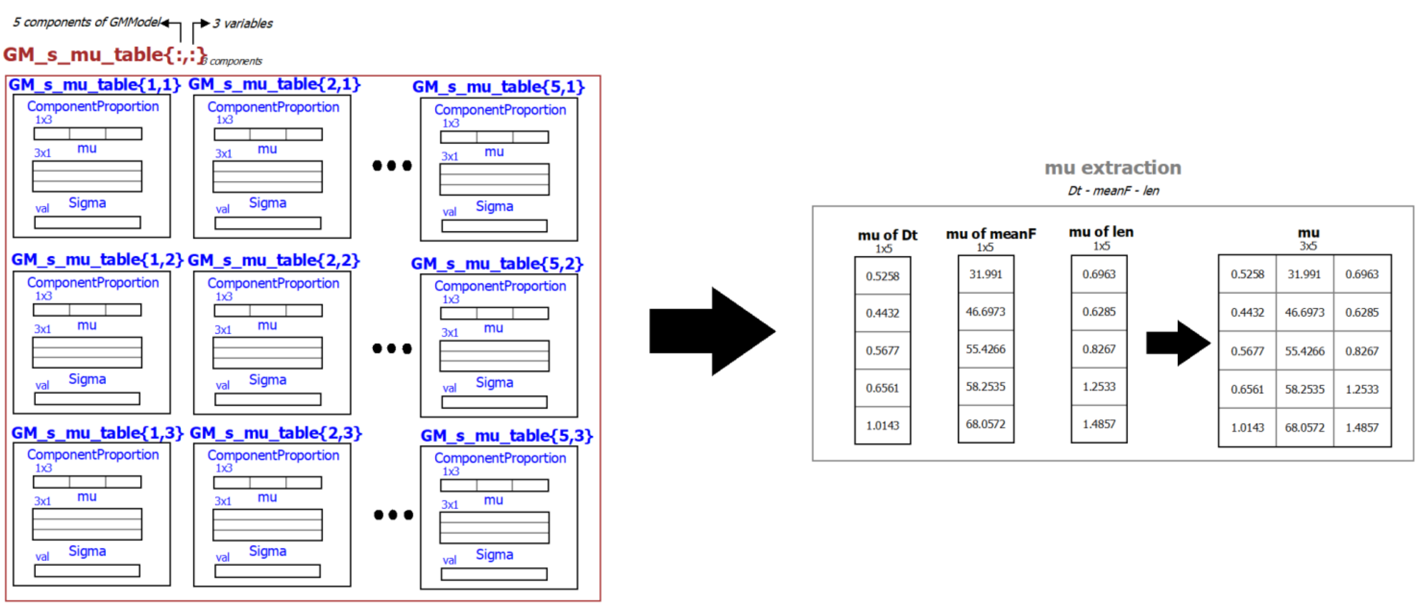


Figure 28: Generating a random mean table

Likewise, the **mixing probabilities** of the final GMM are generated using the 5 GMMs created in the modelisation. Thus. **5 values are generated** from those GMMs and will be used as parameters in the final GMM. The only thing to take into consideration in this case is the fact that the mixing probabilities **must sum to 1**. However, since these are randomly generated, this constraint is not always satisfied and therefore, a **normalisation** in the generated mixing probabilities is required.

