# CMP9135M, Computer Vision
# Workshop - Motion and Observation Models for Visual Tracking

## Introduction

You should use MATLAB for the following tasks. If you do not remember how to use it, please refer to the “Getting Started” part in Workshop 1. Alternatively, you can implement the following example using your language and API of choice.

## Task 1: Random noise

In this task you should use the MATLAB command `randn` to generate a vector of normally distributed random numbers (i.e. Gaussian noise). Use `help randn` for a detailed explanation of this command.

```
n1 = randn(1000,1);      % Gaussian noise with mean=0 and std=1
n2 = 5 + randn(1000,1);  % Gaussian noise with mean=5 and std=1
n3 = 0.2 * randn(1000,1);% Gaussian noise with mean=0 and std=0.2
```

Check the means and standard deviations of the above noise vectors, for example:

```
mean(n1)
std(n1)
```

Are you able to explain the values returned by the last two commands?

NOTE: the above MATLAB functions can also be implemented in other languages using the following formulae (*N* is the number of available samples):

$$mean = \frac{1}{N}\sum_{i=1}^{N} x_i \; std = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - mean)^2}$$

Try also to plot the histogram of the noise vectors, for example by using the following command:

```
hist(n1, 50)
```

The result is an image that shows the probability distribution (although not normalized) of the noise vector, expressed by a histogram with 50 bins. Plot the histograms for all the three noises and check if their previous means and standard deviations are consistent with the plots.

Try for different values of mean and standard deviation, different vector sizes and different number of histogram bins.

## Task 2: Brownian motion model

Create the following MATLAB function, which is a possible implementation of the Brownian motion model seen in the previous lecture, and run it as `[x, y] = brownianModel`.

```
function [x, y] = brownianModel()
% generate the 2D trajectory of a point moving according to a
% Brownian motion model

    % initial position vectors
    x = zeros(1000, 1);
    y = zeros(1000, 1);

    % noise vectors
    nx = 0.5 * randn(1000, 1);
    ny = 0.5 * randn(1000, 1);

    % generate trajectory
    for i = 2:1000
        x(i) = x(i-1) + nx(i);
        y(i) = y(i-1) + ny(i);
    end
end
```

Plot the output vectors with `plot(x,y)`. How does the trajectory looks like? Try plotting also the single coordinates with `plot(x)` and `plot(y)`. Are they similar or not? Why?

Change the above function to accept in input different parameters, such as vectors size, initial position, and standard deviation of the noise. Plot the output for different values of these parameters and analyse the results to understand how they influence the final trajectory.

### Task 3: Constant Velocity motion model

Create the following MATLAB function, which is a possible implementation of the Brownian motion model seen in the previous lecture, and run it as `[x, vx, y, vy] = cvModel`.

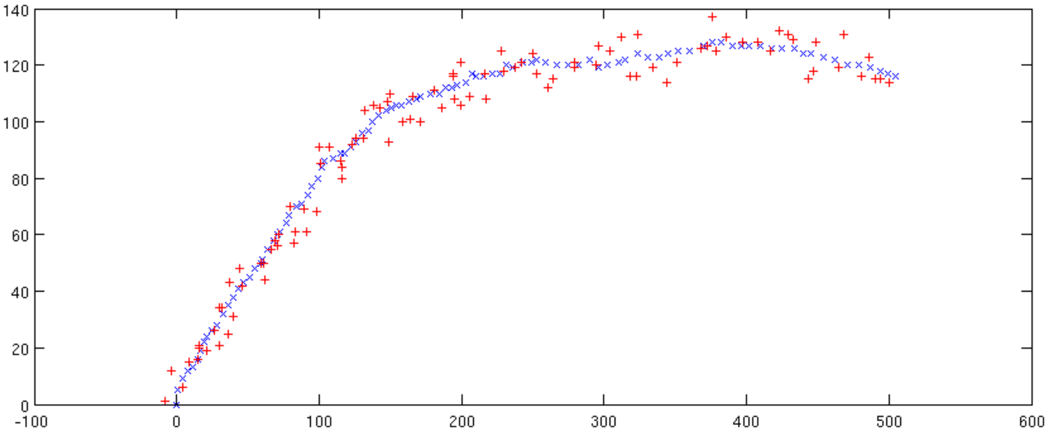
```
function [x, vx, y, vy] = cvModel()
% generate the 2D trajectory and velocity of a point moving
% according to a Constant Velocity motion model

    dt = 0.033;  % time interval

    % initial position and velocity vectors
    x = zeros(1000, 1);     % x position
    vx = zeros(1000, 1);    % x velocity
    y = zeros(1000, 1);     % y position
    vy = zeros(1000, 1);    % y velocity

    % noise vectors
    nx = 0.5 * randn(1000, 1);
    nvx = 0.1 * randn(1000, 1);
    ny = 0.5 * randn(1000, 1);
    nvy = 0.1 * randn(1000, 1);

    % generate trajectory
    for i = 2:1000
        x(i) = x(i-1) + vx(i-1)*dt + nx(i);
        vx(i) = vx(i-1) + nvx(i);
        y(i) = y(i-1) + vy(i-1)*dt + ny(i);
        vy(i) = vy(i-1) + nvy(i);
    end
end
```

Plot the output trajectory with `plot(x,y)`. How does the trajectory looks like? Try plotting also the single coordinates with `plot(x)` and `plot(y)`, as well as the respective velocities `plot(vx)` and `plot(vy)`. Are they as expected?

Change the above function to accept in input different parameters, such as vectors size, time interval, initial position/velocity, and standard deviation of the noise. Plot the output for different values of these parameters and analyse the results to understand how they influence the final trajectory and velocities.

## Task 4: Cartesian observation model

The following graph shows the trajectory of a target moving according to a Constant Velocity model. In particular, the graph shows the real position ($x_p$,$y_p$) in pixels of the target (blue crosses) and the actual position (u,v) in pixels measured by a noisy camera (red crosses). The blue coordinates are contained in the `xp.csv` and `yp.csv` files, while the red coordinates are contained in the `u.csv` and `v.csv` files.



The observation model of the noisy camera can be expressed by the following equations:

$$u = x_p + n_u$$
$$v = y_p + n_v$$

where $n_u$ and $n_v$ are observation noises. Compute the mean and standard deviation of the latter using the above coordinates. Plot also their histograms (5 bins are sufficient in this case). Are these noises zero-mean Gaussians?