MATLAB Practical session IVb: Flow control

Exercise 2: Kinematics

Step 1: Creating subplots with for statement

Open the file 'KS_motion_trial62.mot' with the kinematics values for the simulated motion and plot the ankle, hip flexion, hip adduction and the knee angles in four different subplots on 1 figure. Instead of doing this as performed in exercise session II on plotting, try to do this as efficient as possible, by using the FOR in combination with the SWITCH command. Only one time the subplot command may be written.

- Import data from *'KS_motion_trial62.mot'*
- In total we want to have four different subplots e.g. hip flexion, hip adduction, ankle flexion and knee flexion. Use the FOR statement to create the subplots. The SWITCH command is used to determine the name of the column header to plot. For this dataset we are interested in the following column names: 'hip_flexion_r', 'hip_adduction_r', 'ankle_angle_r', 'knee_angle_r'.
    - Find the column number by searching the matching column header in the imported column header data
    - Create the subplot and plot the data

Step 2: Add titles and labels to the plots

Copy the previous code and adapt it, such that a title is added to each subplot, which state the name of the variable (Hip flexion, Hip adduction, Knee flexion and Ankle flexion). Use an IF statement to add the x-label ('Time (s)') to the lower subplots and the y-label ('Angle (°)') to the most outer left subplots. The output is the same figure as in the Plotting exercise session, but now implemented more efficient.

Step 3: Add another dataset to the figure

Repeat this workflow to open a second data file and plot the same kinematic values on the corresponding subplot. Use a different color for the different files. Implement this also as efficient as possible, using a nested for loop. Only the number in the filename is changed for the second file (*'KS_motion_trial69.mot'*, '69 instead of 62). Use this information to efficiently implement the FOR loop. Use the SWITCH command to change the filename and the color of the line plot.

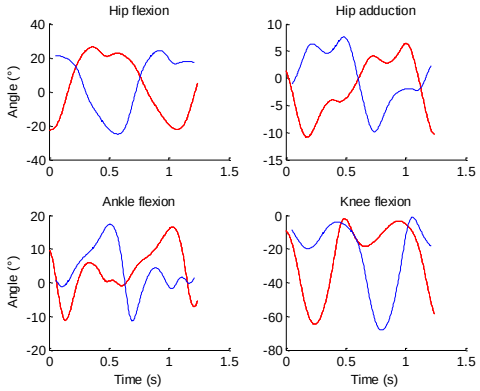

Figure 1 Kinematica data from '*KS_motion_trial69.mot*' and '*KS_motion_trial62.mot*'

Exercise 4: Ages

Step 1: Age of subjects

Determine the age of a person based on the birthdate and a reference date. The birthdate of a person is given by three integers *bd, bm, by* representing the day, month and year of birth. Make a script using the IF statement which determines the age in years of a person at a certain reference date. The reference date is represented by three integers *rd, rm, ry* (day, month, year). Remember to take into account that someone may have had a birthday before the reference date. Test the script for multiple dates.

- Create six variables to define the birthdate and reference date.
- Computation of the age. Remember to take into account that someone may have had a birthday before the reference date. To solve this problem use an IF statement. If the reference date has not been passed during the reference year decrease the age with 1. Make sure no answer or zero is created if the reference day is prior to the birthday.

Step 2: Birthday database

- Import the data from the Excel file named *'Birthdays.xls'*. This file contains birthday data of 1000 subjects. Extract the *'day'*, *'month'* and *'year'* of birth using string compare command ('*strcmp*').
- Copy and adapt the previous code to calculate the age of 1000 subjects. To do so add a FOR statement to execute the previous code for all the subjects.

Step 3: Population groups

To visualize the age distribution for our birthday data we have to create age classes. For this exercise we want to define four age groups (0-25, 26-50, 51-75, 76+). Visualize the distribution over the groups using a bar plot. Save the group number and the age in an excel file as a separated column. For all the age groups we want to know the number of subjects included in the age group. Use a FOR loop to execute the statements for all the subjects. To create the groups use the SWITCH statement. For the definition of the case expressions use a cell array.

- Construct a zeros-vector to store the group number for every subject.
- Construct a zeros-vector to include the number of subjects included in the group.
- Use a FOR loop over all subjects. Define the groups using SWITCH statement. There are four groups in total numbered 1, 2, 3, and 4 for 0-25, 26-50, 51-75, 76+ respectively. (It is also possible to create the age groups using an IF statement. As and exercise you could program this as well).

- Construct a bar plot to visualize the distribution of the ages over the four groups. Add a title and labels to the axes of the bar plot.
- Add the constructed vector containing the age group and the computed ages to the rest of the imported data, and add the column headers 'Age' and 'AgeGroup'.
- Write everything in a new excel file, called *'Birthdays_group.xls'*.

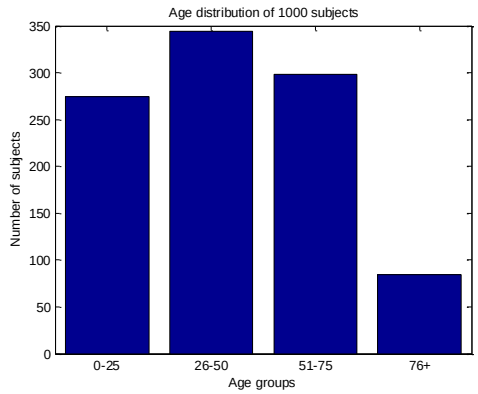

Figure 2 Bar graph showing the age distribution of 1000 subjects

Exercise 5: Metric unit converter

- Write a program to convert metric units to centimeters. Use the SWITCH command to define different cases e.g. mm, cm, dm, m, km. Test your code for different units and values.

```
x = 3.0;          % numeric variable
units = 'mm';     % string variable
```

- Copy the previous code and adapt it to use this program for converting arrays of numbers and units. Use a FOR loop.