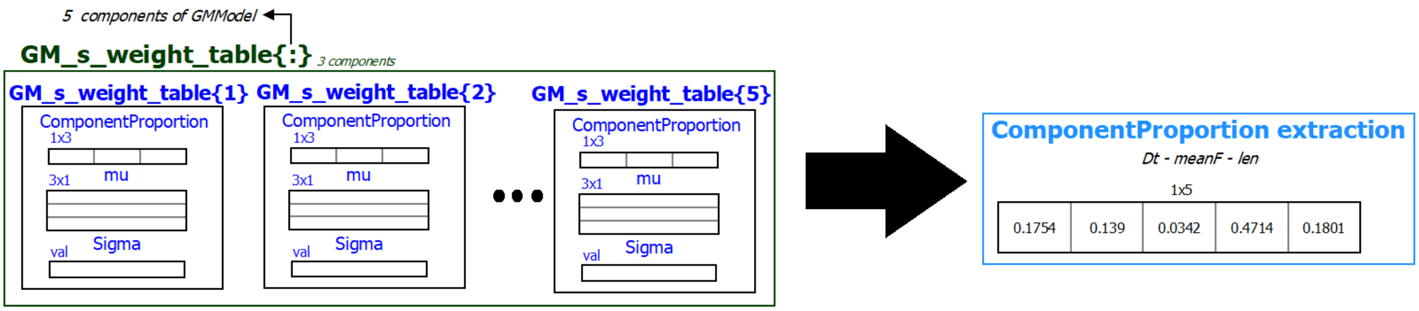


Figure 29: Generating random mixing probabilities

Finally, the **covariance matrix (Sigma)** of the final GMM is generated using the **6 GMMs** created in the modelisation. More specifically, the **6 differentiated values are randomly generated** by the 6 GMMs. From these 6 values, the first 3 are describing the diagonal values of the matrix (variance of each values) while the 4th describes the covariance between the 1st and 2nd variable (thus the values (1,2) and (2,1) of the Sigma matrix), the 5th describes the covariance between the 1st and the 3rd variable (values (1,3) and (3,1) of the Sigma matrix) and finally, the 6th describes the covariance between the 2nd and the 3rd variable (values (2,3) and (3,2) of the Sigma matrix). This way, a **symmetrical covariance matrix** is generated (*figure 30*).

The only problem in this implementation is that, as it is known by the theory, a covariance matrix, apart from symmetrical, is also **positive semi-definite**. However, the random generation does not guarantee this. A workaround in this has been performed and therefore, a **random generation** is performed in a loop up until the generated matrix is positive semi-definite.

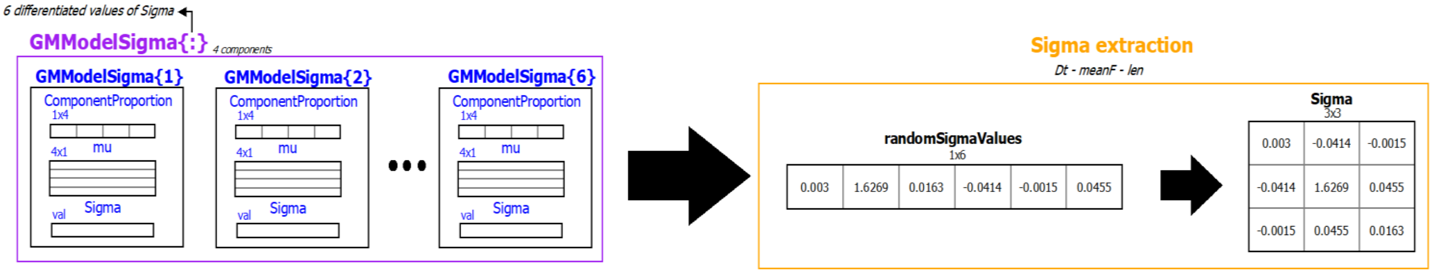


Figure 30: Generating random covariance matrix

The **same procedure** is performed for the extraction of the parameters characterizing the final GMM that describes the random variable of **angle**. In this case, the **mu** parameter is determined by randomly generating the 3 values of mu using the 3 GMMs describing it. The same thing applies to the extraction of the **mixing probabilities**, whereas the 3 values are generated and normalised in order to sum to 1. Finally, the GMM describing the **variance** of the variable is used to randomly generate the random value of **Sigma**. All those are depicted in *figure 31*.

