Matlab excercises

1. Define 2 vectors
   v1 = [0, 0.1, 0.2, ... , 1]
   v2 = [1, 1.1, 1.2, ... , 2]
2. Compute element-per-element product of the vectors
3. Compute dot product of the vectors
4. Compute cross product of the first three elements of the vectors (use cross function)
5. Create the following matrix:
   $$A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}$$
6. Using the previously defined matrix A create a new matrix B (use vertical concatenation ';' operator)
   $$B=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\\ 10 & 11 & 12\end{pmatrix}$$
7. Create a vector b = [1, 4, 7]
8. Create a vector c = [1, 4, 7, 9] using the previously defined vector b (use horizontal concatenation ',' operator)
9. Solve the set of linear equations
   $Ax = b^T$
10. Solve the set of linear equations using the least-squares method
    $Bx = c^T$
    $B^T Bx = B^t c^T$
    $x = (B^T B)^{-1} B^T c^T$
11. Construct the following matrix (zeros function and for statement)
    $$X=\left\{\frac{1}{i+j-1}\right\}_{i,\, j=1..100}$$
12. Save the matrix in ASCII format and load it to the Y variable
13. Create a time vector t (0..1s) and sample the following signal:
    f(t) = sin(2π * 20 * t) + sin(2π * 35 * t + π/4)
    at different frequencies (100Hz, 500Hz, 1kHz, 2kHz)
14. Plot the signal (use plot function)

Signal sampling

1. Create a time vector tref (0..1s) sampled evenly at frequency 1kHz (it will be used for imitating of the continuous signal)
2. Create a time vector t30 sampled evenly at frequency 30Hz.
3. Sample sine waves characterized by the frequency f using 30Hz sampling rate (and 1kHz for reference)
   - f = 6Hz
   - f = 11Hz
   - f = 28Hz

(use sine wave is in the form $f(t) = \sin(2\pi * 20 * t)$)

4. Plot results as points representing 30Hz samples against the 1kHz sampling continuous background
5. Interpret results. What is the apparent frequency of the sine wave sampled at 28Hz?

Convolution

1. Create a time vector t (0..1s) sampled at 100Hz and sample the following signal:
   $f(t) = 3 * \sin(2\pi\ * t) + \sin(2\pi * 8 * t)$
   and a triangle impuls
   h(t) = [0, 0, 0..........0, 0.5, 1, 0.5, 0, ...0]
2. Create a 101-element vector representing normalized impulse delta = [1, 0............0]
3. Create a 30 element convolution kernel in the form of
   - a normalized delta function
   - an amplification kernel eg. [3, 0, 0,............]
   - a shift kernel eg. [0, 0, 0, 0, 0,...,1, 0, ...]
   - an echo kernel eg. [1, 0, 0, 0, 0, ..., 0.5, 0,...., 0.2, 0, ..., 0]
   - first difference kernel [1, -1, 0, 0, ...]
   - square low pass filter [1/n, ... ,1/n (n times), 0, 0, 0, ...] n = 13
   - high-pass filter (delta function shifted by n/2 minus low pass filter)
4. Convolve the kernels with f, h, and normalized delta impulse signals
5. Plot results (the original and transformed signals)

Image binarization

Create an OpenCV application performing the following tasks:

1. Reading and RGB image
2. Allowing user to select reference point in the image – memorize color channels for the point
3. Compute an image in which each of the pixels represents a distance in the color space between the corresponding image point and the reference point (maximum metrics may be used)
4. Threshold the image created in this fashion (preferring pixels lower than some specified threshold). Control threshold rate with trackbar.