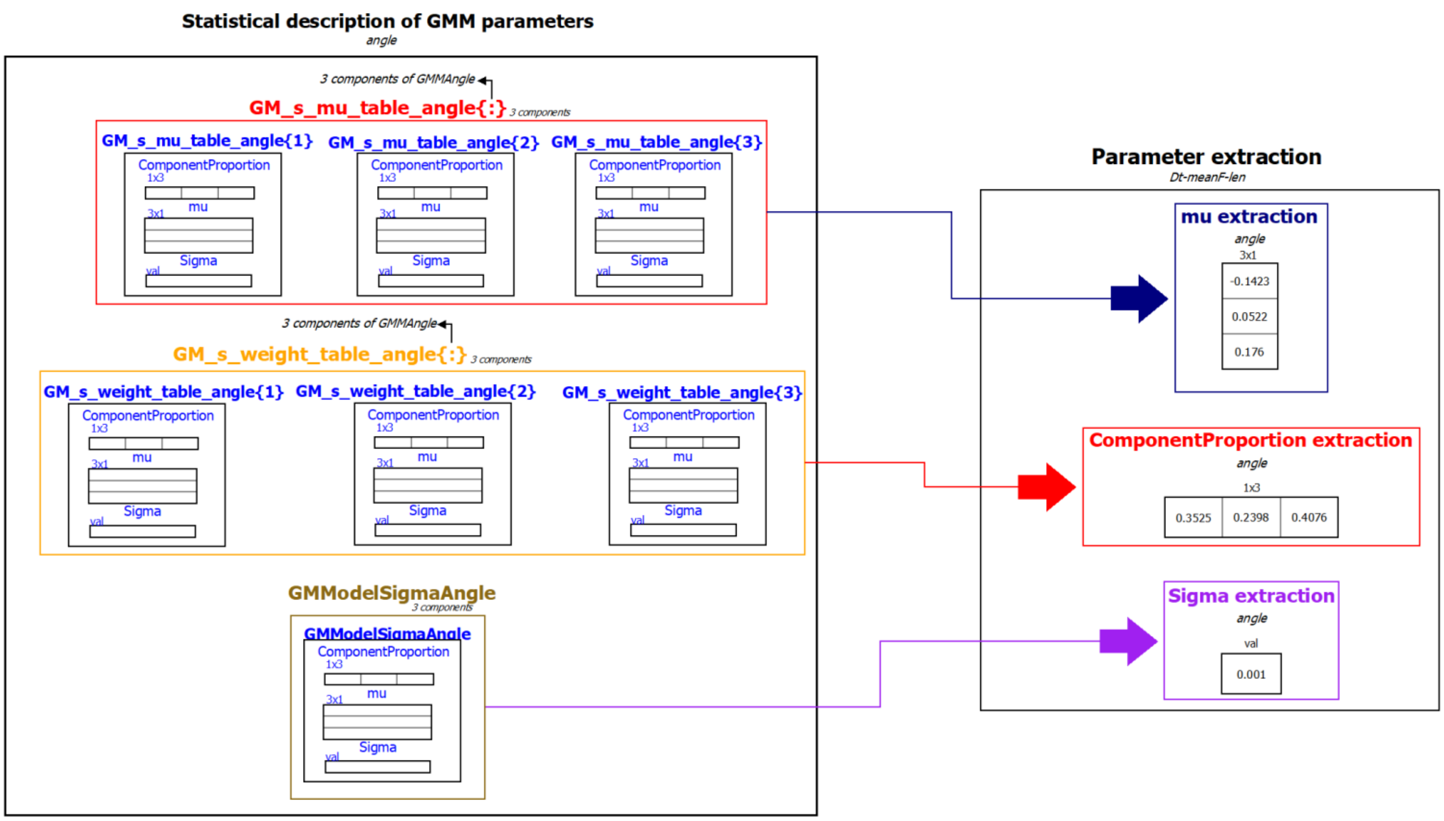


Figure 31: Generating random parameters for the angle variable

Extraction of the random walk

In this step, **two Gaussian Mixture Models** (one for the angle and one for Dt-mean-len) are created using the matlab function $gmdistribution(mu, Sigma, ComponentProportion)$ and the **random parameters** generated in the previous step as input.

Using these GMMs, the random walk is extracted. The GMM describing the **3 variables** (**simulatedGMM**) generates $\boldsymbol{n}$ sets of $\boldsymbol{Dt}$, $\boldsymbol{meanF}$ and $\boldsymbol{len}$ variables using the function $random(simulatedGMM,\ n)$ while the GMM describing the **angle** (**simulatedGMMAngle)** generates $\boldsymbol{n}$ values of angle using the same function. The join of those tables describes a **random walk of $\boldsymbol{n}$ steps**.

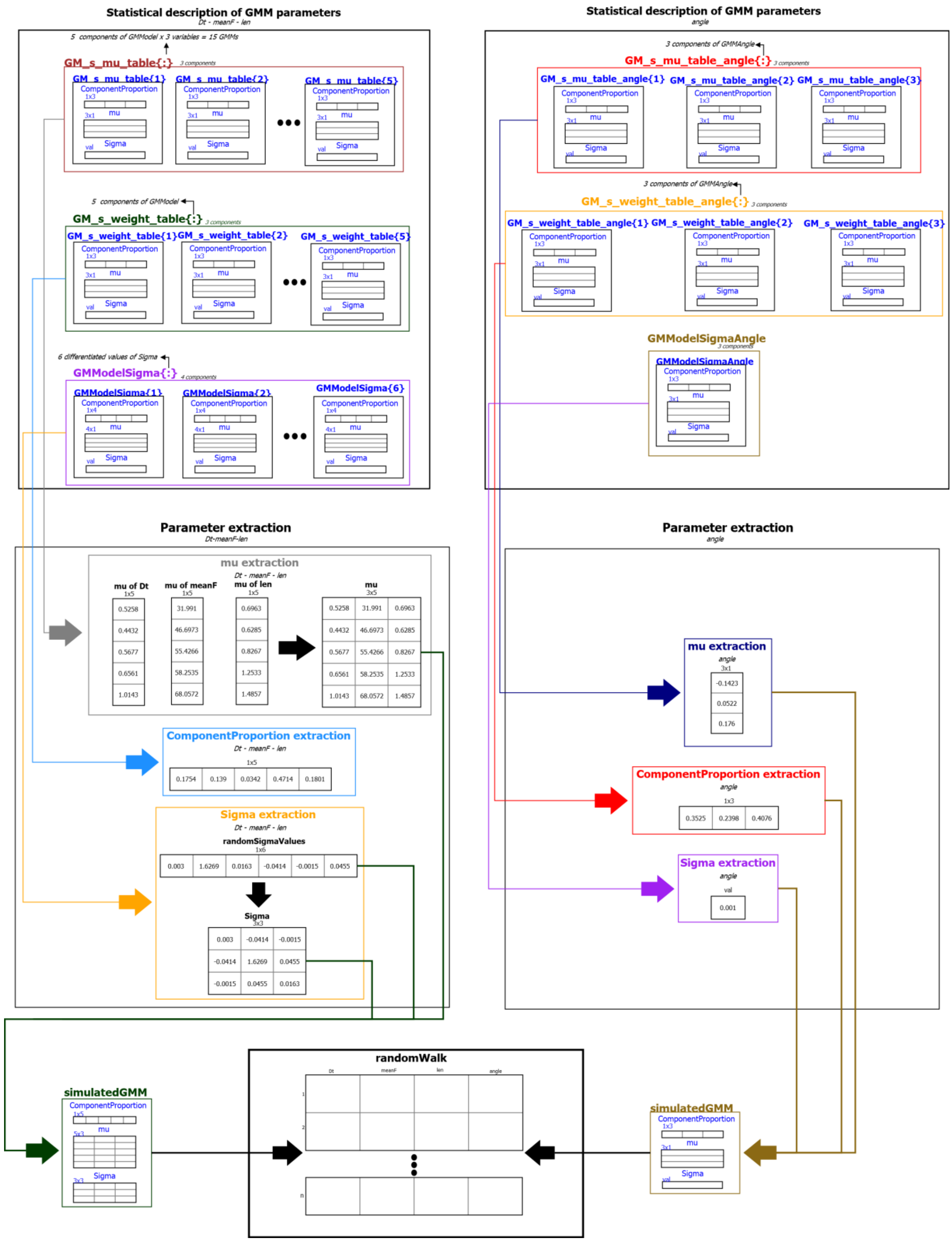


Figure 32: Implementation of the simulator

Evaluation of the simulation

Similarly to *figure 9*, *figure 33* shows the table extracted by the simulator which describes a **randomly generated human gait of 30 steps**. As previously, each column of the table corresponds to the variables of interarrival time (Dt), mean force induced ($meanF$), length (len) and angle of each step, respectively. As a first deduction by comparing the two tables, the values generated can be considered **comparable to the values of a random person** from the database. The only remarkable difference spotted is in the $\boldsymbol{meanF}$ variable. Figure 9 is likely describing a male human with a large weight whose walking induces a big force on each step. However, the database included measurements of **different body types** and sexes and as a result, the measurements describing force contain many different values. In other words, the simulated step is **neutral** in a way and therefore, contain all those different values.

	1	2	3	4
1	0.8112	49.1565	0.7417	-0.2581
2	0.5394	71.4687	0.6599	0.1775
3	0.5638	48.2896	0.4609	0.1879
4	0.9312	49.6309	1.0024	-0.1635
5	0.6665	57.8565	1.2591	-0.2510
6	0.5401	46.5265	0.7537	0.1904
7	0.9173	52.1141	0.8076	-0.0935
8	0.8494	47.9652	0.7456	-0.2636
9	0.5283	73.0216	0.7278	0.1594
10	0.6447	54.7242	1.2783	-0.1570
11	0.5261	71.9611	0.8105	-0.2709
12	0.5921	47.6795	0.5606	-0.1399
13	0.5831	49.6633	0.6402	-0.2439
14	1.0838	50.8945	1.2149	0.2054
15	0.8317	49.5847	0.6289	-0.2375
16	0.8362	50.3186	0.8884	0.1617
17	0.4464	72.7636	0.6881	-0.1362
18	0.5144	74.1892	0.8213	0.2285
19	0.9071	50.5977	0.7564	-0.2742
20	0.6004	48.1749	0.8072	-0.2509
21	0.8199	48.7103	0.7829	0.2908
22	0.5736	74.6675	0.6582	0.1831
23	0.6655	57.0427	1.1608	-0.2444
24	0.8196	50.7135	0.9171	-0.2317
25	0.4763	48.6019	0.5615	-0.2247
26	0.7947	49.7801	0.9104	0.1854
27	0.4972	72.3167	0.9719	0.2343
28	0.5202	73.3215	0.5852	0.1655
29	0.7976	49.3482	0.4664	0.1900
30	0.5140	75.6372	0.7325	-0.2535

Figure 33: Random values of the variables generated by the simulator for n=30

To deepen the understanding about the outcome obtained by the simulator, the **mean speed** of gait was calculated in order to compare it with the theoretical mean speed of a human walking at a normal tempo. Using the length and Dt variables, **speed** can be calculated as:

$$speed = \frac{length}{Dt}\ (m/sec)$$

This formula produces the speed of each step and thus, for the gait of *figure 33*, the **mean speed** was calculated as:

$$meanSpeed = 1.2218\ m/sec = 4.3986\ km/h$$

Which is **close to the average speed of human walk at a normal tempo**.

Finally, using the length and the angle of each step by the generated table, the $\boldsymbol{x}$ and $\boldsymbol{y}$ positions of each step j were calculated as:

$$x_j = \begin{cases} x_{j-1} + len \cdot \cos(angle)\,,\ if\ j > 1 \\ \quad len \cdot \cos(angle)\ ,\ if\ j = 1 \end{cases} \quad \textbf{and}\ \ y_j = len \cdot \sin(angle)$$

These positions were scattered in *figure 34* in order to visualise the generated human walk on the walkway.

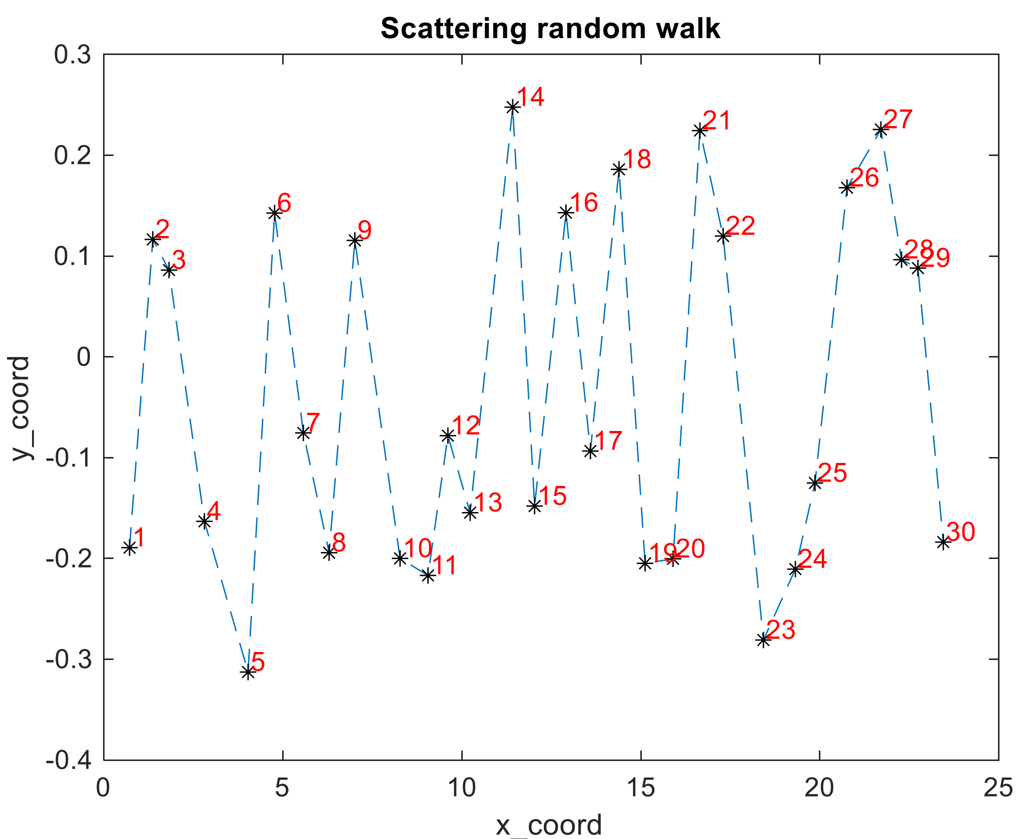


Figure 34: Scattering a random walk

Figure 35 depicts the **first** (out of two) walk of the person used as an example from the database (the one described by the table of *figure 9*).

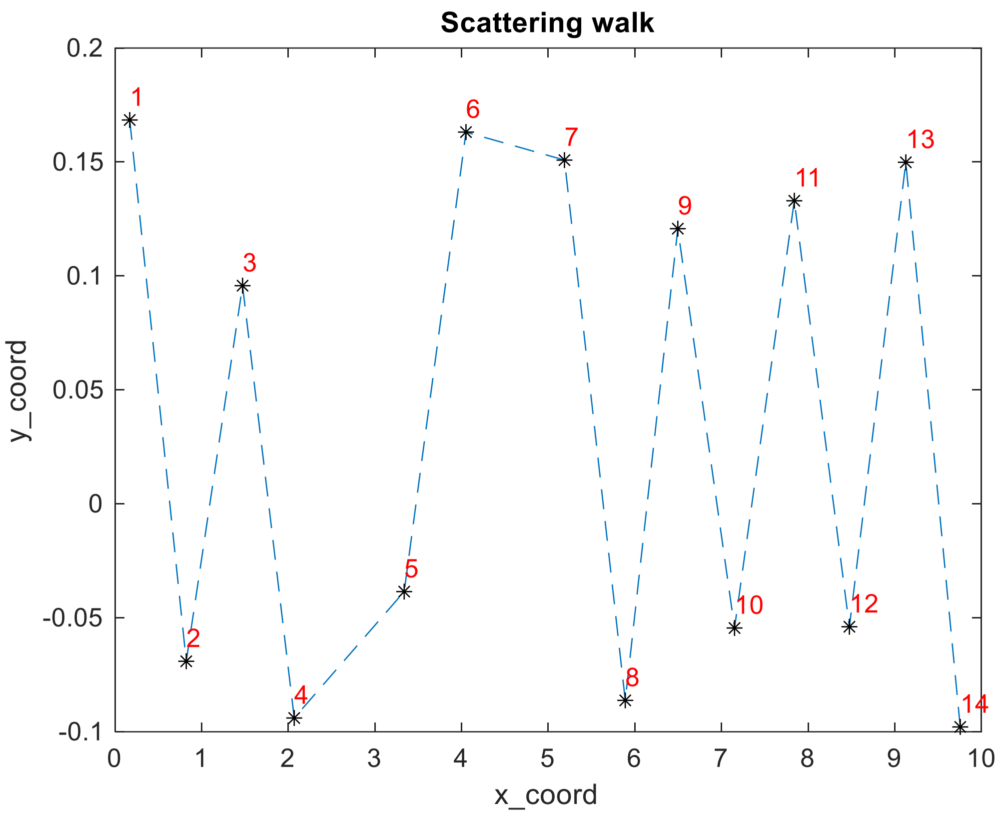


Figure 35: Walk of the subject described by figure 9

As it is visible, the randomly generated walk follows the **same pattern** as the one from the database.

In conclusion, it can be inferred that the outcome of the simulator describes the act of human walk accurately as the data generated match the properties of the experimental data. Thus, this simulator can be used for any further experiments concerning the properties of human walk in order to bypass the experimental procedure.

Outline of the implemented code

The implemented code for all the functions described in this implementation report has been tested on MATLAB R2019b and requires the following files:

- $\boldsymbol{main.m}$: The main file - it connects all the functions to implement all procedures and MATLAB figures included in this report.
- $\boldsymbol{steps\_database.mat}$: The database used for the modelisation.
- $\boldsymbol{clearDb.m}$: The function $clearDb$ that clears trash in database & fills force-time matrixes to match indices.
- $\boldsymbol{retrieveAllVariables.m}$: The function $retrieveAllVariables$ that retrieves all forces, times and coordinates from the database for each step.
- $\boldsymbol{computeAllDesiredVariables.m}$: The function $computeAllDesiredVariables$ that computes the interarrival time between one step and the next, mean force induced, length and angle for each step, as well as clears any wrong values.
- $\boldsymbol{fitGMMtoData.m}$: The function $fitGMMtoData$ that fits a gaussian mixture model to each subject's data and performs the two-sample Kolmogorov-Smirnov tests to determine if the distributions describe the data.
- $\boldsymbol{mu\_weight\_statDescription.m}$: The function $mu\_weight\_statDescription$ that creates a statistical description for mu (mean values) and mixing proportions of mixture components.
- $\boldsymbol{sigmaStatDescription.m}$: The function $sigmaStatDescription$ that create a statistical description for Sigma (differentiated covariance values)
- $\boldsymbol{generateParameters.m}$: The function $generateParameters$ that extracts a random set of parameters to use in the GMM of the simulator

All those files contain extensive comments in order to be easily understandable. A run of the **main** file will produce all the tables acquired by the database and the fitted GMMs for each person, the GMMs of the parameters describing the initial GMMs, the final GMM and finally, the figures included in this report.

The table $\boldsymbol{randomWalk}$ will contain all the important variables of all n steps of the simulated random walk.